Energy Informatics Review

a publication of the ACM Special Interest Group on Energy Informatics

Chair:
Prashant Shenoy  
UMass Amherst, USA

Vice Chair:
S. Keshav  
University of Cambridge, UK

Treasurer:
Xiaofan (Fred) Jiang  
Columbia University, USA

Information Services Director:
Omid Ardakanian  
University of Alberta, Canada

Executive Committee Member:
Minghua Chen  
City University, Hong Kong

Communications Director:
Yashar Ghiassi-Farrokhfal  
Erasmus University

Social Media Director:
Tian Guo  
WPI, USA

ACM BuildSys Steering Committee Chair:
Farrokh Jazizadeh  
Virginia Tech, USA

ACM e-Energy Steering Committee Chair:
Sebastian Lehnhoff  
University of Oldenburg, Germany

Editor-in-Chief:
S. Keshav  
University of Cambridge, UK

Area Editors:
Omid Ardakanian  
University of Alberta, Canada
Sid Chi-Kin Chau  
Australian National University
Minghua Chen  
City University, Hong Kong

David Irwin  
UMass Amherst
Xiaofan (Fred) Jiang  
Columbia University, USA
Steven Low  
Caltech, USA

Klara Nahrstedt  
UIUC, USA
Astrid Nieße  
University of Oldenburg, Germany
Dan Wang  
Hong Kong Polytechnic University

Anke Weidlich  
University of Freiburg, Germany

Publications Chair:
Stephen Lee  
University of Pittsburgh, USA
Message from the Editor-in-Chief

Dear Reader,

I am delighted to welcome you to the inaugural edition of Energy Informatics Review, the magazine of ACM SIGENERGY. We hope to provide a high-quality outlet for research in the general area of energy informatics. EIR will be a rapid turnaround scholarly publication to serve our emerging community.

This inaugural issue contains extended versions of the three Best Papers in ACM eEnergy 2021. It also includes invited papers from some luminaries in our field. The result is an excellent issue that covers a range of topics, some in depth, and some with a broader vision. I look forward to maintaining this level of excellence in the issues to come.

This issue represents the results of many hours of dedicated work by the Publications Chair, the Area Editors, reviewers, and, of course, the authors. My thanks to them all.

Happy reading!

S. Keshav
Editor-in-Chief

University of Cambridge, UK
Message from the Chair

I am delighted to write this column for the inaugural issue of Energy Informatics Review. Let me begin by thanking S. Keshav and the entire EIR team for all the hard work in making the newsletter go from an idea to reality in record time.

As you may know, our SIG was approved by the ACM a year ago, after functioning as an Emerging Interest Group (EIG) for two years. Our transition to a permanent full-fledged SIG enables us to create a vibrant community of researchers, educators and practitioners in the field of Energy Informatics. Our motivation for forming the SIG is to address the much felt need to bring together the computing community to deal with the critical issues of climate change and the transition of the energy system to a carbon-free future. Computing and data science techniques will be key drivers of this transformation!

Since its inception, the volunteer group in charge of running the SIG has been hard at work. The SIG is responsible for the organization of two ACM conferences—e-Energy and Buildsys—and associated workshops. We have worked closely with the organization and steering committees of both conferences to navigate conference organization in the face of pandemic uncertainties. ACM e-Energy was successfully held earlier this summer in a virtual format, and ACM Buildsys will be held in November in a hybrid format. An important goal for the SIG to build a diverse and inclusive community. We are supporting a travel grant program to enable students to attend Buildsys, especially in the light of pandemic travel challenges. Online attendance at Buildsys is free this year for all attendees, and we look forward to your participation.

Workshops are an important part of our conference program. Both e-Energy and Buildsys have organized numerous workshops on cutting-edge topics this year. These workshops span topics such as Climate Change and Sustainability, Machine Learning for Energy, Fairness Accountability and Ethics in Energy systems, and Accessibility of Buildings and Cities.

ACM e-Energy has instituted a new paper process this year which will enable authors to revise and resubmit paper to address reviewer feedback during the review cycle. We hope that this new process will enable the conference to accept more paper while maintaining the high quality of the program.

Finally, we are instituting an awards program to recognize junior, and senior, members of our community and to recognize past papers that have had a strong impact. More information on the awards program will be announced soon. We also established a social media presence for the SIG. Please visit us at energy.acm.org or follow us on social media for up-to-date information about the SIG’s activities. We also look forward to your ideas on how the SIG can serve you better.

Prashant Shenoy
Chair, ACM SIGEnergy

University of Massachusetts Amherst, USA
Contents

Information Batteries: Storing Opportunity Power with Speculative Execution ........ 1
Jennifer Switzer (UC San Diego, USA); Barath Raghavan (USC, USA)

Data Analytics for Cybersecurity Enhancement of Transformer Protection ........ 12
Martiya Zare Jahromi (University of Toronto); Amir Abiri Jahromi (University of Toronto);
Deepa Kundur (University of Toronto); Scott Samer (University of Toronto);
Marthe Kassouf (Hydro-Quebec Research Institute (IREQ))

The Next Stage of Green Electricity Labeling: Using Zero-Knowledge Proofs for Blockchain-
based Certificates of Origin and Use ......................................................... 20
Johannes Sedlmeir (University of Bayreuth, Germany); Fabiane Völter (Fraunhofer FIT, Germany);
Jens Striiker (University of Bayreuth, Germany)

Privacy-Preserving Energy Storage Sharing with Blockchain and Secure Multi-Party Com-
putation ............................................................................................................. 32
Nan Wang (Australian National University); Sid Chi-Kin Chau (Australian National University);
Yue Zhou (Australian National University)

Towards Online Optimization for Power Grids .................................................. 51
Deming Yuan (Nanjing University of Science and Technology); Abhishek Bhardwaj
(Australian National University); Ian Petersen (Australian National University);
Elizabeth Ratnam (Australian National University); Guodong Shi (The University of Sydney)

Online Peak-Aware Energy Scheduling with Untrusted Advice .......................... 59
Russell Lee (UMass Amherst); Jessica Maghakian (Stony Brook University);
Mohammad Hajiesmaili (UMass Amherst); Jian Li (SUNY-Binghamton University);
Ramesh Sitaraman (UMass Amherst & Akamai Tech); Zhenhua Liu (Stony Brook University)

A Holistic Review on Advanced Bi-directional EV Charging Control Algorithms .... 78
Xiaoying Tang (The Chinese University of Hong Kong, Shenzhen); Chenxi Sun (The Chinese
University of Hong Kong, Shenzhen); Suzhi Bi (Shenzhen University); Shuyao Wang (Shenzhen
University); Angela Yingjun Zhang (The Chinese University of Hong Kong)

Empiricism and Collaboration on Grid Data Analytics: The Need for a New Information
Ecosystem .......................................................................................................... 89
Alexandra von Meier (UC Berkeley); Laurel N. Dunn (PingThings, Inc.)

Characteristic Profile: Improved Solar Power Forecasting Using Seasonality Models ... 95
Julian de Hoog (University of Melbourne); Manessha Perera (The University of Melbourne);
Peter Ilfrich (Melbourne, Australia); Saman Halgamuge (The University of Melbourne)

Automated Generation of Models for Demand Side Flexibility Using Machine Learning:
An Overview .................................................................................................... 107
Kevin Förderer (Karlsruhe Institute of Technology); Veit Hagenmeyer (Karlsruhe Institute of
Technology); Hartmut Schmeck (Karlsruhe Institute of Technology)
Public review for

Information Batteries: Storing Opportunity Power with Speculative Execution

Jennifer Switzer, Barath Raghavan

The large scale penetration of intermittent renewables in the energy mix has increased the availability of negative-priced excess power production, which is termed as opportunity power. This excess production can be absorbed using grid-scale energy storage, which is prohibitively expensive today, or elastic loads in the grid. Data centres are the prime example of large elastic loads. This paper contributes to the growing literature on data centre demand response and speculative load shifting by showing the feasibility of performing function-level pre-computation and caching using microbenchmarks and real market prices. The authors augment the Rust compiler toolchain to add memoization to pre-computable tasks, and develop recurrent neural networks for predicting the future availability of opportunity power and upcoming tasks in data centres.

Public review written by

Omid Ardakanian
University of Alberta, Canada
Information Batteries

Storing Opportunity Power with Speculative Execution

JENNIFER SWITZER, UC San Diego, USA
BARATH RAGHAVAN, USC, USA

Coping with the intermittency of renewable power is a fundamental challenge, with load shifting and grid-scale storage as key responses. We propose Information Batteries (IB), in which energy is stored in the form of information—specifically, the results of completed computational tasks. Information Batteries thus provide storage through speculative load shifting, anticipating computation that will be performed in the future.

We take a distributed systems perspective, and evaluate the extent to which an IB storage system can be made practical through augmentation of compiler toolchains, key-value stores, and other important elements in modern hyper-scale compute. In particular, we implement one specific IB prototype by augmenting the Rust compiler to enable transparent function-level precomputation and caching. We evaluate the overheads this imposes, along with macro-level job prediction and power prediction. We also evaluate the space of operation for an IB system, to identify the best case efficiency of any IB system for a given power and compute regime.

1 INTRODUCTION

Within the next twenty years humanity must eliminate fossil fuel use to avoid dangerous climate change [28, 37]. To do so requires renewable electricity generation. While there have been dramatic decreases in the cost of wind and solar, their intermittency in response to weather and solar irradiance is a widely-known issue [9, 10]. This intermittency is often out of phase with overall demands, so when renewable production is high, prices tend to be low (Figure 1).

The research community has explored myriad responses to this intermittency [3, 13, 22, 27, 33, 34, 51, 52, 55, 60], but has been stuck between the Scylla of dynamic load shifting and the Charybdis of expensive grid-scale storage. Dynamic load shifting at grid scale requires the rare combination of flexible, large, and ubiquitous loads. Grid-scale storage with proven technology requires nearby hydroelectric capacity or expensive battery arrays.

In this paper we aim for the best of both worlds: storage of surplus renewable production through the load shifting of computation with speculative execution. Computation is near-ininitely divisible, flexible, large, ubiquitous, and can be stored cheaply. This approach, Information Batteries (IB), entails storing energy as completed precomputations that can, as we show, meet or exceed the end-to-end efficiency of grid-scale storage using existing infrastructure. Not all workloads or conditions will yield high efficiency with this approach, so a key aspect of our exploration is its limits.

Background. Even with relatively modest adoption renewables and despite the inherent statistical multiplexing of large power grids, grid operators will soon face the problem of too much power. During the middle of the day in California it is now often the case that there is too much power being produced, largely due to solar, driving the price of electricity negative [9]. This problem has arisen with just

20% of California’s electricity generation coming from solar (the renewable source with the greatest growth potential) [2].

Today we face both power dumping and load shedding due to insufficient power during peak times. This disconnect in supply and demand, as renewable energy sources under-produce during low availability times and over-produce during high availability [58, 61], is the key challenge that must be solved to adopt renewables.1

An oft-proposed strategy to address this problem is to increase grid-scale storage via traditional energy storage systems such as lithium-ion batteries and pumped hydro [6]. However, such systems require a relatively high initial investment and have sitting constraints; also, adding enough storage to soak up all excess production is prohibitively expensive [10]. Similarly, smart-grid advocates often observe that if the grid could simply signal to individual devices, such as appliances, when to consume power, this demand shifting could adapt to power availability. However, as with storage, demand shifting requires both grid upgrades—a proposition that a number of governments have balked at—and wide-scale adoption of new smart loads.

Approach. With Information Batteries, we propose the storage of energy as information, using the large and growing footprint of computing to perform both the functions of storage and load shifting. This approach hinges on three observations. First, data centers worldwide consume large amounts of electricity (250–500 TWh in 2018) and are projected to become even more power-hungry

---

1This negative-priced power and curtailed power (dumped power) are together referred to as opportunity power [10].
As we show, it seems that this is only possible for some workloads generation, not the peaks and troughs of generation.

Contributions.

This paper makes three contributions:

1. We introduce the idea of Information Batteries that provide a new speculative load shifting mechanism to address grid-scale renewable energy intermittency.

2. We explore the design space of Information Batteries and show that in some common power and compute regimes there is the potential for an IB system to deliver efficiencies better than the best grid-scale storage available while in others IB systems provide little benefit.

3. We implement a proof-of-concept IB system by augmenting the Rust compiler to enable transparent function-level precomputation and caching. We evaluate overhead, along with macro-level job prediction and power prediction.

2 CONTEXT

In this section we motivate the need for and feasibility of Information Batteries. We focus on two renewable energy markets: the Midcontinent ISO (MISO), which operates in the Southern and Midwestern United States and parts of Canada, and the California ISO (CAISO), which covers all of California. This allows us to narrow our scope while still considering both wind-dominant (MISO) and solar-dominant (CAISO) markets [35].

2.1 Availability of opportunity power

Opportunity power in CAISO and MISO is significant, growing, and often available. Current estimates place the yearly combined opportunity power of CAISO and MISO in 2017 between 7–20 TWh per year [9, 10]. In MISO, opportunity power is available 99% of the time (meaning opportunity power is available somewhere 99% of the time, since prices are location-dependent), and often in intervals of >100 hours [10]. In CAISO, some solar generators experience 3.3 hours of opportunity power per day [9].

Solar and wind energy are projected to be the fastest growing sources of electricity generation in the U.S., and currently account for 10% of total U.S. electricity generation [7, 18]. As solar and wind generation grow, so too will the amount of curtailed and negative priced power. Indeed, [9] measures the compound annual growth rate of opportunity power in CAISO to be 40%. Assuming 1.5 TWh of opportunity power in CAISO in 2017 (a conservative estimate) and a constant growth rate, CAISO alone could provide 22 TWh of opportunity power by 2025, enough to power all of Los Angeles.

2.2 Limitations of traditional energy storage

Energy storage is a simple response to overproduction. However, current battery prices make this untenable; grid-scale lithium-ion storage costs $356 per kWh today [38], not including installation costs. A naïve analysis of CAISO and MISO data, assuming 1.5 TWh/year and 6 TWh/year of opportunity power, respectively, yields a conservative estimate of $35 million to add one hour of storage to CAISO, and $140 million to add one hour of storage to MISO. A more complex analysis [10] suggests that adding grid-scale storage provides diminishing returns, and that adding 50 hours of storage to MISO would cost $50–400M per wind generation site, on par with the cost of the turbines themselves.

2.3 Non-computational load shifting

Existing non-computational flexible loads include manufacturing facilities, EV charging, and adaptive home appliances [15, 52]. Household or otherwise small-scale (but widespread) loads are a popular demand shifting target, such as in smart homes with or without storage [3, 22, 27, 33, 51, 55, 60] or in smart buildings more generally [34]. EV charging is a growing, flexible load [13]. Such load shifting requires accurate prediction capabilities [30, 32] such as of renewable generation and weather [5, 46, 47].

2.4 Computational load shifting

Some have considered to simply store power in data centers using the batteries in those facilities rather than shifting load [19, 21, 59]. Prior work has suggested shifting the loads themselves to leverage surplus power [11], for example by examining the price for computation (or power itself) in different regions [31, 41].

While some companies, such as Google, balance their data center power usage with renewable energy power purchase agreements, this still addresses only average power generation, not the peaks and troughs of generation.
Data centers have more than enough capacity to soak up opportunity power. American data centers consume 70TWh/year, 1.8% of the country’s total energy consumption [49]. With opportunity power in CAISO and MISO estimated at 7–20TWh/year [9, 10], opportunity power has the potential to provide between 10–30% of the energy needed by data centers.

Prior work also considered scheduling large-scale compute tasks under a variety of constraints. Speculative execution has been long known in computing [8, 50], but has not been applied in this context. Google recently announced their “carbon-intelligent computing platform” that attempts to reduce carbon emission of datacenters by aligning time-insensitive tasks with periods of high renewable energy availability [42]. In this paper, we will refer to this and similar techniques as time-shifted compute.

3 INFORMATION BATTERIES

Next we address the challenges of: 1) properties of computational tasks, hardware, and energy grids that make them more or less suited to Information Batteries, 2) the key elements of an IB system, and 3) the best case scenario/theoretical limits of IB systems.

3.1 Prediction

There is an element of speculation to any demand shifting: the assumption that the work done will be useful at some future time. However, most flexible loads are general tasks that will almost certainly be useful (e.g., charging an EV, washing clothes), and are therefore non-speculative. Precomputation, on the other hand, is speculative. Given the infinite space of possible computations, more work is needed to identify computational work that will be useful at some point in the future. The ability to predict future computations—to perform task prediction—is therefore key.

3.2 Granularity

Prior work on flexible loads has focused on high-granularity tasks—washing clothes, charging a car, even time-shifting large-granularity computational tasks [42]. Small demand shifting is comparatively understudied. IB systems allow for tuneable load granularity, since computational tasks can (at least in theory) be broken into different-sized tasks (although data dependencies and complexity considerations may in practice favor large tasks).

3.3 Speculative demand shifting

The IB approach of demand shifting differs from prior approaches in two ways: the load itself is not pre-existing (as it is speculative), and the granularity at which we shift varies. Computation can be speculatively executed at many granularities, from whole-system to individual instructions.

Speculative load shifting uses energy to store not energy but information. This is different from conventional load shifting, which does not require storage (as the load itself has been shifted). However, since data storage is far less expensive than energy storage, this requirement is a minor imposition.

3.4 Computational loads

Information Batteries are well-suited to tasks with high predictability and a large potential speedup. Potential speedup is the ratio between run time and IB cache latency. Given the same cache latency, longer running tasks will have higher potential speedup. Applications that fit these requirements include:

1. **Machine learning.** OpenAI notes that AI workloads, particularly training, are growing exponentially [39]. Such workloads are ideal for Information Batteries due to their size, latency insensitivity, and high predictability.

2. **Video transcoding.** Video streaming now accounts for 75% of web traffic [4], so video transcoding—the process of converting a video from one resolution to another—has become an important cloud workload. Video transcoding has the potential to be highly predictable, since consumer behavior drives which videos are requested.

3. **Large-scale data analytics.** Companies like Facebook collect a huge amount of data [24], the analysis of which is both time and resource intensive. In many cases, this work can be performed asynchronously.

In addition to predictability of macro workloads, the IB approach can precompute some or all of the fragments of expected jobs and then reassemble these fragments on demand. Thus predictability extends not only to whole jobs but sub-job fragments. Furthermore, with careful binary and execution trace analysis, it is likely possible to perform computations using speculatively precomputed fragments of different jobs, as many workloads have some commonality. Indeed, assembling whole computations out of fragments of code is well studied in the literature in very different contexts, such as in the case of Return-oriented Programming [43, 45]. Exploring the efficiency of fragment precomputation and reassembly is worthy of study but beyond the scope of this paper.

3.5 Grid properties

For an IB system to be successful, renewable generation must be large, long-lasting (e.g., with high duty cycles), and predictable. This is influenced by consumption patterns, the power mix of the grid, the physical locations of power sources, and the amount of traditional (battery) storage available.

3.5.1 **Renewable energy availability.** The grid power mix impacts energy availability and pricing. Energy generation from fossil fuels can be scaled up or down to meet demand, hydro generation is relatively stable and flexible, and intermittent renewables (primarily wind and solar) produce according to environmental conditions. These different patterns have an impact on pricing: when generation is higher than demand, prices drop, as expected.

3.5.2 **Power prices.** Negative power prices are the best indicator of uneconomic production. Prices are used as a proxy for renewable oversupply because, unlike renewable production data, they tell us something about the current balance of supply and demand. If renewable production is high, but so is consumer demand, then adding more demand will be less useful. In order to maximize the benefits of the system, we should prioritize scheduling work for periods of time when renewable production is high and demand is low, e.g. when prices are low.
3.6 Framework
The key idea behind Information Batteries is quite simple: when renewable energy is available in excess, we use it to speculatively perform computation. The challenge is in determining what computation to perform, where and when, and how these computations should be done to make it efficient to retrieve their results later. The energy expended to perform the computation is therefore stored as the result of a computational task.

4 DESIGN
Next we describe the key elements of an IB system, with a focus on two goals: to maximize the shift of compute to opportunty power, and to do so at a lower cost than traditional energy-storage systems.

4.1 Overview
We explore the components of an IB system, from compiler toolchains to ML-based tools for power and job prediction. We discuss how these could be combined in a production system.

Compiler Toolchain. Application binaries must be augmented to allow for run-time caching and retrieval of results.

Key-Value Store. The system must include an efficient key-value store for run-time caching and retrieval. In particular, the latency of precomputed result retrieval from the key-value store must be significantly less than the latency of computing that result directly.

Job Scheduler. A management system must be in place for scheduling incoming jobs during periods of opportunity power, and returning results once they are available. In particular, jobs must be prioritized according to their latency sensitivity and run time. The job scheduler must therefore be aware of the target workload.

Predictors. Information Batteries rely on speculative execution, and therefore require accurate predictions of future conditions. Specifically, the IB system must be able to predict: 1) whether or not opportunity power will be available in the near future (e.g., what the price of energy will be) and 2) what jobs will be requested in the near future, so that these jobs can be precomputed.

4.2 Program instrumentation
Memoization is a well-known technique used to improve the performance of programs by caching the results of expensive computations for later use. Previous work has considered the benefits of program memoization at the trace level, the basic-block level, and the function level [14, 20, 25]. In this paper, we employ function-level memoization, but later propose generalizations to this.

Prior work has shown that a subset of functions can benefit from software memoization at compile and load time [53]. These functions share the following characteristics: expensive, side-effect free, critical, and repetitive arguments [54].

We apply these memoization techniques to precompute results for memoizable functions in the program’s critical path. These results obviate later execution with non-renewable power. We achieve this by instrumenting all application binaries to use a key-value store to memoize the results of function calls.

4.2.1 Compiler extensions. We consider one specific approach to program instrumentation—compiler extension—and highlight other possibilities later. In this approach, all source code is compiled using our customized compiler, which inserts the appropriate hooks to precompute and retrieve precomputed results.

For each call to a precomputable function, the compiler inserts a fetch instruction to first check whether the function has been precomputed, and return the precomputed result if it exists. Note that since the fetch occurs at run time, it is important that its overall latency is significantly lower than the run time of the function to be computed. It is therefore important both that our cache implementation be efficient, and that our definition of precomputable function excludes those with extremely short runtimes.

4.2.2 Caching infrastructure. The IB cache must enable fast caching and retrieval of precomputed results. In addition, for deployability, we would like to use a generic key-value store for caching rather than a IB-specific system. Here, latency and hit rate are important because we do not wish cache retrieval to itself induce higher overheads than computation itself. In addition, the memory footprint of the cache is important; results that require too much storage will increase the cost of the cache infrastructure itself.

---

3 There is a long history of work in lower-level aspects of the relationship between energy and information, such as work on adiabatic computing [16].

4 The definition of precomputable function is flexible, and may differ between workloads.
There are three main components to the IB manager: the scheduler, which receives computational tasks and schedules them to maximize grid power savings; the price predictive engine, which makes predictions about upcoming power prices; and the precomputation engine, which makes predictions about upcoming tasks, and performs precomputations on these predicted tasks whenever opportunity (negative-priced) power is available. If the incidence of negative-priced power is too low, the threshold of what is considered “negative” can also be set to some small positive number.

The scheduler receives tasks, and determines when and how they should be computed. Submitted tasks are of the form source code, deadline, where deadline indicates the latest timestamp at which the result is needed.

If opportunity power is currently available, the task can be computed immediately. Otherwise, the scheduler asks the renewable predictive engine when opportunity power will next be available. If the next available window of opportunity power is within the task’s deadline, the task is scheduled.

Note that even if a task is not precomputable, the scheduler will still attempt to schedule it for a period of opportunity power. Thus any task with a generous enough deadline can be scheduled to use opportunity power. The scheduler also forwards the source code and the time it was received to the precomputation engine, which bases its predictions on this real-time stream of task requests.

The renewable predictive engine forms the core of the scheduler; the scheduler’s overall effectiveness is dependent on the ability of the solar predictive engine to correctly predict opportunity power. An inaccurate solar predictor risks missing out on opportunity power, or erroneously scheduling tasks during regular grid power.

The precomputation engine is responsible for predicting upcoming tasks, pre-computing them, and caching the results. There are two main components of the precomputation engine: the task predictor and the precompute manager. The task predictor receives a continuous stream of task requests from the scheduler and uses a recurrent neural network to predict task requests.

The precompute manager consists of a single event loop that queries energy prices every 5 minutes until they fall below some small threshold. At this point, the manager requests 5 minutes worth of computational tasks from the task predictor. The precomputation engine functions in 5-minute increments since that is the smallest granularity at which energy prices are set in CAISO and MISO.

### 4.3 Scheduling

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache latency</td>
<td>Storage and retrieval overhead</td>
</tr>
<tr>
<td>Cache hit rate</td>
<td>Accuracy of task predict model</td>
</tr>
<tr>
<td>Job length</td>
<td>Average job length</td>
</tr>
<tr>
<td>Price predict false positives</td>
<td>Fraction of time model mistakenly predicts negative-priced power</td>
</tr>
<tr>
<td>Price predict false negatives</td>
<td>Fraction of time model fails to predict negative-priced power</td>
</tr>
</tbody>
</table>

Table 1. Input parameters for the system-level IB simulator.

<table>
<thead>
<tr>
<th>Before Instrumentation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ...</td>
</tr>
<tr>
<td>2 fn square(a: u32) -&gt; u32 ( a * a );</td>
</tr>
<tr>
<td>3 let result = square(x);</td>
</tr>
<tr>
<td>4 ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>After Instrumentation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ...</td>
</tr>
<tr>
<td>2 fn square(a: u32) -&gt; u32 ( a * a );</td>
</tr>
<tr>
<td>3 let result = memoize(square, &quot;square&quot;, x);</td>
</tr>
<tr>
<td>4 fn memoize(f: fn(usize) -&gt; usize, fname: String, a: usize) -&gt; usize {</td>
</tr>
<tr>
<td>5 let cached = db.get(fname, a);</td>
</tr>
<tr>
<td>6 if cached.is_some() {</td>
</tr>
<tr>
<td>7 return cached.unwrap();</td>
</tr>
<tr>
<td>8 }</td>
</tr>
<tr>
<td>9 else {</td>
</tr>
<tr>
<td>10 let computed = f(x);</td>
</tr>
<tr>
<td>11 db.put(fname, a, computed);</td>
</tr>
<tr>
<td>12 return computed;</td>
</tr>
<tr>
<td>13 }</td>
</tr>
<tr>
<td>14 }</td>
</tr>
</tbody>
</table>

Fig. 4. Instrumented programs first check for precomputed, cached results and use those results if found.

### 4.4 Integration

Information Batteries are designed to work with existing data centers. Some, very limited processing power is reserved for the IB manager, which manages the scheduling of both real-time computational tasks and precomputation. A cluster of machines or VMs is designated for precomputation. The IB cache, which stores the results of these precomputations is kept local for quick retrieval. No additional infrastructure is needed.

Although Information Batteries have thus far been described as an alternative to traditional energy storage, it is also possible to use them in conjunction with traditional batteries. Figure 2 illustrates the flow of energy in a traditional and information battery system, and how these two could be combined.

### 5 IMPLEMENTATION

Next we describe our proof-of-concept implementation of Information Batteries, which has three key components: 1) a Rust compiler instrumentation, which replaces function-level precomputation. 2) a price prediction model for both CAISO and MISO, and 3) a function-level task prediction model. Benchmarking these components allows us to realistically parameterize our IB simulator.

#### 5.1 Rust compiler instrumentation

We augment the Rust compiler to do function-level precomputation. This is accomplished at the MIR (Mid-Level Intermediate Representation) stage of the Rust compiler [29]. At this stage, the program has been converted into a control-flow graph (CFG) representation [44]. We implemented our instrumentation as an extra pass through the CFG, which we call the memoize pass.

The memoize pass replaces each function call with a call to the memoize function

---

1. We require that the input binary include the definition of memoize.
and 2) executes and caches the result of executing the function if not. We refer to this instrumentation as the \texttt{memoize} wrapper. For simplicity, we support only functions with the function signature \texttt{fn(u32)->u32}, but in practice other signatures could be supported.

Figure 4 shows the instrumentation of a simple program with \texttt{memoize} (code simplified for readability).

5.2 Cache
The performance of the precomputation engine is highly dependent on the cache implementation. Latency should be minimized as much as possible. For our proof of concept, we use pickleDB-rs [40] to implement a simple, local key-value store. However, any key-value store could be used in practice.

5.3 Price Predictor
We implemented our price models using TensorFlow, as Recurrent Neural Networks (RNN) with one LSTM layer and one dense layer. This is similar to techniques used for weather prediction [30, 57]. We collected training data from historical 5-minute Location Marginal Prices (LMPs) reported by MISO [36] and CAISO [23].

LMP is the dollar cost of supplying the next MW of power at a specific geographic region [48]. Since pricing is location dependent, the model makes predictions based on time of day and geography. Given a location and a 5-minute interval, the price predictor returns a prediction for the next hour’s worth of LMP prices. We define opportunity power as being available any time the LMP prediction drops below some small, tune-able threshold.

5.4 Task Predictor
We implemented task prediction using TensorFlow, as an RNN with one LSTM layer and one dense layer. We collected training and validation data from several open-source Rust crates: Substrate [17], Iced [26], and Juniper [12]. For each crate, we generated a function-level trace on their provided example applications, and used these traces to train the model.

The resulting model takes as input a series of function calls, and a specification, \( N \), for how far in the future to predict. It returns a prediction for the next \( N \) function calls. This is similar to techniques used for text prediction [56]. Note that this approach is quite simplistic. We do not consider, for instance, the value of arguments to the function. Our implementation is intended as a small proof-of-concept only.

6 EVALUATION
Our evaluation is in two parts. First, we microbenchmark each component of our proof-of-concept implementation described in Section 5. We then use these microbenchmarks, combined with real price data from CAISO and MISO, to provide a realistic parameterization of a system-level simulation.

6.1 Microbenchmarking
Next we present our microbenchmarks of function-level memoization and price prediction.

6.1.1 Function-level memoization. Figure 6 shows the latency of function calls with memoization, as compared to traditional compute. There are two sources of added latency from memoization: (1) the cost to check the cache for a precomputed result, and to return it if it exists; (2) the cost to store a computed result. We measured these values for our proof-of-concept implementation. The results, summarized in Table 2, were used to parameterize our system-level simulations.

6.1.2 Price predictor. We measured the performance of the price predictor on both CAISO and MISO data. Table 3 summarizes the results.

6.1.3 Task predictor. Our naive task predictor had a top-1 accuracy prediction accuracy of 46% for predicting the next 10 function calls given the previous 10. This means that each individual function in the predicted sequence has a 46% chance of being correct.

6.2 System-level simulation
We built a system-level simulator to explore the performance of Information Batteries at scale. Our simulator takes as input a time-series of energy prices, and a set of parameters that define the performance of individual components of the system (Table 1).

The output of the simulator is a simulated 100-day run of an IB system, reporting 1) cycles\textsubscript{avail}, the total amount of opportunity power that was theoretically available for compute, 2) cycles\textsubscript{op}, the amount of opportunity power that was actually used for compute,
and 3) cycles\textsubscript{grid}, the amount of grid power that was used for compute. The run is considered successful if the amount of grid power used for compute is less than what would have otherwise been used in a traditional computing system.

For each 5-minute time interval, the simulator decides (based on the provided traces and the specified performance of the price predictor), whether or not to anticipate a period of negative-priced power. If negative-priced power is anticipated, it “schedules” jobs. It then records the amount of grid and opportunity power “used” in the interval. We summarize the logic of the simulator in Figure 3.

Our initial simulation results highlight the importance of accurate price prediction (Figure 7), and low cache latency (Figure 8), and relatively accurate task prediction (Figure 5). We use the results from microbenchmarking to realistically parameterize our system-level simulator according to the metrics described in Table 1.

We then evaluate the overall effectiveness of Information Batteries according to: 1) the efficiency of the system, in terms of the amount of processing offloaded from grid power to opportunity power, and 2) dollar cost relative to traditional batteries.

6.2.1 Energy savings. A key goal of the IB system is to make use of opportunity power that might otherwise be wasted. Thus, a central metric for any IB system is the amount of processing power offloaded from non-opportunity power to opportunity power.

We measure these savings in terms of the number of machine cycles offloaded. Figure 3 shows these results for a wind-dominant power profile (MISO). We parameterize the simulator as follows: (1) Memoization overhead of 1 ms (2) Task prediction accuracy of 50%; (3) For the price predictor, false positive and false negative rates of 0.1%. The length of tasks is varied.

Note that IBs always incur more machine cycles than traditional compute. This is to be expected, since the IB system must perform the same computational task with additional pre- and post-processing. However, in these scenarios, the grid power consumed by the IB system is less than that for traditional compute, since a large fraction of the compute was offloaded to opportunity power.

The overall energy savings of the IB system is the difference between the grid power consumed in the traditional compute scenario and the grid power consumed by the information batteries.

6.3 Cost of storage

In some scenarios, Information Batteries are more cost-effective than traditional energy storage systems. First we explain how to think about the storage capacity of an IB system. Then we compare potential IB systems with conventional grid-scale battery storage.

Unlike conventional batteries, Information Batteries have both a “charge rate” measured in Watts, corresponding to the power draw at which precomputation can be done by a given IB system (limited by data center capacity)—and a prediction time horizon that is not present in ordinary batteries. In some ways this makes an IB system incommensurable with conventional batteries.

6.3.1 IB memoization overhead. Our function-level precomputation has an overhead of 34 μs for each put, 0.87 μs for each cache miss and 2.2 μs for each cache hit and retrieval on a 2.6 GHz Intel Core i7 CPU. This is extraordinarily efficient by virtue of its simplicity. For example, any job with just one second or longer run time would experience less than 1% overhead due to memoization.

6.3.2 Battery comparison. Consider for a moment a hypothetical IB system that has a one day time horizon, and can predict with perfect accuracy. The IB system can thus precompute a day’s worth of tasks. Consider a hyper-scale data center that is 100 MW with this one-day prediction horizon; an IB system in this data center could store a monumental 2400MWh via precomputation alone.

It is rare for a data center to have full-day lookahead. Instead, we might more realistically have, on average, 90 minutes prediction ability with 90% accuracy.6 With one-hour lookahead, such a data center could store 150 MWh, significantly more than most grid-scale battery-based storage projects. Given the negligible overhead of memoization, the key efficiency parameter is job prediction.

Using 90% as a canonical target efficiency, as it closely matches lithium-ion battery efficiencies, such an IB-enabled data center

---

6There is little public data on workloads and scheduling; our estimate here is based upon our experience working in such hyper-scale compute environments.
would match the storage capacity of a 150,000 kWh lithium-ion storage array, which, at current lithium-ion prices of $356 / kWh [38] would cost $53.4 million.

7 CONCLUSION
In this work we have shown that Information Batteries have the potential to provide a cost-effective means to cope with growing renewable intermittency using large-scale computing infrastructure. Key to the IB approach is that it is not a general-purpose solution, but is likely to be effective for many common workloads.

7.1 Future directions
This paper merely introduces and explores one avenue of implementation and evaluation of Information Batteries. Much remains to be studied. In particular, we believe that there are three lines of worthwhile future research on this topic: improved prediction, improved integration into large systems, and support for the precomputation and recomposition of fragments of computational tasks.

7.1.1 Prediction. The smart grid research community has done extensive work to improve prediction of price and power availability. In addition, the distributed systems community has done extensive work to characterize workloads in a wide range of settings. As we showed, as prediction accuracy improves, the efficiency of an IB system will improve, so there is substantial low-hanging fruit in incorporating state-of-the-art predictors.

7.1.2 Integration. While we frame our IB system prototype as an end-to-end system, any real-world deployment of this approach would necessarily omit the toy compute controllers we built for testing and instead integrate IB decision-making into an existing compute controller (e.g., a Kubernetes controller managing a whole data center). The criteria used by such data center operators to use the IB approach would necessarily be dependent upon their costs, business models, and the types of workloads they typically serve.

7.1.3 Computation. The efficiency of an IB system depends in large part on how well jobs can be accurately predicted and precomputed. But this precomputation need not be merely binary in nature—indicating whether a whole task should be precomputed or not—but instead can reflect a complex planning strategy that decomposes compute workloads into precomputable sub-units. There remains substantial work to be done on integrating ideas from related analyses performed in dramatically-different contexts, such as return-oriented programming, to identify which tasks can be meaningfully fragmented and then reassembled. In addition, efficient caching and retrieval of fragmented, precomputed results is challenging, as the greater complexity of fragmented precomputation, the more expensive retrieval is likely to be; this may require storage of program control-flow graphs along with compute fragments, so as to easily identify cached results that will meet the needs of new tasks. Finally, our exploration in this paper leveraged compiler support for program instrumentation, but in a real deployment it would be ideal to support unmodified program binaries.

REFERENCES


This paper presents machine learning approaches to detect cyberattacks on electric grid transformers. Specifically, the authors compare the effectiveness of multiple anomaly detection techniques to detect malicious tampering to transformers’ differential protective relays. The anomaly detection techniques use different types of autoencoder neural networks, where each focuses on a particular type of transformer fault. The authors train and validate their models using the D6 benchmark test system. The results show that their LSTM model outperforms the others on one-phase-to-ground and two-phase-to-ground faults. However, for three-phase-to-ground fault, the other models tend to detect anomalies better. The authors also evaluate their models’ performance on unseen anomalies and find that a Linear AutoEncoder model performs best. The reviewers though the authors addressed an important topic in security for smart grid systems, and rigorously validated their results using a standard benchmark test system.

Public review written by
David Irwin
University of Massachusetts Amherst, USA
Data Analytics for Cybersecurity Enhancement of Transformer Protection

MARTIYA ZARE JAHROMI, University of Toronto, Canada
AMIR ABIRI JAHROMI, University of Toronto, Canada
DEEPA KUNDUR, University of Toronto, Canada
SCOTT SANNER, University of Toronto, Canada
MARTHE KASSOUF, Hydro-Quebec Research Institute (IREQ), Canada

Electric power substations are experiencing an accelerated pace of digital transformation including the deployment of LAN-based IEC 61850 communication protocols that facilitate accessibility to substation data while also increasing remote access points and exposure to complex cyberattacks. In this environment, machine learning algorithms will play a vital role in cyberattack detection and mitigation and natural questions arise as to the most effective models in the context of smart grid substations. This paper compares the performance of three autoencoder-based anomaly detection systems including linear, fully connected, and convolutional autoencoders, as well as long short-term memory (LSTM) neural network for cybersecurity enhancement of transformer protection. The simulation results indicated that the LSTM model outperforms the other models for detecting cyberattacks targeting asymmetrical fault data. The linear autoencoder, fully connected autoencoder and 1D CNN further outperform the LSTM model for detecting cyberattacks targeting the symmetrical fault data.

CCS Concepts:
- Computer systems organization → Embedded and cyber-physical systems.
- Computing methodologies → Anomaly detection.

Additional Key Words and Phrases: cybersecurity, data analytics, machine learning, transformer protective relays

1 INTRODUCTION

The rapid integration of standard and interoperable information and communication technologies (ICT) in substations [12, 33] has accelerated the frequency and complexity of electric utility cyberattacks [13]. Attacks against electric power substations such as that on the Ukrainian grid in 2015 have caused significant societal and economic damage including loss of life [10, 28, 30]. As such, the North American Electric Reliability Corporation (NERC) has taken initial steps towards safeguarding cyber-assets by mandating the critical infrastructure protection (CIP) standards [1].

The emergence of standardized and interoperable communication protocols such as IEC 61850 and industrial internet of things (IIoT)-based applications renders traditional security-by-obscurity and perimeter defense security strategies obsolete [2]. Yet, these transformations facilitate accessibility to high fidelity substation data to lay a powerful groundwork for developing machine learning-based data analytics for cybersecurity enhancement [20]. Cybersecurity of substations has been analyzed in the literature from two perspectives: 1) cybersecurity risk assessment/impact analysis [7, 18, 26] and 2) cyberattack detection, mitigation and prevention [5, 15–17, 29]. Most of the approaches solely focus on information technology (IT) data. For instance, some of these approaches attempt to detect cyberattacks by examining the intruders’ footprints on the communication packets. This is while the cyberattack signatures on the operational technology (OT) data have been commonly neglected. This trend is expected to rapidly change in the coming years by the introduction of novel cyberattack detection systems that rely on both information technology (IT) and operational technology (OT).

Anomaly-based techniques have three main advantages over misuse-based techniques. First, anomaly-based techniques can adaptively learn the time varying dynamics and operating points of power systems to establish comprehensive baselines for system behaviors. Second, anomaly-based detection techniques only require training on normal (non-attack) data, which is available in abundance compared to cyberattack data making it possible to easily obtain the necessary training sets for model optimization. Third, anomaly-based techniques are capable of detecting unencountered zero-day cyberattacks. The primary disadvantage of anomaly-based techniques is the potential for high false detection rates because previously unseen system behaviors can be categorized as anomalies [4, 11, 34].

Machine learning-based anomaly detection systems have been extensively examined for cyberattack detection in smart grids. An artificial intelligence-based approach has been proposed in [23] to identify compromised meters. An intrusion detection system has been proposed in [3] for wide area measurements. An unsupervised anomaly detection system has been proposed in [21] to differentiate cyberattacks from disturbances and faults in smart grids. A machine learning-based method has been proposed in [6] to detect cyberattacks against state estimation. In [32], the margin setting algorithm has been employed to defend smart grids against false data injection attacks. Despite the considerable potential of machine learning-based anomaly detection systems, they have received less attention in the literature compared to analytical approaches for cybersecurity enhancement of substations due to the lack of high fidelity data in traditional substations. A 1-dimensional convolutional based autoencoder has been employed in [24] to identify cyberattacks against distance protective relays. A fully connected autoencoder has been employed in [19] to enhance the cybersecurity of the transformer differential protection. In [4], data analytics comprising long short-term memory neural network and ridge based regression classifier have been used to identify the root causes of the transmission protection mal-operation. Yet, different autoencoder-based anomaly detection systems for cybersecurity enhancement of protective relays have not been compared previously in the literature.
This paper compares the performances of different autoencoder-based anomaly detection systems as well as LSTM for cybersecurity enhancement of transformer protection. Specifically, we employ a variety of autoencoder-based anomaly detection systems as well as the LSTM neural network for cybersecurity enhancement of transformer protection using OT data. The performance of different autoencoder-based anomaly detection systems and the LSTM neural network for identifying different types of cyberattacks are measured and compared.

2 THE FALSE DATA INJECTION ATTACK AGAINST TRANSFORMER PROTECTION

Cyberattackers may target confidentiality, integrity, or availability (C-I-A) of data. Confidentiality aims to prevent users/devices from accessing unauthorized data. Integrity is about validity and correctness of data. Availability deals with the accessibility of data within a reasonable amount of time to an authorized user/device. Availability and integrity of data are paramount for OT systems like protective relays because they rely on real time data to identify abnormal conditions such as faults to actuate circuit breakers. Prominent examples of cyberattacks on the availability and integrity of data include distributed denial of service (DDoS) and false data injection (FDI) attacks, respectively.

The main protection of transformers are typically differential protective relays. Differential relaying monitors the currents entering and leaving a transformer calculating the geometrical sum (called the differential current) of the current phasors at all the terminals of the transformer. If the differential current takes a value of zero or a very small value due to measuring inaccuracies, the differential relay will stay inactive because the system is considered to either be operating normally or have an external fault. If the differential current takes a large value, it indicates an internal fault, within the scope of the relay, and the differential relay will trip the circuit breakers of the transformer [8, 22].

Moreover, the merging units receive the commands from the differential protective relay in the form of GOOSE packets through the process bus and send the trip signals to the circuit breakers after performing the digital-to-analog conversion.

An FDI attack is considered here where the intruder manipulates the magnitude and phase angle of the current measurements, yielding different elements of the differential protective relay to issue false tripping commands to the transformer circuit breakers. Multiple scenarios can be considered for the execution of the FDI attack against transformer protection including: 1) the installation of malicious firmware on the merging units through a supply chain attack or by physical access to the merging units, and 2) the injection of false data to the process bus through remote rogue connections using stolen legitimate substation operator credentials.

3 MACHINE LEARNING-BASED ANOMALY DETECTION SYSTEMS

The main objective of the proposed anomaly detection systems is to detect the malicious tampering of current measurements by an attacker to illegitimately trigger different elements of the differential protective relay of a transformer. The differential protective relay of a transformer is designed to detect different types of faults including three-phase-to-ground, two-phase-to-ground, and single-phase-to-ground. Therefore, we design and train a separate anomaly detection system for each type of fault. Each of the anomaly detection systems becomes operational by the activation of the corresponding element of the differential protective relay. We further investigate the possibility of considering a universal architecture for anomaly detection systems for different types of faults by comparing the architectures obtained for each type of fault.

The choice of machine learning models for anomaly detection depends on the nature and dimensionality of the input data. The input data to the anomaly detection system for differential protective relays is composed of two time series of three-phase current measurements. This results in high dimensionality of the input data and complicates feature extraction for machine learning. Moreover, the evolving and clandestine nature of cyberattacks limit the possibility of effective modeling of anomalous behaviour of cyberattacks in contrast to normal behaviour in substations for which there is significantly more data and more predictable characteristics. In this environment, semi-supervised and unsupervised machine learning approaches are in a superior position for cyberattack detection in contrast to supervised machine learning approaches. Advances in semisupervised and unsupervised machine learning has made it possible to solve classification problems, including anomaly detection, with high-dimensional data sets that can suffer from complex structure, sparsity or overfitting [25]. The autoencoder-based anomaly detection systems and LSTM neural network learn to compress the input data into a smaller latent space, then reconstruct the input data from the latent space with a low reconstruction error. Since we train the autoencoders and LSTM neural network with benign or attack-free current measurement sequences, we expect to observe high reconstruction error when feeding malicious current measurement sequences as input [31]. We define the reconstruction error...
for a data sequence $X_t$ as given in (1). A data sequence is considered anomalous if the reconstruction error is above a predefined threshold as given in (2).

$$MSE_t = ||X_t - M(X_t)||^2$$  \hspace{1cm} (1)

$$MSE_t > \epsilon \implies \text{anomalous data sequence}$$  \hspace{1cm} (2)

where $MSE_t$ is the Mean Squared Error, $X_t$ denotes the input data sequence, $M$ denotes the autoencoder or LSTM model, $M(X_t)$ denotes the output data sequence, and $\epsilon$ denotes the threshold considered on the reconstruction error for anomaly detection.

**Linear Autoencoders.** A linear autoencoder consists of an input layer, a code layer with a size smaller than input/output layers, and an output layer. In a linear autoencoder, all the activation functions in each layer are linear. The linear autoencoder model is similar to dimensionality reduction in Principal Component Analysis (PCA).

**Fully Connected Autoencoders.** In fully connected neural networks, all the neurons in each layer are connected to all the neurons in the subsequent layer. From a technical perspective, a fully connected autoencoder consists of two parts; an encoder and a decoder, as illustrated in Fig. 2. An encoder consists of an input layer, a variable number of hidden layers, and a code (embedding) layer. The code layer connects the encoder and decoder and its size is smaller than input and output layers. The decoder consists of the same number of hidden layers as the encoder and an output layer.

**1D Convolutional Neural Networks.** Unlike fully connected networks, neurons of each layer in CNN are not connected to all the neurons in the following layer and parameter sharing exists that reduces storage. 1D-CNN is a good candidate for anomaly detection because it is capable of detecting localized anomalies due to its window-based nature. Different types of layers are used in CNN autoencoders including convolutional, pooling, and upsampling [25, 27]. The convolution layer works based on convolution operation as given in (3).

$$S(i) = (X * f)(i) = \sum_{j=0}^{n} X(j)f(i-j)$$  \hspace{1cm} (3)

where $X$ denotes the input of the operation, $S$ denotes the output, $f$ denotes the convolution filter and $n$ denotes the length of the convolution filter.

In convolution layers, various filters are applied in parallel to the input to produce a set of linear activations. Each linear activation is followed by a non-linear activation function. To reconstruct the original input in the decoder, upsampling and convolution layers are combined. This combination is also known as transposed convolution or deconvolution. Fig 3 shows an example of an upsampling operation.

**Long Short-Term Memory Networks.** Other types of networks, called recurrent networks, consist of neurons that have self-connections or connections to neurons from previous layers. This recurrence provides the ability for the network to retain what happened in the past (short-term memory). The new state $h_t$ is expressed as:

$$h_t = f_w(h_{t-1}, x_t)$$  \hspace{1cm} (4)

where $x_t$ is the input vector at time step $t$, $h_{t-1}$ denotes the old state, and $f_w$ is a function with parameters $w$.

Consider Fig. 4 as a simple example of an RNN. Using the recurrent formula in (4) at each time step, we can process a sequence of vectors $X = \{x_1, x_2, ..., x_n\}$ using the same function $f$ and weights $w$ at every time step.

To address the vanishing gradient problem in RNNs, Long Short Term Memory (LSTM) networks have been designed. An LSTM network is a good candidate for anomaly detection because it is capable of detecting non-local, long term anomalies. Fig. 5 shows an LSTM unit. The horizontal line on top of the unit is responsible for passing the cell state which facilitates the long-term memory for relevant components of the data. LSTM unit consists of three gates. Forget gate is responsible for removing the parts of the cell state that are no longer needed. Input gate adds the information needed to the cell state. Output gate produces the output. It is possible to stack up arrays of LSTM units to enable more complex LSTM networks.

4 TRAINING, VALIDATION AND OPTIMIZATION OF THE ARCHITECTURE OF THE ANOMALY DETECTION SYSTEMS

In this section, we provide information about the test system and the training data set. We explain the approach employed for optimizing the architectures of the proposed machine learning-based anomaly detection systems. In addition, a set of metrics are presented for measuring the performance of the anomaly detection systems.

4.1 Test System

The IEEE power system relaying committee (PSRC) D6 benchmark test system is considered for generating the training data sets [14]. This test system connects a power plant with four 250 MVA generator units to a 230 kV transmission network through two parallel 500 kV transmission lines. The 230 kV transmission network is modeled as an infinite bus. Differential protective relays protect the power plant transformers as illustrated in Fig. 6.
4.2 Training Data Set

OPAL-RT HYPERSIM is employed to implement and simulate the PSRC D6 test system and generate the training data sets. The simulations are performed for a duration of 1.5 seconds with the fault start varying randomly between $t=1$ s to $t=1.02$ s to ensure the fault occurs at different parts of the current waveforms. Note that the period of one cycle is approximately 0.0167 s in a 60 Hz power system. Moreover, the generation levels are changed between 350 MW and 360 MW in 2 MW step size in each simulation to generate data sets under different operating conditions. The simulations are performed for different types of faults including three-phase-to-ground, two-phase-to-ground and single-phase-to-ground faults. The fault impedance is assumed to be zero. In total, 20,736 simulations are performed to generate training data sets for each type of fault. The anomaly detection systems are trained with 80% of the 20,736 simulations for different types of faults including three-phase-to-ground, two-phase-to-ground and single-phase-to-ground faults. The fault signatures.

The three-phase current measurements are collected from CT1 and CT2 at the sampling rate of 4800 samples per second in compliance with IEC 61850-9-2 standard for SV packet specifications [9]. An important parameter for training of the machine learning-based anomaly detection system is the input data length. A sliding window of 10 ms, i.e., 48 samples of current measurements per phase, is fed to the anomaly detection systems as input. As such, the input data to the anomaly detection systems contain $6 \times 48 = 288$ samples in total. In order to obtain the input data, we extracted a 20 ms window from each 1.5 s simulation containing 47 samples before the starting point of the fault and 47 samples after the starting point of the fault. Next, we slide the 10 ms input window of the anomaly detection systems over the 20 ms window of data sample by sample. This amounts to 48 windows of input data per simulation with 10 ms duration.

4.3 Optimizing the Architecture of the Machine Learning-Based Anomaly Detection Systems

We used the grid search method for hyperparameter tuning and optimizing the architecture. In this method, we consider values in Table 1 for each hyperparameter. Different possible combinations of hyperparameters are then tested using grid search. The best hyperparameter values are selected based on the validation error observed in the grid search.

The test data set includes 2074 simulation data sets. We replaced 207 of the test data sets, i.e., 10% of the test data sets, with FDI cyberattack data sets to create an imbalanced test data set. We use an imbalanced test data set because cyberattacks in power systems are rare events compared to normal behavior. The hyperparameters considered and tested for each model are listed in Table 1.

The FDI cyberattack data sets considered cover various situations ranging from naive scenarios where the cyberattacker only understands the principles of transformer differential protective relays to very sophisticated cyberattacks where the cyberattacker has some knowledge of power system dynamics and transformer fault signatures.

The cyberattack data are generated by OPAL-RT HYPERSIM. We considered three different scenarios for cyberattack data generation. In the first scenario, random false data are generated by OPAL-RT HYPERSIM with the appropriate magnitude to mimic a fault condition. In the second scenario, the tap setting of the current transformer in the test system are modified in OPAL-RT HYPERSIM.
such that transformer differential protective relay receives current measurements with larger magnitude, mimicking a fault condition. In the third scenario, a fault condition is simulated in OPAL-RT HYPERSIM and used as a replay attack.

4.3.1 Linear Autoencoder Architecture. We employed a fully connected autoencoder with one hidden layer, one input layer, and an output layer. All the activation functions in the model are linear.

4.3.2 Fully Connected Autoencoder Architecture. We used a fully connected autoencoder with the same number of hidden layers in encoder and decoder. In the encoder part, the number of neurons in hidden layers monotonically decreases from the input to the code layer. We used the Adam optimizer for model optimization. Fig. 7 shows the fully connected autoencoder architecture obtained for anomaly detection considering the current measurements triggering the three-phase-to-ground fault element of the differential protective relay. For the sake of brevity, the autoencoder architectures obtained for different types of faults are summarized in Table 2.

4.3.3 1D Convolutional Neural Network Architecture. In the 1D CNN architecture, the first layer is zero padding. In the encoder part, we use convolution and max-pooling layers. In the decoder part, there are deconvolution layers, a combination of upsampling and convolution layers. Max pooling and upsampling layers both have window sizes of 2. Filter size for convolution layers is tuned in the hyperparameter tuning step. Table 3 summarizes the details of the architecture obtained for anomaly detection. For conciseness, the hyperparameters selected for 1D CNN are summarized in Table 2.

4.3.4 LSTM Architecture. The many to many one direction LSTM network is considered with input is a sequence of 48 vectors of size 6. We considered stacks of LSTM unit arrays followed by a dense layer. The learning rate, number of LSTM layers, and the size of LSTM unit arrays are tuned in the hyperparameter tuning step and summarized in Table 2. Fig. 8 shows the LSTM architecture obtained for anomaly detection considering the current measurements triggering the two-phase-to-ground fault element of the differential protective relay. \( X_t \) represents the current sample at the time step \( t \), \( Y_t \) represents the output of time step \( t \). \( h \) represents the hidden state, and \( c \) represents the cell state.

The similarity of architectures obtained for anomaly detection systems for different types of faults in Table 2 indicates that a universal architecture can be possibly designed for different types of faults.

5 SIMULATION RESULTS

The performance of the anomaly detection systems are measured using precision and recall metrics, which are more appropriate for imbalanced datasets. It is worth noting that the accuracy metric is not helpful because cyberattacks are rare events. The correct selection of the threshold value plays a vital role on the performance.

5.1 Performance Analysis of Autoencoder-Based Anomaly Detection Systems

The performance of the linear autoencoder, fully connected autoencoder, 1D convolutional autoencoder, and LSTM are measured for detecting cyberattacks against different elements of the transformer differential protective relay. We use the precision-recall curve to understand the performances of the four models for different possible thresholds.

### Table 2. Selected Parameters for The Proposed Models. (A: One-Phase-To-Ground Faults, B: Two-Phase-To-Ground Faults, C: Three-Phase-To-Ground Faults)

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear AE</td>
<td>Learning rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Code Size</td>
<td>40</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Fully Connected AE</td>
<td>Learning rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Depth</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Code Size</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>1D CNN</td>
<td>Learning rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Depth</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Convolution filter size</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Filter count</td>
<td>48</td>
<td>28</td>
<td>48</td>
</tr>
<tr>
<td>LSTM</td>
<td>Learning rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>Depth</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>LSTM units count</td>
<td>40</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

### Table 3. 1D Convolutional Network Architecture for Anomaly Detection in One-Phase-To-Ground Fault Measurements

<table>
<thead>
<tr>
<th>index</th>
<th>layer type</th>
<th>Output Dimensions</th>
<th>parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input</td>
<td>Length: 288</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Zero Padding</td>
<td>Count: 1</td>
<td>pad size = 3</td>
</tr>
<tr>
<td>3</td>
<td>Convolution</td>
<td>Length: 292</td>
<td>filter size = 6</td>
</tr>
<tr>
<td>4</td>
<td>max pooling</td>
<td>Count: 164</td>
<td>window size = 2</td>
</tr>
<tr>
<td>5</td>
<td>Convolution</td>
<td>Count: 146</td>
<td>filter size = 6</td>
</tr>
<tr>
<td>6</td>
<td>max pooling</td>
<td>Count: 73</td>
<td>window size = 2</td>
</tr>
<tr>
<td>7</td>
<td>Convolution</td>
<td>Count: 73</td>
<td>filter size = 6</td>
</tr>
<tr>
<td>8</td>
<td>up sampling</td>
<td>Count: 146</td>
<td>window size = 2</td>
</tr>
<tr>
<td>9</td>
<td>Convolution</td>
<td>Count: 146</td>
<td>filter size = 6</td>
</tr>
<tr>
<td>10</td>
<td>up sampling</td>
<td>Count: 292</td>
<td>window size = 2</td>
</tr>
<tr>
<td>11</td>
<td>Convolution</td>
<td>Count: 292</td>
<td>filter size = 6</td>
</tr>
<tr>
<td>12</td>
<td>up sampling</td>
<td>Count: 292</td>
<td>window size = 2</td>
</tr>
<tr>
<td>13</td>
<td>cropping</td>
<td>Count: 288</td>
<td>size = 288</td>
</tr>
</tbody>
</table>

### Table 4. Performance of the Anomaly Detection Systems. (A: One-Phase-To-Ground Faults, B: Two-Phase-To-Ground Faults, C: Three-Phase-To-Ground Faults)

<table>
<thead>
<tr>
<th></th>
<th>Linear AE</th>
<th>Fully-connected AE</th>
<th>1D CNN</th>
<th>LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Prec</td>
<td>0.990</td>
<td>1</td>
<td>0.995</td>
<td>1</td>
</tr>
<tr>
<td>A Rec</td>
<td>0.995</td>
<td>1</td>
<td>0.995</td>
<td>1</td>
</tr>
<tr>
<td>B Prec</td>
<td>0.991</td>
<td>1</td>
<td>0.991</td>
<td>1</td>
</tr>
<tr>
<td>B Rec</td>
<td>0.995</td>
<td>1</td>
<td>0.995</td>
<td>1</td>
</tr>
<tr>
<td>C Prec</td>
<td>0.991</td>
<td>1</td>
<td>0.989</td>
<td>1</td>
</tr>
<tr>
<td>C Rec</td>
<td>0.991</td>
<td>0.986</td>
<td>1</td>
<td>0.969</td>
</tr>
</tbody>
</table>
The LSTM model outperforms the other three models for the one-phase-to-ground and two-phase-to-ground faults as illustrated in figures 9 and 10. However, the linear autoencoder, fully connected autoencoder and 1D CNN approximately have similar curves and outperform the LSTM model for the three-phase-to-ground fault as illustrated in Fig. 11. Table 4 summarizes the results when the threshold is selected such that the precision is equal to 1. It is worth noting that even a subtle change in the performance of anomaly detection systems for protective relays is significant because the misoperation due to cyberattacks has the potential to cause major disturbances and widespread blackouts in power systems. Given the symmetry of three-phase-to-ground faults and asymmetry of single-phase-to-ground and two-phase-to-ground faults, we conclude that LSTM performs better for asymmetrical faults and is weaker than the other models for symmetrical faults. We feel that such a trend will generalize to other more complex systems beyond the benchmark system employed in this paper because of the inherent ability of LSTM to recognize time series patterns and manage long-term memory patterns in contrast to the other models.

### 5.2 Impact of Data Granularity on Anomaly Detection Performance

In this case study, we investigate the impact of generation level granularity on the performance of each of the machine learning algorithms while considering the three-phase-to-ground-fault. Thus, test data sets are generated for the three-phase-to-ground-fault with finer generation level granularity compared to the training data set, i.e., the generation levels are changed with 1 MW step size in each simulation. Next, we measured the performance of the linear autoencoder, fully connected autoencoder, 1D convolutional autoencoder and LSTM for detecting cyberattacks while considering cyberattack data and data of finer generation level granularity that have not been considered in the training step.

Table 5 summarizes the results when the threshold is selected such that the precision is equal to 1. The comparison between the results in Table 4 and Table 5 show that the performance of all four models significantly drops when they are exposed to data captured from other generation levels that are not included in the original dataset. Yet, the linear autoencoder model outperforms the three other models for one-phase-to-ground and two-phase-to-ground faults. The linear autoencoder, fully connected autoencoder and 1D CNN further outperform the LSTM model for the three-phase-to-ground fault. The impact of input data granularity on the performance of the deep learning-based anomaly detection systems is further investigated using a sensitivity analysis. The results showed that the performance of the all four models significantly drop when they are exposed to the previously unseen system behaviors. Yet, the linear autoencoder model outperformed the three other models when it is exposed to the previously unseen system behaviors.

### 6 CONCLUSION

This paper presented four machine learning-based anomaly detection systems including linear autoencoder, fully connected autoencoder, convolutional autoencoder, and LSTM neural network for cybersecurity enhancement of transformer differential protection for anomaly detection in transformer relays. The simulation results underscore the fact that the LSTM model outperforms the other models for one-phase-to-ground and two-phase-to-ground faults. The linear autoencoder, fully connected autoencoder and 1D CNN further outperform the LSTM model for the three-phase-to-ground fault. The impact of input data granularity on the performance of the deep learning-based anomaly detection systems is further investigated using a sensitivity analysis. The results showed that the performance of the all four models significantly drops when they are exposed to the previously unseen system behaviors. Yet, the linear autoencoder model outperformed the three other models when it is exposed to the previously unseen system behaviors.

### REFERENCES


---

#### Table 5. Performance of The Anomaly Detection Systems While Considering Unseen Data with Finer Granularity.

<table>
<thead>
<tr>
<th>Linear AE</th>
<th>Fully-connected AE</th>
<th>1D CNN</th>
<th>LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.84</td>
<td>1</td>
<td>0.8</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.58</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 9. Precision-recall curve for one-phase-to-ground faults
Fig. 10. Precision-recall curve for two-phase-to-ground faults
Fig. 11. Precision-recall curve for three-phase-to-ground faults


Public review for

The Next Stage of Green Electricity Labeling: Using Zero-Knowledge Proofs for Blockchain-based Certificates of Origin and Use

Johannes Sedlmeir, Fabiane Völter, Jens Strüker

This is a timely article addressing the problem of verifying the Guarantees of Origin for green energy generation. A potential solution is proposed based on blockchain architecture in combination of cryptographic zero-knowledge proofs, which improves the transparency and privacy of green energy labelling in generation and consumption. However, the paper only presented rather high-level ideas. There are still missing some system details and a security analysis of the solution. It is hoped that a more complete study and a proof-of-concept implementation will be provided to demonstrate the practical value of the blockchain solution.

Public review written by

Sid Chi-Kin Chau
Australian National University, Australia
The Next Stage of Green Electricity Labeling: Using Zero-Knowledge Proofs for Blockchain-based Certificates of Origin and Use

JOHANNES SEDLMIEIR, FIM Research Center, University of Bayreuth, Germany
FABIANE VÖLTER, Project Group Business & Information Systems Engineering of the Fraunhofer FIT, Germany
JENS STRÜKER, Fraunhofer Blockchain Lab, University of Bayreuth, Germany

The labeling of electricity is considered an important mechanism to differentiate renewable power generation and, thus, to incentivize the expansion of green energy. However, today’s systems for documenting and trading green energy certificates suffer from multiple challenges. These could be addressed by a digital solution that holistically collects and processes production and consumption data. Blockchain-based architectures have repeatedly been suggested for this purpose since they can provide transparency and can likely be accepted by a broad group of stakeholders. Yet, there are significant scalability and privacy issues of a blockchain-based approach for storing and processing fine-grained production and consumption data. In this paper, we propose and discuss a potential solution that leverages succinct cryptographic zero-knowledge proofs to balance the required level of transparency and privacy while at the same time providing a high degree of scalability.

ACKNOWLEDGMENTS
This work is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) based on a resolution of the German Bundestag (funding code: 03E16026A). We thank Matthias Babel, Alexander Bogensperger, Dennis Jelito, Timon Rückel, Benjamin Schellinger, Nils Urbach, Andreas Zeiselmair, Till Zwede, and our research partner Stiftung Umweltenergieeich for their valuable support.

1 INTRODUCTION
To meet the Paris climate targets and ultimately achieve a 100% renewable energy system, many researchers consider the electrification of the heating and traffic sector as mandatory [Hansen et al. 2019]. Moreover, a consensus prevails that so-called green or renewable hydrogen that is produced with green electricity will be required for industrial process heat in the future, irrespective of its extent [van Renssen 2020]. In general, owing to the increasing need for sector coupling and electrification, overall electricity consumption is elevating [Fridgen et al. 2020]. However, it is controversial where the necessary additional renewable energy will come from and, thus, how to close the so-called green electricity gap [Bloomberg New Energy Finance 2018]. As covering 100% of future electricity use with renewable energy seems to be out of reach for a decade at least [Diesendorf and Elliston 2018], differentiating between the degree of “green” is vital for both renewable energy producers and consuming enterprises striving for CO₂ reductions [Comello et al. 2021]. Besides the ecological aspect, this is mainly due to the necessity of verifiable reporting [Financial Times 2021; Sullivan and Gouldson 2012] since shareholders increasingly expect enterprises to disclose the amount of CO₂ emitted through their operations [Hefron 2021]. Furthermore, there is increasing regulation that requires verifiable reporting; for instance, the planned extension of emission trading systems to the traffic and building sector in the European Union will require the holistic documentation of emissions [Reuters 2021]. Businesses are thus increasingly demanding verifiably green products [Whelan and Kronthal-Sacco 2019]. Moreover, they have strong incentives to differentiate from competitors through a lower carbon footprint in order to fulfill their customers’ desire for carbon-neutral products [Palacios-Argüello et al. 2020]. The latter also reflects in the increasing demand of green energy tariffs by residential consumers [Herbes and Ramme 2014].

The special physical properties of electricity pose a challenge for a consistent labeling required for verifiable reporting. The tracking or even modification of electricity flows through the grid to ensure that a specific consumer receives “physically” green energy is complex and hardly practical or even impossible at scale. Rather, a common method is to approach the challenge from an economic perspective by somehow “offsetting” the energy consumed, i.e., decoupling the physical flow of electrical energy from its commercialization: Energy producers using renewable energy facilities may claim “certificates of origin” for every unit of renewable energy produced. Stakeholders can then trade these certificates separately from the physical energy. Upon consumption by a consumer relying on a renewable energy tariff, certificates are cancelled out [Morthorst 2003]. The act of cancellation hence indirectly corresponds to a “proof of use”. Such systems exist in Europe, where they are called guarantees of origin (GOs), the U.S., and Asian countries [Hamburger 2019]. Nevertheless, existing certificate-based solutions for labeling electricity suffer from significant challenges. For example, they often do not accurately represent carbon emissions and lack transparency and verifiability for end-consumers. This results in susceptibility to fraud and low consumer trust [de Chalendar and Benson 2019; Hamburger 2019].

Several studies have highlighted the suitability of blockchains for the documentation and trading of GOs in order to overcome the existing challenges: Rather than centralized systems, they provide a high degree of transparency by design and have the capability to unite stakeholders on a single, neutral platform [e.g. Albrecht et al. 2018; Castellanos et al. 2017; Knirsch et al. 2020; Richard et al. 2019]. Yet, the replicated data storage and processing of a blockchain represent a
hurdle regarding privacy and scalability requirements in general [Munilla Garrido et al. 2021; Zhang et al. 2019] and specifically for a sophisticated labeling system that involves millions of generators’ and consumers’ personal or business-sensitive information [Alt and Wende 2020]. On the other hand, we observe that innovative solutions in decentralized finance have started to address privacy and scalability issues through succinct cryptographic zero-knowledge proofs (ZKPs). In this paper, we hence propose a design how to address challenges of electricity labeling with blockchain technology and ZKPs. We propose and evaluate an architecture for a verifiable, scalable, and privacy-oriented electricity labeling system that can help promote the use of renewable energy. Thus, we also aim to contribute to research on “green IS” demanded by Watson et al. [2010] and Goebel et al. [2014].

The remainder of this paper is structured as follows. We first briefly introduce the concepts of blockchain technology and ZKPs as the technology stack for our architecture in section 2. Next, we review challenges of labeling in general and GOs in particular and summarize related work in section 3. After that, we present the design of our proposed labeling solution in section 4. We then explore some characteristics of our approach in more detail in section 5 and discuss associated challenges and opportunities. We conclude by pointing out avenues for future research in section 6.

2 TECHNICAL BUILDING BLOCKS

2.1 Blockchain Technology

Blockchains are generally defined as a particular type of electronic ledger where data is replicated across multiple servers (“nodes”) in a peer-to-peer network. This physically and organizationally decentralized yet logically centralized data management is achieved through an append-only structure in which selected nodes batch and order transactions into blocks that reference the corresponding previous block through a hash-pointer [Butijn et al. 2020]. Changing a single bit in any transaction, or their order, would render the chain of hash-pointers in an inconsistent manner. Accordingly, tamper sensitivity and the subsequent ease to detect fraud are among the distinct properties of a blockchain. The eligibility to append a block and the decision-making about which transactions to include and in which order is decided decentrally through a so-called consensus mechanism [Xiao et al. 2020]. Provided a specific threshold of the network (in some consensus-related metric) is honest, there are firm guarantees regarding retrospective immutability, non-censorship, and the correct execution of transactions. In general, consensus mechanisms combine cryptographic tools such as one-way functions, verifiable pseudo-random functions, and digital signatures and are heterogeneous in terms of security assumptions, latency, finality, and energy consumption [Kannengießer et al. 2020; Sedimeir et al. 2020; Xiao et al. 2020]. Besides public, permissionless blockchains where anyone can participate, there are also permissioned blockchains where participation is limited, e.g., to a consortium from industry or the public sector [Amend et al. 2021; Wüst and Gervais 2018].

So-called smart contracts extend the functionality of blockchains beyond their initial purpose in applications, namely simple payments in cryptocurrencies like Bitcoin, to the execution of Turing-complete programming logic. Smart contracts are scripts that are redundantly executed and hence cross-checked by all nodes to make sure that the result is correct [Buterin 2013]. Consequently, blockchain technology facilitates general-purpose digital platforms while avoiding dependencies on one or a few distinct entities’ availability or honesty [Alt 2020]. This avoidance of a single point of failure from an integrity and availability perspective makes blockchains highly attractive for critical infrastructures and the digital collaboration of mutually distrusting organizations [Fridgen et al. 2019]. Consequently, blockchains have, among others, been suggested in the energy sector for decentralized storage and control in power grids, peer-to-peer energy trading in smart grids, imbalance settlement, electric vehicle charging, e-roaming, carbon emission and green certificate trading, and fine-grained investments through tokenization [Albrecht et al. 2018; Andoni et al. 2019; Bao et al. 2020; Gorenflo et al. 2019; Wang et al. 2021].

Nonetheless, blockchain adoption faces many challenges. The deliberate redundancy of blockchain transaction storage and operation leads to scalability challenges as every node needs to process all other nodes’ transactions [Gudgeon et al. 2020; Sedimeir et al. 2021a; Zhou et al. 2020]. Moreover, the open availability of the same data to all blockchain nodes leads to a degree of information exposure that can be problematic [Kannengießer et al. 2020; Platt et al. 2021; Zhang et al. 2019]. Examples are conflicts with confidential corporate information and anti-trust regulation, as well as data protection regulation that inhibits the storage of sensitive customer data [Schellinger et al. 2022; Tatar et al. 2020]. The immutability and in particular lack of deletion capabilities on blockchains further exacerbate issues with data protection regulation [Rieger et al. 2019]. Consequently, first innovative solutions to scalability and privacy-related issues of blockchains have appeared, many of which are based on cryptographic techniques and specifically ZKPs.

2.2 Zero-Knowledge Proofs

The notion of ZKPs was first introduced by Goldwasser et al. [1989]. ZKPs are a special form of interactive protocols between a so-called “prover” and a “verifier” in which the prover wants to convince the verifier about a specific statement. ZKPs have the additional property that the prover learns nothing beyond the truth of this statement. Depending on the precise definition of what constitutes “knowledge” and consequently “learn”, there are many nuances like perfect, statistical, and computational zero-knowledge. ZKPs can be “succinct”, i.e., the proof size and the computational complexity of proof verification are at least exponentially smaller than the complexity of the original computation.

However, besides a few special applications such as anonymous credentials [Camenisch and Lysyanskaya 2001] that allow for the selective disclosure of attributes in a digital certificate without revealing the value of the signature, practical implementations or even applications of ZKPs remained rare until recently. This was probably owing to the high computational complexity of proof generation and the lack of convenient programming languages or libraries to implement ZKPs. A period of rapid improvements starting from Groth et al. [2006] led to the first implementations of practical and general purpose ZKP libraries, e.g., in Pinocchio [Parno et al. 2013]. In recent
years, different flavors have emerged, such as succinct hybrid arguments of knowledge (SNARKs) [Gennaro et al. 2013] and scalable transparent arguments of knowledge (STARKs) [Ben-Sasson et al. 2019] that differ in cryptographic security assumptions and setup. They all have in common that they allow to create succinct proofs for the correct execution of computations without the need to reveal inputs or intermediate steps. Often, hashes, Merkle proofs, or signature checks for the inputs are a part of the program to force provers to commit to using variables that are unknown to most parties but fixed or that have specific required properties. Domain-specific languages such as Bellman and starkjs allow to compile rather general programs into arithmetic circuits, and libraries such as libsnark and circom translate these circuits into proving and verification programs. As generic tools for creating and verifying ZKPs have matured and improved in ease of use, performance, and scope over the last years, ZKPs have appeared in blockchain applications for the verification of off-chain computations’ (that would previously have been executed in a replicated way in a smart contracts) integrity without revealing sensitive data, starting with the privacy-oriented cryptocurrency Zcash [Sasson et al. 2014; Schellinger et al. 2022]. Regarding the applicability in the energy sector, Wang et al. [2021] propose ZKPs for privacy-preserving energy storing.

One important application associated with blockchains that heavily inspired our architecture are zk-rollups. A so-called operator or aggregator, who may be a single service provider or a consortium, collects users’ signed cryptocurrency transactions and maintains an associated book of accounts. This book of accounts is either stored on-chain, with domain-specific optimizations regarding storage and computation, or off-chain (in this case, the zk-rollup is sometimes called “Validium”). In case the book keeping occurs entirely off-chain, only the book of accounts’ Merkle root is stored on-chain. Any change of the state of this hash that represents the book of accounts’ state needs to be legitimized by the aggregator through a ZKP that the owner of the respective account authorized the transaction with their signature – a so-called “validity proof” [Buterin 2021]. Consequently, the operator does not need to be trusted regarding the correctness of the book keeping, because the validity of each modification is redundantly verified by all nodes [Gluchowski 2019]. By “compressing” many transactions into a single update of the book of accounts and a corresponding succinct ZKP (or sometimes just a succinct non-interactive proof that is not necessarily zero-knowledge) that proves the legitimacy of the update, zk-rollups such as Aztec, Hermez, Loopring, StarkDex, and zk-Sync that run as applications on the public Ethereum blockchain already allow for a significant boost in throughput compared to the base layer today [Schaffner 2021].

3 RELATED WORK
3.1 Guarantees of Origin
GOs were introduced in 2001 in the European Union to inform end consumers about the share of green energy in their consumed electricity mix [European Union 2009]. This, in turn, aims to incentivize households and companies to make decisions that minimize their carbon emissions when choosing an electricity supplier or tariff, and, therefore, to advance investments in renewable energy generation [Hamburger 2019; Morthorst 2003]. In practice, however, the design of existing GO systems is coming with a handful of downsides. First, they are often designed as national electronic registries in which GOs are traded for each generated MWh independently of their generation place and time [Will et al. 2017]. Utilities can buy GOs to label non-renewably produced electricity and sell it as green on their consumers’ bills [Bogensperger and Zeiselmaier 2020]; a practise that is often criticized as “greenwashing” [Johns 2021; Will et al. 2017]. For example, under the German register of guarantees of origin (GOR), GOs can stem from an arbitrary place and be “consumed” within one year. This disentanglement of GOs with the physical reality can reduce consumers’ trust [Hamburger 2019], resulting in indifference [Bogensperger and Zeiselmaier 2020; Hanemann et al. 2015; Jansen and Seebach 2009; Jochem et al. 2015] and a low willingness to pay premiums for green energy tariffs [Jansen 2017]. Second, low resolutions in time discourage time-specific consumption according to the current supply of green energy and do not accurately reflect the carbon emissions caused by the power grid [de Chalendar and Benson 2019]. Consequently, they fail in their goal to incentivize required investments, for example, storage facilities [Bogensperger and Zeiselmaier 2020]. Third, the current practices cause considerable in-transparencies regarding the feed-in of green energy and can lead to a gap between a country’s disclosed energy mix and its actual consumption [Kaenzig et al. 2013]. Fourth, the lack of a transparent process and the fragmentation of documentation also open the potential for fraud, such as the double-counting of green energy through not properly removing or invalidating certificates upon usage [Correctiv 2021; Hamburger 2019]. Thus, a certificate of use is not bound to a certificate of origin, which hinders consumers from verifying that the electricity they consumed was green. Consequently, there is a strong need for an end-to-end and internationally harmonized digital process [Bogensperger and Zeiselmaier 2020; Hamburger 2019]. Finally, the Paris climate agreement has led to many commitments to decarbonization goals by enterprises. In addition to the electrification of transportation and industrial heat, more and more manufacturers are interested in actively managing production processes according to the “greenness” of electricity in the grid. While stakeholders like transmission system operators (TSOs) could provide the carbon emissions associated with the current electricity mix as a service, they also have to cope with the results on the grid level that more short-time purchases have on consumption habits [Strüker et al. 2021].

The lack of a “global” perspective, insufficient harmonization between countries [Hamburger 2019], and the low degree of transparency [Knirsch et al. 2020] are considered primary causes of consumers’ lack of ability to verify whether suppliers actually bought a sufficient number of GOs for the sum of their sales. Further, the increasing cross-border trade of energy and the demand for international carbon certificates require a system for digitally verifiable and interoperable GOs. Article 15(9) of the Renewable Energy Directive already expects member states to “recognise GOs issued by other Member States” [European Union 2018, p. 119]; however, to date, varying legal and regulatory frameworks, as well as different progress of the digital transformation in the electricity sector across Europe pose a challenge for cross-border trading [Jackson et al. 2018]. Cross-border labeling requires the synchronization of registries among involved countries and trust in the correctness of data.
3.2 Existing solution approaches

To overcome these issues, all involved stakeholders could agree on a centralized platform (for example provided through a European institution) that is responsible for the aggregation and verification of GO-related data. However, it is often difficult to agree on a central trusted party within federal ecosystems, and the reliance on a single institution bears the risk of manipulation, aggregation of market power, or being compromised by an attack. For these reasons, among others, leveraging blockchain technology to provide a neutral platform has often been suggested [e.g. Albrecht et al. 2018; Castellanos et al. 2017; Knirsch et al. 2020; Richard et al. 2019; Velazquez Abad and Dodds 2020]. For example, Castellanos et al. [2017] simulate blockchain-based green certificate trading and, thus, focus on a market-based solution. However, the authors do not discuss scalability and neglect privacy and data sovereignty issues related to blockchain-based marketplaces such as the de-anonymization of users [Béres et al. 2020]. Diniz et al. [2021] present a blockchain-based architecture to facilitate reporting processes. The authors address the high complexity involved in reporting processes and aim to improve information flows, yet they do not discuss scalability requirements. To mitigate the sensitive information exposed on the blockchain, they propose to record certificate information only in encrypted form. However, trading patterns recorded on blockchains can allow for de-anonymization [Liu et al. 2021]; and by encrypting generation- and consumption-related information, the desired transparency of accounting is compromised. Knirsch et al. [2020] highlight the importance of verifiability for end-consumers and suggest a decentralized and permissionless system for the trading and verification of GOs. To prevent the necessity of trust in a central player, the authors suggest not to rely on distribution system operators (DSOs) or TSOs as natural intermediaries to issue green certificates. Producers and consumers can then trade GOs directly on a blockchain. The authors suggest to conduct audits of generation as well as consumption units to detect sources of wrong information. However, consumers have to rely on producers for cancelling certificates upon consumption. Furthermore, while providing mechanisms for the transparent verification of GOs, the authors neglect privacy-related aspects: Their approach only provides pseudonyms for customers, which has already proven to be insufficient to conceal identities in the long-term in cryptocurrencies [Biryukov et al. 2014]. Furthermore, the authors aim to ensure scalability by solely storing the hash reference of transactions. This approach is unlikely sufficient for an internationally-scalable solution: Considering the 195 Mio. private households in Europe, for example, would require about 2,260 transactions being processed per second on average in case households update their consumption solely on a daily basis. A daily update still does not provide high timely granularity; and yet, public permissionless blockchain systems such as the Ethereum blockchain are limited to only about 15 tx/s [Schäffer et al. 2019]. In addition, transaction fees have increased significantly over the last years, representing an additional burden for generators as well as consumers. In comparison to Knirsch et al. [2020], Karakashev et al. [2020] focus on protecting producers’ and consumers’ sensitive information. The authors explicitly focus on delivering a privacy-preserving solution by making use of ring signatures and stealth addresses. However, their approach does not ensure global verifiability as total production and consumption cannot be transparently retraced.

Thus, while previous authors have argued for the benefits of blockchain-based solution approaches for trading GOs, they have not yet sufficiently focused on aligning privacy, data sovereignty, and scalability requirements. Consequently, to the best of our knowledge, to date no proposed approach balances out transparency and verifiability with privacy and data sovereignty for all stakeholders and at the same time achieves the scalability that is necessary for an international GO system that operates close to real-time.

3.3 Requirements of Labeling Systems

Our analysis of the challenges of existing approaches to electricity labeling (see also Table 1 in the supplements for a summary) allowed us to derive a set of requirements that energy labeling systems need to fulfill to overcome. First, labeling systems need to prevent the double-counting of green energy and allow for verification across borders. In turn, this aspect requires end-to-end verifiability starting from the very source of data generation. Furthermore, a platform for GOs needs to be open, support the harmonization of existing solutions, and may not discriminate against the participation of stakeholders bound to different legislations, data processing paradigms, or other parameters. At the same time, no personally identifiable information or otherwise sensitive data, such as utilities’ and electricity producers’ business secrets, may be accessible to third parties that have no legitimate need to see it. Also, the platform needs to scale in terms of a high resolution in time and space to allow for fine-grained GOs that reflect the regional and temporal specificities of electricity generation and consumption, even with the expected increase in the number of generation assets to be sustainable in the long run. Lastly, labeling procedures need to be automated end-to-end and should involve as little manual interaction as possible. These requirements can be summarized as verifiability, privacy, openness and scalability.

4 PROPOSED ARCHITECTURE

We propose an alternative architecture to electricity labeling that addresses the requirements outlined in section 3.3. First, in line with the literature presented, we suggest using blockchain technology as an underlying neutral and non-proprietary infrastructure to facilitate a decentralized electricity labeling system. This building block allows for transparency and enables stakeholders to verify transactions independently. Second, to strike a balance between verifiability and privacy, our solution lever ZKPs. Last, to meet the requirement of scalability, we further make use of the succinctness of ZKPs. Thus, we design an interoperable, scalable system that prevents fraudulent or incorrect accounting by design while maintaining the required information privacy for all participants. We will illustrate our approach using an example of n generators, one utility, and m consumers. We also start with only two labels (green and grey electricity) and postpone the discussion of how to implement additional discrete labels and spatial origin to future work.

Prerequisites. As a foundational layer in our process, electricity generators and consumers need to establish an authenticated communication channel to reliably send their generation and consumption data
to their electricity utility. To do so, in line with Knirsch et al. [2020], we propose relying on a conventional public key infrastructure (PKI) or newer developments like self-sovereign identity (SSI) [Sedimeir et al. 2021b] for providing a secure bilateral communication layer and certificate-based proofs of the authenticity of sensor data. In specific, generators can digitally sign the produced quantities of electricity at its source to make the data trustworthy. As such, the proposed solution relies on data logging modules integrating crypto-chips and digital certificates that are currently being rolled out in many countries in the same way as previous propositions including Knirsch et al. [2020] and Karakashev et al. [2020] discussed. However, this is not an indispensable prerequisite for our approach: End users can check whether their data is referenced truthfully by the system by readily comparing their meter reading with the information in their (verifiable) electricity bill, and audits on the side of generators can be conducted. Nonetheless, cryptographically verifiable generation checks can happen by default rather than through costly or tedious manual effort. Moreover, using digital certificates, assets can prove their generation attributes automatically on request, which makes new generation facilities’ onboarding easier. For example, assets can prove that they have been attested the attribute of “renewable wind energy” by their certified manufacturer or another entity such as DSOs or TSOs that are considered trustworthy regarding this claim. Consumers’ devices also periodically digitally sign their consumption data with the private key in their digital meter or data logging device and communicate it to the utility.

**Bootstrap a utility.** In the beginning, a utility, which acts as the operator or aggregator that creates proofs of correct accounting (and, thus, proofs of origin and use) in our design, creates a large Merkle tree [Merkle 1987]. Initially, all leaves contain an agreed-upon, public value (e.g., 0). The number of leaves should be significantly larger than the maximum number of accounts that the utility expects to manage at any time. For example, for a utility that usually holds a million user accounts, we would suggest a depth of 22 and, thus, \( N = 2^{22} \approx 4.2 \) million leaves, which corresponds to the maximum number of accounts that can be managed. As such, the proposed architecture is of fixed size: all involved stakeholders must agree beforehand how many consumers and generators can at maximum be reflected in the process. While dynamically extending the Merkle tree during operation is technically feasible, it would significantly complicate the designed system, so we will not consider this in this work. The Merkle tree is locally stored by the utility and serves as a record for accumulated generation and consumption data. Any leaf in the Merkle tree represents either a consumer or a generator and contains the aggregated consumption or generation since their onboarding. The left half of the Merkle tree will contain consumers; the right half will contain generators.

**Onboarding process and data collection.** Now, a consumer can be onboarded by the utility as follows: The consumer’s digital meter establishes a bilateral, end-to-end encrypted communication channel with the utility and sends a request of being onboarded, together with a presentation of a digital certificate that proves that the metering device is certified. The presentation includes the metering device’s public key for signing consumption data and the (signed) current meter reading (“balance”). Consecutively, the utility replaces one empty leaf in the right half of the Merkle tree by the sensor’s public key (which is essentially a pseudonym for the consumer), the current meter reading, the timestamp of account creation, and an array initialized with zeros that represents the amount of green and grey energy that the consumer has received during the contract period so far (see also Figure 2 in the supplementary material). Similarly, generation units can be onboarded. If there has been a consumer or utility request to de-register, for instance, because they want to switch their energy provider, accounts have to be “offboarded”. This involves overwriting the previously described account details in the corresponding leaf of the Merkle tree with the initial publicly known value (e.g., 0) so that the leaf is prepared to be re-used in a later onboarding process. Based on consumption and generation data, the utility can locally update the leaves periodically (in discrete steps, which we call “epochs” that may be, e.g., 15 minutes), incorporating new data communicated by generation or consumption units’ sensors during the respective epoch. In specific, the utility’s locally stored Merkle tree serves as a running record for accumulated labelled generation and consumption data. To do so, utilities need to maintain local data registries for recording all account balances and accumulate this generation and consumption data. The symmetry in treatment of generators and consumers also allows for a rather flexible change of roles, which is necessary for so-called prosumers to efficiently participate in the system.

**Proving local integrity.** So far, the described architecture essentially corresponds to how a utility would handle their bookkeeping today while the data is managed in an unusual data structure of a Merkle tree. In any stage of the process, the Merkle tree that includes transaction data is kept entirely off-chain. The Merkle tree and differences of subsequent states of the Merkle tree represent fine-granular and, thus, highly sensitive consumption and generation data that must not be distributed to other stakeholders. For example, fine-grained consumption data can be used to derive information on a consumer’s habits or location [Hinterstocker et al. 2017]. In a first step, the utility can increase transparency and give consumers and auditors some control on what it is doing by updating the Merkle root of its off-chain accounting data periodically on a blockchain. Every consumer and generator can then verify that the data from their metering device has been included truthfully by requesting their historic account states. For example, a presentation of a digital certificate that proves that the metering device is certified, the presentation includes the metering device’s public key for signing consumption data and the (signed) current meter reading (“balance”). Consecutively, the utility replaces one empty leaf in the right half of the Merkle tree by the sensor’s public key (which is essentially a pseudonym for the consumer), the current meter reading, the timestamp of account creation, and an array initialized with zeros that represents the amount of green and grey energy that the consumer has received during the contract period so far (see also Figure 2 in the supplementary material). Similarly, generation units can be onboarded. If there has been a consumer or utility request to de-register, for instance, because they want to switch their energy provider, accounts have to be “offboarded”. This involves overwriting the previously described account details in the corresponding leaf of the Merkle tree with the initial publicly known value (e.g., 0) so that the leaf is prepared to be re-used in a later onboarding process. Based on consumption and generation data, the utility can locally update the leaves periodically (in discrete steps, which we call “epochs” that may be, e.g., 15 minutes), incorporating new data communicated by generation or consumption units’ sensors during the respective epoch. In specific, the utility’s locally stored Merkle tree serves as a running record for accumulated labelled generation and consumption data. To do so, utilities need to maintain local data registries for recording all account balances and accumulate this generation and consumption data. The symmetry in treatment of generators and consumers also allows for a rather flexible change of roles, which is necessary for so-called prosumers to efficiently participate in the system.

The analogous construction holds for a generation unit that sends its generation data to the utility.

**Proving global integrity.** This combination of the on-chain Merkle root and the Merkle proof is not yet sufficient for a certificate of origin or use: While the blockchain transaction receipt and Merkle proof allow for local integrity checks from the side of consumers and generators already, so far, there is no way to check whether the
generation and consumption data for green and grey electricity match globally at any time and, thus, whether or not certificates have been double-used. As only the utility has access to all generation and consumption data and, thus, a global overview of total generation and consumption, an additional mechanism is necessary to exclude manipulations on the utility’s behalf. Against this backdrop, we propose relying on ZKPs. Specifically, the utility must prove the legitimacy of any (batched) update of the Merkle tree that they perform on the blockchain, i.e., once per epoch. This involves that proofs for three different types of operation need to be given:

1. Proving the correctness of the on- and offboarding procedures.
2. Confirming that account balances have been updated according to the signed consumers’ and generators’ sensor data.
3. The total sum of generation and consumption match for each label.

Requirement (3) represents the core requirement for globally correct accounting under the condition that the local accounting is correct. Accordingly, the utility aims to prove that the total amount of energy sold to consumers is not larger than the total generation for each label and epoch. To facilitate labeling and prevent the “greenwashing” of energy, this process involves not only proving the matching of the total sum of generation and consumption but also the matching of energy types: The sum of green consumption when updating the Merkle tree may never exceed the sum of green production. At the same time, the sum of grey consumption may also never exceed the sum of grey production.

As outlined in section 2.2, ZKPs can be used to convince another party of the correctness of some mathematical statement without providing any further information. Applying this to the underlying context, the utility aims to prove the legitimacy of the Merkle tree updates to all involved stakeholders. As such, the utility seeks to prove the correctness of every update of the Merkle root that happens on-chain. To generate the ZKPs, exemplified with SNARKs, all stakeholders initially must agree on a proving and verification key generated from public code representing all conditions that the proof must satisfy by a key generation algorithm. This involves a so-called trusted setup that can be conducted as multi-party computation, so that the presence of one honest party ensures the correctness of ZKPs checked with the verification key [Bowe et al. 2018]. To calculate the ZKP, the utility uses public inputs (previous and suggested updated Merkle root), a private input or “witness” (the whole Merkle tree and the signed updates from the consumers and generators that authorize their on-/offboarding or include their updated balances), and the generated proving key. On this basis, the utility executes a proving algorithm that uses the proving key to generate a proof attesting that the utility knows a correct witness to the proposed update. Thus, the utility can prove that it updated the Merkle root correctly according to pre-defined conditions by revealing publicly only the previous and updated Merkle root and the ZKPs that contains no more information than the correctness of the updating procedure according to the conditions set in the public algorithm from which the proving and verification key were computed. In particular, the ZKP demonstrates that the updated Merkle root was computed from a Merkle tree representing leaves with the property that the cumulative increase in consumption balances for each label does not exceed the corresponding cumulative increase in generation balances.

To make this proof of global integrity accessible and, thus, verifiable to all relevant stakeholders, the utility sends the generated ZKP and the updated Merkle root to a smart contract, which can verify the ZKP with the help of the verification key and the previous Merkle root. If the ZKP is considered valid by the majority of blockchain nodes (in the metric corresponding to the consensus mechanism), the Merkle root is updated in the smart contract accordingly. This approach allows any stakeholder to verify that the global accounting has been conducted correctly. Consequently, the combination of local proofs of integrity using Merkle proofs with ZKPs attest the correct global accounting and in particular prevent double usage.

To summarize, in our proposed architecture, utilities receive all meter readings from generators and consumers, which resonates with familiar responsibilities regarding data availability and protection in today’s power systems. Consumers and generators still need to trust utilities regarding data privacy as of today when they communicate their generation and consumption data for billing purposes. The fundamental change in introducing the ZKP-based labeling is that consumers and auditors need to trust the utility solely with data privacy but not with the integrity of the accounting. We illustrate the overall architecture of our proposed solution in figure 1.

5 DISCUSSION

Technological Readiness. The proposed architecture relies on building blocks that are already in productive use in decentralized finance, for example, in the form of zk-rollups [Schäffer et al. 2019]. In parallel to the underlying use case, payments require a certain level of trust in the operator regarding the protection of users’ privacy, but the need for trust in the operator’s accounting needs to be avoided for acceptance reasons. To achieve the former, current payment solutions like StarkWare [2021] also rely on off-chain accounting and batching of transactions to achieve scalability requirements. What’s more, the energy sector has less rigid requirements regarding data availability than the use cases of existing zk-rollup implementations. This is mainly related to on-chain data availability. It is likely not essential to recover an account if the utility crashes or loses data in the case of electricity: re-starting the process through onboarding all entities will likely be sufficient as this use cases focuses on ensuring that in each epoch no more green energy was consumed than generated. By contrast, in the case of digital assets, losing the state of the book of accounts is problematic because users cannot reconstruct the respective part of the Merkle tree to claim their locked funds. Thus, we expect that the proposed solution suits the underlying use case very well. Moreover, as we argued in section 4, the responsibilities for data collection and evaluation in our design reflect the status quo in the energy sector well.

On the other hand, there are some requirements of our use case which exceed those of existing use cases. First, we need very high throughput for close-to-real-time accounting, because every single consumer will need a transaction every 15 minutes or so. Additionally, it may be more difficult to parallelize the processing of transactions as there will not always be a pair of transactions that guarantees constant supply, as in the case of payments. However, we expect
that partial updates should be feasible by building clusters of generating and consuming units when the number of accounts is large. Besides, questions of error handling, e.g., in the case of delayed communication of sensor data, need to be addressed.

**Complexity & Costs.** The proving and verification complexity of ZKPs have decreased significantly over the last years due to many theoretical advances and performance optimizations. For example, when using Poseidon as a hashing mechanism [Grassi et al. 2021], verifying the computation of a single hash contributes around 300 constraints; a metric for the complexity of proof generation. Thus, proving that an update from a previous to a new Merkle root is legitimate when every single account is updated involves re-computing both Merkle roots from the leaves, which corresponds to 2\cdot 2N hashes for the trees that we illustrate here. Checking that the accumulated consumption of green energy in an epoch is lower than the accumulated generation (and potentially the correct range of the transactions’ timestamp) only adds a small number of additional constraints. From ongoing projects on the public Ethereum blockchain and our own first tests with the circom, snarks and rapidsnark libraries, we know that an operator (i.e., the utility) with dedicated hardware can prove approximately $2^{27} \approx 128$ million constraints in two minutes with a 64-core server [Hermez Network 2021]. Without any optimizations, this would allow the management of around 100,000 accounts and updates of each of them every two minutes. Looking at the fast developments in the last years, including the appearance of new solutions such as STARKs, which do not involve a trusted setup, and the utilization of graphics processing units (GPUs), we expect that proofs for Merkle trees involving millions of accounts can be created in a reasonable time with common enterprise hardware and, thus, at low costs in the near future.

Regarding costs, the verification of a Groth16-SNARK currently costs around 200,000 gas on the public Ethereum blockchain, plus around 9,400 gas per public input [Eberhardt 2021]. In our architecture, the only inputs for the ZKP are the root hashes of the utility’s previous and updated Merkle tree, so the gas costs stay below 250,000 gas, which is $\frac{1}{20}$ of the current public Ethereum block capacity. For comparison, a simple payment on Ethereum consumes 21,000 gas, while more complex transactions in decentralized finance are on the order or 100,000 gas. Yet, the costs of verifying a SNARK to a utility would still be considerable today: around 50 USD using gas prices from October 2021, which would amount to almost 2 million USD per utility and year. With a block time of around 13 seconds, if a utility has their SNARK verified by a smart contract every 15 minutes, this occupies around 0.02 % of the total Ethereum capacity. Given a potentially large number of utilities, this still seems impractical. Initially, we thus recommend to move to a permissioned architecture, the only inputs for the ZKP are the root hashes of the utility’s previous and updated Merkle tree, so the gas costs stay below 250,000 gas, which is $\frac{1}{20}$ of the current public Ethereum block capacity. For comparison, a simple payment on Ethereum consumes 21,000 gas, while more complex transactions in decentralized finance are on the order or 100,000 gas. Yet, the costs of verifying a SNARK to a utility would still be considerable today: around 50 USD using gas prices from October 2021, which would amount to almost 2 million USD per utility and year. With a block time of around 13 seconds, if a utility has their SNARK verified by a smart contract every 15 minutes, this occupies around 0.02 % of the total Ethereum capacity. Given a potentially large number of utilities, this still seems impractical. Initially, we thus recommend to move to a permissioned blockchain that is operated by several organizations in the energy sector; this does not involve gas costs and allows for a significantly larger number of SNARKs to be verified per time period as higher hardware requirements allow for better performance [Sedlmair et al. 2021a]. Moreover, since the ZKP in our architecture only has two public inputs, aggregated verification techniques through recursion [Bowe et al. 2020] or batching [Gailly et al. 2021] can be used to significantly further reduce the ZKP verification effort: In each epoch, utilities can send their SNARK to an untrusted aggregator that creates a ZKP that attests the correctness of many utilities’ SNARKs and that is exponentially less costly to verify than the whole of individual SNARKs (public inputs, however, contribute linearly). This approach

![Fig. 1. Overall architecture of our labeling platform.](image_url)
further enhances the scalability in system with many utilities and indicates practicability also with a permissionless blockchain in the future.

Double-spend Prevention for Generators. Our architecture design makes sure that the double-use of green electricity certificates is not possible, as consumers can verify whether their own consumption data is reflected in the on-chain Merkle root and whether the utility does the global accounting correctly. Signed smart meter data further increases authenticity guarantees. However, so far, our design is still vulnerable to generation units that fraudulently register at different utilities and subsequently submit their sensor data attesting the generation of a certain amount of green energy to multiple utilities. Thus, mechanisms to prevent the double-commercialization of generated labeled electricity are needed. A promising approach beyond spot-checks by auditors or regulators that could integrate well with the proposed blockchain-based approach could involve a token that a generation unit receives on its installation by the local grid operator. Generating units could then transfer their “registration token” to a specific utility. The utility then needs to additionally prove in their ZKP that they only included generation data from units that had transferred their registration token to this utility. Future research could compare this and alternative approaches regarding their complexity and the level of authenticity that they can provide.

Trading. Considering the trading only on the level of utilities and energy producers already significantly reduces the complexity of electricity trading and, thus, makes it easier to detect the double-use of GOs both for centralized and decentralized trading platforms. For a decentralized platform on a permissionless blockchain – an approach that several papers like GECKO have suggested [Knirsch et al. 2017] – GOs would be traded as tokens. This is possible in combination with our approach, where utilities not only publish the Merkle root of their generation and consumption database but additionally the difference between consumption and generation for each label. Since this data is already highly aggregated over many generators and consumers, it exhibits significantly less privacy and scalability issues than consumers’ and generators’ individual data. Nonetheless, this information may still not be appropriate for publishing on a blockchain because net balances for each label and epoch can be sensitive business data and may even be problematic regarding antitrust regulation. Consequently, it may be important to add a ZKP-based level of privacy and scalability also on this level, where the aggregate consumption and generation or their difference can be referenced as public information or private inputs to further ZKPs. In the latter case, this could directly integrate with constructions as suggested by Karakashev et al. [2020], which — as we argued in section 3.2 — may be problematic to introduce on the level of small producers because it is very difficult to find compromised sensors that fraudulently generate GOs in an anonymized system, but which may be suitable to apply to the business-to-business (B2B) trading layer. Essentially, the cryptographically verifiable aggregation below the utility layer in our approach aims to mitigate the privacy and scalability challenge for GOs on a decentralized infrastructure without a significant tradeoff in risk mitigation. We believe that our architecture can be used as a scalable, privacy-oriented, decentralized and, thus, neutral base layer for aggregate GO trading and documentation on a B2B layer.

6 CONCLUSION AND OUTLOOK

In this paper, we identified the requirements of verifiability, privacy, openness and scalability for the documentation of GOs and proposed a technical architecture that builds on blockchain technology and ZKPs to solve these challenges. We provide transparency and verifiability by design by storing Merkle roots and verifying ZKPs on-chain with no access restrictions for stakeholders. Since on-chain data does not reveal any information itself but only the adherence to ex-ante-defined data processing rules, we protect all involved stakeholders’ privacy and data sovereignty by design, too. Lastly, succinct ZKPs allow for a future increase in the number of production facilities and a high time-resolution of their production and end consumers’ consumption despite blockchains’ limited performance. Future research can consider introducing additional labels to further distinguish between renewable sources of energy to reflect consumer preferences beyond carbon emissions. This may not only include discrete labels but also continuous labels, for example, to facilitate regional GOs. In addition, researchers can analyze to what extent our proposal is applicable also to related areas that require transparency and that need to prevent double-counting by design while ensuring scalability and the privacy of the involved stakeholders. In particular, we suggest the trading of carbon emissions as a promising area in which the combination of blockchains, ZKPs, and potentially PKI- and certificate-based digital identities for assets and similar ideas will be useful. Moreover, future research could address the trading of labelled energy tokens to allow utilities to purchase additional green energy to satisfy consumer demand.

Our research is bound to several limitations. While we have already fully designed our labeling platform’s roles and information flows and started with the implementation, we still lack a practical evaluation of the architecture in a field test. This also includes an encompassing security model and analysis. For example, assets’ digital identity certificates could be used for access management if a zero-trust paradigm in terms of security is chosen [Buck et al. 2021]. Furthermore, collecting experiences regarding users’ trust in a blockchain- and ZKP-based labeling mechanism also is essential: Winther and Ericson [2013] already demonstrated that Norwegian consumers mistrust GOs not only owing to unreliable methodology but also because physical electricity is untraceable. Further research could thus also analyze whether the well-known “trust machine” blockchain will help convince consumers of the integrity guarantees and tamper-resistance of the proposed solution [Völter et al. 2021].

In sum, the lack of verifiable GOs with a high temporal and spatial granularity represents a critical barrier to popularizing green energy products and is required for an efficient computation of carbon footprints and associated certificates. We believe that the proposed architecture provides a transparent, accurate, scalable, and privacy-preserving alternative to current GO practises and existing blockchain-based architectures that have been proposed. Further, we conclude that a base layer of digital identities for organizations, natural persons, and things, combined with new and potentially decentralized platforms for the verifiable exchange of data and privacy-enhancing cryptographic solutions, may be a promising approach to address many of the challenges both in and beyond the energy sector.
Table 1. Overview of problems of current GOs registries and sources (supplementary material).

<table>
<thead>
<tr>
<th>#</th>
<th>Challenge</th>
<th>Specifics</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Temporal/spatial decoupling</td>
<td>Allows to label grey energy as green (&quot;greenwashing&quot;)</td>
<td>[Will et al. 2017], [Johns 2021]</td>
</tr>
<tr>
<td>2</td>
<td>Temporal/spatial decoupling</td>
<td>Indifference and minimal user acceptance</td>
<td>[Hamburger 2019], [Bogensperger and Zeisel-mair 2020], [Hunimann et al. 2015], [Jansen and Seebach 2009], [Jochem et al. 2015], [Jansen 2017]</td>
</tr>
<tr>
<td>3</td>
<td>Temporal/spatial decoupling</td>
<td>Low willingness to pay premiums for green electricity tariffs</td>
<td>[Will et al. 2017], [Johns 2021]</td>
</tr>
<tr>
<td>4</td>
<td>Intransparency of feed-in quota</td>
<td>Gap between disclosed energy mix and actual consumption, potential for fraud</td>
<td>[Kaenzig et al. 2013], [Correctiv 2021]</td>
</tr>
<tr>
<td>5</td>
<td>Non-automated processes</td>
<td>Complexity and high costs for suppliers and regulators</td>
<td>[Bogensperger and Zeisel-mair 2020]</td>
</tr>
<tr>
<td>6</td>
<td>Limited scalability</td>
<td>Long-term sustainability regarding increase in decentralized energy production facilities not expected</td>
<td>[Bogensperger and Zeisel-mair 2020], [Bañales 2020]</td>
</tr>
<tr>
<td>7</td>
<td>Verifiability</td>
<td>Difficulties to detect and prevent double-counting</td>
<td>[Hamburger 2019]</td>
</tr>
<tr>
<td>8</td>
<td>Lack of cross-border harmonization</td>
<td>(EU) countries` registries are not synchronized</td>
<td>[Hamburger 2019], [Jackson et al. 2018]</td>
</tr>
</tbody>
</table>

Fig. 2. Account structure of the Merkle tree based accounting.
Public review for

Privacy-Preserving Energy Storage Sharing with Blockchain and Secure Multi-Party Computation

Nan Wang, Sid Chi-Kin Chau, Yue Zhou

This paper proposes a novel blockchain-based solution to ensure privacy and secure transactions for peer-to-peer energy storage sharing within a community. The multi-layered solution is based on integrating zero-knowledge proof and multi-party computation protocol in an Ethereum blockchain network and is demonstrated on a small case-study. The paper was selected for publication in this journal as it addresses a highly interesting solution for an important issue that has been missing in peer-to-peer storage sharing literature until now, and as it was well written and presented in a convincing manner. The authors replied to all concerns of the reviewers. For example, they clarified that their system can verify if there is any dishonest user, but it cannot identify an individual dishonest user, because it is fundamentally impossible to identify this in multi-party computation with a dishonest majority of users. It was also mentioned that the suggested scheme is rather complex, and it was questioned whether it can be scaled up to be adapted at reasonable costs. This discussion, however, goes beyond the scope of the given paper and is left for future work.

Public review written by

Anke Weidlich

University of Freiburg, Germany
Privacy-Preserving Energy Storage Sharing with Blockchain and Secure Multi-Party Computation

NAN WANG, Australian National University
SID CHI-KIN CHAU, Australian National University
YUE ZHOU, Australian National University

Energy storage provides an effective way of shifting temporal energy demands and supplies, which enables significant cost reduction under time-of-use energy pricing plans. Despite its promising benefits, the cost of present energy storage remains expensive, presenting a major obstacle to practical deployment. A more viable solution to improve the cost-effectiveness is by sharing energy storage, such as community sharing, cloud energy storage and peer-to-peer sharing. However, revealing private energy demand data to an external energy storage operator may compromise user privacy, and is susceptible to data misuses and breaches. In this paper, we explore a novel approach to support energy storage sharing with privacy protection, based on privacy-preserving blockchain and secure multi-party computation. We present an integrated solution to enable privacy-preserving energy storage sharing, such that energy storage service scheduling and cost-sharing can be attained without the knowledge of individual users’ demands. It also supports auditing and verification by the grid operator via blockchain. Furthermore, our privacy-preserving solution can safeguard against a majority of dishonest users, who may collude in cheating, without requiring a trusted third-party. We implemented our solution as a smart contract on real-world Ethereum blockchain platform, and provided empirical evaluation in this paper.

Additional Key Words and Phrases: Privacy-Preserving, Energy Storage Sharing, Blockchain, Secure Multi-Party Computation

1 INTRODUCTION

Energy storage can buffer energy in a storage medium, which is useful for temporal shifting of energy demands and supplies. In addition to absorbing excessive renewable energy, energy storage can effectively reduce the consumption cost under dynamic time-of-use (ToU) energy pricing plans by storing energy during off-peak periods and discharging during peak periods. But the present cost of energy storage systems remains considerably expensive. Energy storage also incurs significant maintenance cost over time, with only limited life cycles. There is a possibility of a future technological breakthrough that may significantly reduce the current cost of energy storage in the near future. Hence, despite its benefits, the current users are reluctant to immediately adopt energy storage at a wide scale. However, rather than postponing the use of energy storage, a more viable solution to improve the cost-effectiveness of present energy storage is by sharing energy storage among multiple users or out-sourcing to a third-party energy storage operator. In fact, time-sharing and out-sourcing have been popular concepts, particularly in cloud computing. Energy storage may also become an out-sourcible resource in a similar fashion.

Currently, there are multiple possible paradigms of energy storage sharing. First, in community sharing [28], a group of local users, who do not own individual energy storage, can connect to a shared energy storage facility. The shared energy storage will be utilized by the users based on a coordination mechanism. The associated cost will be split among the users in a fair manner. Second, a non-local third-party energy storage operator can provide an outsourcing service as cloud energy storage [30]. The energy storage operator can offset the energy consumption of remote users by exporting energy from its energy storage facility. Third, the users, who have their own energy storage, can pool their energy storage resources together to support each other in peer-to-peer sharing [9].

All of these energy storage sharing paradigms can be effectively supported by the notion of virtual net metering (VNM) [18, 37], which is a flexible bill crediting system for transferring the credits or debits of a user’s energy account to another, even though they do not share the same physical metering infrastructure. By VNM, energy storage operators can possibly transfer the credits of their energy export to offset the debits of energy import of other users. VNM has been used to enable community solar energy sharing in practice [19]. It can also enable energy storage sharing among a group of geographically distributed users and energy storage operators.

Although sharing can improve the cost-effectiveness, there is a heightened concern of user privacy nowadays. Users may need to disclose private energy demand data to a third-party energy storage operator in order to schedule the use of shared energy storage. This may reveal sensitive personal data (e.g., working patterns, number of occupants, and vacation periods). Potential misuses and breaches of personal data may lead to serious undesirable consequences. To bolster user privacy, stricter privacy protection legislations are being introduced in various countries to restrict personal information revelations to a third-party (e.g., GDPR in Europe). Because of these privacy concerns and privacy-related legislations, we are motivated to ensure proper privacy protection in energy storage sharing with a third-party operator.

In this paper, we introduce the concept of “privacy-preserving energy storage sharing”, by which a third-party energy storage operator should be given only minimal information for its energy storage service operations without being able to compromise personal data for other unintended purposes. But the key challenge is how to design an effective solution to enable proper energy storage service scheduling and cost-sharing among users, without the knowledge of individual users’ energy demands, and yet that can still be verified and audited to eliminate any fraud.

We provide a feasible solution to enable privacy-preserving energy storage sharing, by drawing on several recent technologies.
First, blockchain (e.g., Bitcoin, Ethereum) is a disruptive paradigm that enables decentralized verifiable applications without trusted intermediaries by integrating a tampering-resistant ledger with a distributed consensus protocol. Blockchain is an effective platform to support transparent energy storage sharing and auditable VNM with grid operators. But blockchain by default does not ensure privacy, and transaction data is entirely disclosed on the ledger. Recently, there is a new trend of supporting privacy on blockchain. For example, there are privacy-oriented cryptocurrencies, like ZCash, Monero [4, 34], that utilize zero-knowledge proofs for privacy-preserving digital asset management, without revealing them. In this paper, we utilize privacy-preserving blockchain to conceal the private data in cost-sharing and VNM for energy storage sharing.

Second, secure multi-party computation (or simply called multi-party computation) has been a subject of extensive research [13], which provides a general framework to allow multiple parties to jointly compute a function while concealing the private inputs. Recently, efficient multi-party computation protocols based on secret-sharing (e.g., SPDZ) have been applied to many practical applications like privacy-preserving machine learning [10]. In this paper, we apply multi-party computation to energy storage service scheduling with concealed individual users’ demands. Moreover, we integrate multi-party computation with privacy-preserving blockchain to support confidential cost-sharing and verifiable VNM settlement.

In summary, this paper presents an integrated solution to enable privacy-preserving energy storage sharing in all the stages, as outlined in the following (and also illustrated in Figure 1):

(a) Multi-party Computation for Energy Storage Scheduling: First, the users can compute their aggregate day-ahead demands by secure multi-party computation, without revealing individual demands. Then, they can derive the optimal energy storage service schedule subject to energy storage service constraints.

(b) Privacy-preserving Cost-sharing Payment: The users can split the cost of energy storage service based on a fair cost-sharing scheme in a privacy-preserving manner. The users can make energy storage service payments via privacy-preserving blockchain, without disclosing individual transactions. After receiving the payments, the energy storage operator will issue verifiable receipts on blockchain ledger.

(c), (d) Energy Storage Operation & Virtual Net Metering Settlement: The users and energy storage operator will follow the energy storage service schedule. They do not need to exchange energy directly, and the energy flows through the grid. They will settle their energy accounts via VNM. With verifiable receipts on blockchain ledger, the users can offset their energy consumption by the energy export from energy storage, which will be audited by the grid operator.

Particularly, we should ensure privacy protection throughout the integrated process of scheduling, cost-sharing, payment and auditing, without requiring a trusted third-party. While it may be easier to ensure privacy in individual processes separately, it is challenging to ensure privacy in the integrated process. For example, one can schedule a service, or make a payment separately in a privacy-preserving manner. However, it is harder to verify the payment with respect to the scheduled service with privacy protection.

Furthermore, privacy also poses a significant challenge to the correctness and integrity of operations. Because of concealing their demands, dishonest users may attempt to cheat by paying less to energy storage service or claim more in VNM than what they ought to. These dishonest users may even collude to coordinate their actions in cheating. Hence, it is critical to safeguard against the presence of dishonest users. Remarkably, our privacy-preserving solution is able to safeguard against a majority of dishonest users (namely, more than 50% of users may be dishonest).

This paper is organized as follows. We first review the related work and background in Section 2. We then formulate the problem and models in Section 3, and present the basics of cryptographic components and multi-party computation in Sections 4-5. The privacy-preserving solution is presented in Section 6. We next provide an empirical evaluation of our implementation on Ethereum blockchain platform in Section 7. We also discuss several extensions as well as the limitations of our solution in Section 8. We conclude this work with future work in Section 9.

2 RELATED WORK AND BACKGROUND

2.1 Energy Storage Sharing

Optimizing energy storage under dynamic pricing plans has been a popular research topic [17, 22, 33]. Recent studies proposed various paradigms for energy storage sharing among multiple users, for instance, cloud energy storage [30], virtual community sharing [28] and peer-to-peer sharing [9]. Notably, there are many studies about privacy in smart grid in other aspects. For example, [24, 26] employed energy storage to hide private consumption behavior by mixing random energy storage charging and discharging to mask the consumption patterns. [38] presents privacy-preserving data aggregation for smart meters that aggregates users demands. None of these studies addressed the privacy aspect in energy storage...
sharing. To the best of our knowledge, this is the first paper to address the issue of privacy-preserving energy storage sharing and its cost-sharing.

2.2 Virtual Net Metering
To enable energy consumers to share physically disconnected energy storage from energy storage operators, one can rely on Virtual Net Metering (VNM) [18, 37] for transferring the credits or debits of a user’s energy account to another. When energy consumers import energy from the grid, they will incur debits in their energy accounts. On the other hand, when energy storage operators export energy to the grid, they will earn credits in their energy accounts based on feed-in tariffs. However, simultaneous exporting energy from energy storage operators and importing energy from energy consumers with the same amount of energy should be able to offset each other.

In practice, the credits of energy export of energy storage operators may be transferred to offset the partial debit of energy import of energy consumers. In this case, it may not require simultaneous energy export and energy import in VNM. However, maintaining instantaneous energy balance at VNM is still important to ensure the stability in power distribution network. While there may be additional costs in power distribution network such as power transmission cost, balancing the energy generation and consumption should be the major component in VNM. Note that VNM is entirely an account balancing process, without the need to configure the energy flow in the power distribution network. VNM has been proposed to enable novel applications, such as transactive energy transfer in an energy exchange market, community solar energy and shared energy storage [19].

2.3 Blockchain Technology
There is an increasing number of applications of blockchain technology to energy systems. For example, the study [21] applied blockchain to mitigate trust in peer-to-peer electric vehicle charging. Blockchain has been applied to microgrid energy exchange and wholesale markets by prosumers [32]. Renewable energy credits and emissions trading are also applications of blockchain [25]. In these applications, the goal of blockchain is to improve transparency and reduce settlement times, since blockchain system can ensure integrity and consistency of transactions and settlement on an open ledger. See [1] for a recent survey about blockchain applications to energy systems.

Note that none of these studies have considered the privacy on blockchain, even though the transaction data on the ledger is entirely disclosed to the public. Our work is one of the first studies to explicitly address privacy in blockchain applications of energy systems. Supporting privacy on blockchain is a crucial research topic in cryptography and security. There have been several privacy-preserving blockchain platforms with support of privacy (e.g. ZCash, Zether [4, 6, 34]). Our work draws on similar concepts from privacy-preserving blockchain, but also integrates specifically with the application of energy storage sharing, for example, to support auditable VNM. Our solution is implemented as a smart contract on permissionless Ethereum blockchain platform, but it can also be implemented on a permissioned blockchain platform.

2.4 Privacy-Preserving Solutions
We briefly survey and compare various approaches of privacy-preserving solutions in the literature. There are two major approaches: (1) data obfuscation that masks private data with random noise, (2) secure multi-party computation that hides private data while allowing the data to be computed confidentially. Differential privacy [16], a main example of data obfuscation, is often used in privacy-preserving data mining to extract certain data properties in a relatively large dataset. There is an intrinsic trade-off between the accuracy and privacy of differential privacy. On the other hand, secure multi-party computation [15, 20] traditionally employed garbled circuits [23] and homomorphic cryptosystems [12, 31], which have a high computational complexity. Recently, information-theoretical secret-sharing (e.g., SPDZ [13, 14]) has been utilized for secure multi-party computation, which provides high efficiency. This work employs secure multi-party computation for privacy-preserving energy storage operation scheduling and cost-sharing computations without disclosing private energy demands.

3 MODELS AND FORMULATION
In the following, we first formulate the energy storage sharing model without considering privacy. In the subsequent sections, we will incorporate privacy protection in the model.

3.1 Problem Setup
First, we describe several key components in the model (and list some key notations in Table 1):

<table>
<thead>
<tr>
<th>N</th>
<th>Total number of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uᵢ</td>
<td>The i-th user</td>
</tr>
<tr>
<td>pₜ(t)</td>
<td>Energy price of time-varying pricing scheme at timeslot t</td>
</tr>
<tr>
<td>Bₜ(t)</td>
<td>Capacity of energy storage at timeslot t</td>
</tr>
<tr>
<td>pₙₑ</td>
<td>Per-unit service fee of energy storage at each timeslot</td>
</tr>
<tr>
<td>hₜ(t)</td>
<td>State-of-charge of energy storage at timeslot t</td>
</tr>
<tr>
<td>εₑ, εₐ</td>
<td>Charging and discharging efficiency ratios</td>
</tr>
<tr>
<td>rₑ, rₐ</td>
<td>Charging and discharging rate constraints</td>
</tr>
<tr>
<td>xₑ(t)</td>
<td>Charging rate from the grid to the energy storage</td>
</tr>
<tr>
<td>xₐ(t)</td>
<td>Discharging rate from the energy storage to Uᵢ</td>
</tr>
<tr>
<td>yₑ(t)</td>
<td>Uᵢ’s residual consumption rate from the grid at t</td>
</tr>
<tr>
<td>yₐ(t)</td>
<td>Total residual consumption rate of all users at timeslot t</td>
</tr>
<tr>
<td>Costₑss</td>
<td>Total cost of energy storage service</td>
</tr>
<tr>
<td>Costᵢ</td>
<td>Uᵢ’s partial original cost without energy storage service</td>
</tr>
<tr>
<td>Pᵉₛₙ</td>
<td>Uᵢ’s payment under proportional cost-sharing scheme</td>
</tr>
<tr>
<td>Pᵉₐᵢₙ</td>
<td>Uᵢ’s payment under egalitarian cost-sharing scheme</td>
</tr>
<tr>
<td>Δᵢ</td>
<td>(= Costᵢ − Pᵉₙₛ) Uᵢ’s saving from energy storage service</td>
</tr>
</tbody>
</table>

(1) Time-Varying Energy Pricing Plan: We consider discrete timeslots, indexed by t ∈ {1, ..., T}, where T is the number of timeslots in a day. The energy price of a time-varying time-of-use (ToU) pricing plan at timeslot t is denoted by pₜ(t). We suppose that the next-day ToU prices (pₜ(t))ₜ=₁ are announced before the end of today to all users and energy
storage operator, such that they can plan their consumption in a day-ahead manner.

(2) **Energy Users**: There are $N ( \geq 3)^2$ users, each denoted by $U_i$ where $i \in \{1, \ldots, N\}$. $U_i$ has certain energy demand over time, represented by a non-negative demand function $a_i(t) > 0$ for all $t$. The users aim to reduce their energy costs by utilizing a third-party energy storage service that stores energy at lower energy prices beforehand. We consider day-ahead energy scheduling, whereby $U_i$ forecasts her planned energy demand $a_i(t)$ in advance, and requests energy storage service in a day-ahead manner. If the energy from energy storage service is insufficient, $U_i$ will need to acquire additional energy from the grid for the residual consumption rate denoted by $y_i(t)$ at the respective price $p(t)$.

(3) **Energy Storage Service**: The energy storage service is provided by an energy storage operator, who has energy storage characterized by capacity $B(t)$, which is time-varying for modeling dynamic energy storage capacity. The energy storage is constrained by charging efficiency ratio $e_c \leq 1$ and discharging efficiency ratio $e_d \geq 1$, charge rate (i.e., ramp-up) constraint $r_c$ and discharge rate (i.e., ramp-down) constraint $r_d$. Let $b(t)$ be the current state-of-charge in the energy storage at time $t$, and $x^+(t)$ be the charging rate from the grid to the energy storage, whereas $x^-(t)$ be the discharging rate from the energy storage to $U_i$. When the energy storage is utilized, there is a per-unit service fee at each timeslot, $p_s$, which allows the energy storage operator to cover the wear-and-tear and maintenance cost.

Next, we will describe energy storage service scheduling in Section 3.2 and fair energy storage service cost-sharing in Section 3.3. We will present the blockchain model in Section 3.4, and incorporate privacy protection in the security and threat models in Section 3.5.

### 3.2 Energy Storage Service Scheduling

The energy storage service requires reservations from the users. The energy storage service operations will then be scheduled accordingly to minimize the overall energy cost. We formulate the optimization problem of energy storage service scheduling in (P1).

$$(P1) \quad \min \sum_{t=1}^{T} \left( p(t) \cdot (x^+(t) + \sum_{i=1}^{N} y_i(t)) + p_s \cdot x^+(t) \right) \quad (1)$$

s.t. $b(t+1) - b(t) = e_c x^+(t) - e_d x^-(t)$,

$$0 \leq b(t) \leq B(t), b(0) = 0, b(T+1) = 0, \quad (2)$$

$$x^+(t) \leq r_c, \quad (3)$$

$$\sum_{i=1}^{N} x^+_i(t) \leq r_d, \quad (5)$$

$$x^+_i(t) + y_i(t) = a_i(t), \quad (6)$$

$$\forall t \in \{1, \ldots, T\}, \forall i \in \{1, \ldots, N\} \quad (7)$$

The objective of (P1) is the total cost, including energy storage charging $x^+(t)$ and residual consumption $y_i(t)$ at the respective energy price $p(t)$ of timeslot $t$, as well as the energy storage service fee $p_s \cdot x^+(t)$. Constraint (2) updates the state-of-charge considering charging and discharging efficiency ratios. Constraint (3) ensures feasible state-of-charge. We assume that the initial and final state-of-charge are 0. Constraints (4)-(5) ensure the charging and discharging rates within the respective rate constraints. Constraint (6) ensures the balance of demands, such that each user’s demands are satisfied completely. Note that we do not consider the cost of energy distribution in power distribution network. This is sufficient to certain scenarios, for example, when the users are close to the energy storage operator.

We note that (P1) however relies on the knowledge of individual user’s demand $a_i(t)$. Hence, we present an alternate problem (P2).

$$(P2) \quad \min \sum_{t=1}^{T} \left( p(t) \cdot (x^+(t) + y(t)) + p_s \cdot x^+(t) \right) \quad (8)$$

s.t. $b(t+1) - b(t) = e_c x^+(t) - e_d x^-(t)$,

$$0 \leq b(t) \leq B(t), b(0) = 0, b(T+1) = 0, \quad (9)$$

$$x^+(t) \leq r_c, \quad (10)$$

$$x^-(t) \leq r_d, \quad (11)$$

$$x^+(t) + y(t) = a(t), \quad (12)$$

$$\forall b(t) \geq 0, x^-(t) \geq 0, y(t) \geq 0, x^+(t) \geq 0, \forall t \in \{1, \ldots, T\}.$$  

(P2) considers the total demand $a(t) = \sum_{i=1}^{N} a_i(t)$, total discharging rate $x^-(t) = \sum_{i=1}^{N} x^+_i(t)$ and total consumption rate $y(t) = \sum_{i=1}^{N} y_i(t)$, as well as the balance of the total demand in Constraint (13).

By Theorem 1, energy storage service scheduling can be solved by (P2), instead of (P1), involving no individual demand $a_i(t)$.

**THEOREM 1.** If $(x^-(t), y(t))^T_{t=1}$ is an optimal solution of (P2), then $(x^+(t), y(t))^T_{t=1}$, where $x^+_i(t) = \frac{a_i(t)}{a(t)} \cdot x^-(t)$ and $y_i(t) = \frac{a_i(t)}{a(t)} \cdot y(t)$, is an optimal solution of (P1).

See Appendix A for the proof.

**Remarks:** Note that when the energy storage discharges at rate $x^-(t)$, it can simultaneously compensate the users’ consumption at the same rate. This can be attained via VNM. We assume that the energy storage operator announces the parameters $p_s, e_c, e_d, r_c, r_d, (p(t), B(t))^T_{t=1}$ in advance. Everyone can compute the solution to (P2) with the knowledge of $(a(t))^T_{t=1}$.

### 3.3 Fair Cost-sharing of Energy Storage Service

After scheduling the energy storage service, the users are supposed to share and pay the associate cost to the energy storage operator. Next, we formulate how the cost of energy storage service should be shared among users in a fair manner. In (P2), in addition to the cost that is paid directly by the users to the grid (i.e., $\sum_{t=1}^{T} p(t) \cdot y(t)$), there is a cost incurred by the energy storage service as follows:

$$\text{Cost}_{\text{ess}} = \sum_{t=1}^{T} (p(t) + p_s) \cdot x^+(t) \quad (14)$$
Let $\text{Cost}_i = \sum_{t=1}^{T} p(t) \cdot x_i(t)$ be the partial original cost of $U_i$ in Eqn. (15) that would have been covered by energy storage service, which provides a basis on how to split $\text{Cost}_{\text{ess}}$. Note that the other part in Eqn. (15) (i.e., $\sum_{t=1}^{T} p(t) \cdot y_i(t)$) will be paid regardless of energy storage service. Noteworthily, if a user does not get any benefit from energy storage service (i.e., $a_i(t) > 0$ only when $p(t)$ is the lowest), then we have $x_i(t) = 0$ in (P1) and $\text{Cost}_i = 0$.

Suppose that each $U_i$ contributes payment $P_i$ to cover the energy storage service cost $\text{Cost}_{\text{ess}}$. A cost-sharing scheme denoted by $(P_1, \ldots, P_N)_{i=1}^{N}$ is called budget-balanced, if $\sum_{i=1}^{N} P_i = \text{Cost}_{\text{ess}}$, whereas it is called weakly budget-balanced, if $\sum_{i=1}^{N} P_i \geq \text{Cost}_{\text{ess}}$. A cost-sharing scheme $(P_1, \ldots, P_N)_{i=1}^{N}$ is called individually rational, if $\text{Cost}_i \geq P_i$ for all $i \in \{1, \ldots, N\}$. Evidently, each user would prefer an individually rational cost-sharing scheme. Otherwise, some users would rather not utilize energy storage service, as it will cost more.

We define two fair cost-sharing schemes, which are based on similar ideas in [8, 9], and show them to be individually rational by Theorem 2.

3.3.1 Proportional Cost-sharing Scheme.

One simple fair way is that each $U_i$ should pay proportionally to $\text{Cost}_i$. Namely,

$$p_i^{\text{pp}} = \frac{\text{Cost}_i}{\sum_{i=1}^{N} \text{Cost}_i}$$

Thus, each user has the same ratio of payment over individual cost (i.e., $\frac{p_i^{\text{pp}}}{\text{Cost}_i} = \frac{\text{Cost}_i}{\sum_{i=1}^{N} \text{Cost}_i}$). It is easy to check that proportional cost-sharing is budget-balanced (i.e., $\sum_{i=1}^{N} p_i^{\text{pp}} = \text{Cost}_{\text{ess}}$). Note that the payments are always non-negative (i.e., $\sum_{i=1}^{N} p_i^{\text{pp}} = 0$).

3.3.2 Egalitarian Cost-sharing Scheme.

Given a payment to energy storage service $P_i$, define the user’s saving of utilizing energy storage service by $\Delta_i = \text{Cost}_i - P_i$. Another fair cost-sharing scheme is that each user should split $\text{Cost}_{\text{ess}}$ in a way that attains the same saving for every user. Namely,

$$p_i^{\text{ega}} = \frac{\text{Cost}_i - \frac{\sum_{i=1}^{N} \text{Cost}_i - \text{Cost}_{\text{ess}}}{N}}{\sum_{i=1}^{N} x_i(t)} \cdot \sum_{t=1}^{T} p(t) = \sum_{t=1}^{T} x_i(t) \cdot \sum_{i=1}^{T} \frac{p(t) + p_a}{N} \cdot \sum_{t=1}^{T} x_i(t)$$

Thus, each $U_i$ attains the same saving as: $\Delta_i^{\text{ega}} = \frac{\sum_{i=1}^{N} \text{Cost}_i - \text{Cost}_{\text{ess}}}{N}$. It is easy to check that egalitarian cost-sharing is also budget-balanced (i.e., $\sum_{i=1}^{N} p_i^{\text{ega}} = \text{Cost}_{\text{ess}}$).

As a comparison, proportional cost-sharing guarantees the same percentage of savings (i.e., $\frac{\Delta_i^{\text{pp}}}{\text{Cost}_i}$) among users, whereas egalitarian cost-sharing guarantees the same savings (i.e., $\Delta_i^{\text{ega}}$) among users.

**Theorem 2.** If $(x^+(t), x^-(t))_{t=1}^{T}$ is an optimal solution of (P2) and let $x_i^+(t) = \frac{a_i(t)}{a(t)} \cdot x^+(t)$ and $y_i(t) = \frac{a_i(t)}{a(t)} \cdot y(t)$, then proportional and egalitarian cost-sharing schemes are individually rational.

Let $\hat{p}_i(t) = \frac{x_i^+(t) - p(t)}{a(t)}$ and $\text{Cost}_{\text{org}} = \sum_{i=1}^{T} x_i^+(t) \cdot p(t)$.

The proportional and egalitarian cost-sharing payments are given as follows:

$$\begin{cases}
p_i^{\text{pp}} = \frac{\text{Cost}_{\text{org}}}{N} \cdot \sum_{i=1}^{T} a_i(t) \cdot \hat{p}(t), \\
p_i^{\text{ega}} = \frac{\text{Cost}_{\text{org}}}{N} \cdot \sum_{i=1}^{T} a_i(t) \cdot \hat{p}(t) - \frac{\text{Cost}_{\text{org}} - \text{Cost}_{\text{ess}}}{N}
\end{cases}$$

See Appendix A for the proof.

**Remarks:** Egalitarian cost-sharing may have negative payments (i.e., $p_i^{\text{ega}} < 0$), when $\text{Cost}_i < \frac{\text{Cost}_{\text{org}}}{N}$. Namely, a user may be paid by other users who have larger original costs, in order to maintain equal savings among all users. In this case, such a user is not benefited sufficiently from energy storage service because of the presence of other users and capacity constraint, and hence, will be compensated by other users in egalitarian cost-sharing.

One may argue whether proportional cost-sharing is better than egalitarian cost-sharing, because it rules out negative payments. Here, we provide a solution to support both cost-sharing schemes. We will leave the decision of adopting which scheme to the users.

3.4 Blockchain Model

In this section, we describe a blockchain model for payments of energy storage service. We consider an account-based blockchain model like Ethereum (which is a general-purpose blockchain platform [35]), whereas Bitcoin operates with a different transaction-output-based model for cryptocurrency transactions only. Smart contracts are programming code on a blockchain that can provide customized computation tasks to each transaction (e.g., verification, data processing). Our payment system can be implemented as a smart contract.

The payment and auditing of energy storage service are carried out on a blockchain. Each user has an account on the blockchain. Users can top-up their accounts in advance. For cost-sharing, the users can initiate a joint payment transaction to the energy storage operator. The transaction records on the blockchain will also be used to verify VNM settlement by the grid operator.

Our blockchain model is based on a common model in the cryptography literature (e.g., Zether [6] that was built on Ethereum), which can be incorporated with privacy protection to conceal the transaction records. The blockchain consists of several components:

1. **Ledger:** An append-only ledger on a blockchain holds the records of all accounts and transactions. Note that by default, there is no privacy protection to the ledger, such that the account details and transaction histories are visible to the public. On Ethereum, one can create tokens on the ledger to represent certain digital assets. Our payment system is implemented by tokens, which allows us to incorporate privacy protection. To pay for energy storage service, users are required to purchase tokens that will be subsequently transferred to the energy storage operator and redeemed.
(2) **Accounts:** An account is identified by a public key $K^P$ and an address $ad$, which is the hash of the public key: $ad = H(K^P)$, where $H(\cdot)$ is a cryptographic hash function. The user manages the account by a private key $K^S$. Each account holds a balance of tokens, denoted by $\text{Bal}(ad)$, which by default is a publicly visible plaintext. Each $U_i$ has an account associated with a tuple $(ad_i, K^P_i, K^S_i, \text{Bal}(ad_i))$. We denote the energy storage operator’s account address by $ad_{ess}$. 

(3) **Transactions:** To initiate a transaction of tokens from $ad_i$ to $ad_j$ with transaction value $val_i$, the user submits a transaction request to the blockchain: $tx = (ad_i, ad_j, val_i)$, along with a signature $\text{sign}_{K^S_i}(tx)$ using the private key $K^S_i$ associated with $ad_i$. The transaction request will be executed only if $\text{Bal}(ad_i) \geq val_i$. A multi-transaction can also be requested. Let $\text{mtx} = (ad_i, ad_i, val_i)_{i=1}^N$, $\text{mtx}$ will be executed, only if $\text{Bal}(ad_i) \geq val_i$ for all $i$ and multi-signature $\text{sign}_{K^S_i}^N(\text{mtx})$ is present. Depending on the cost-sharing scheme, a user will pay either $P^\text{Ess}$ or $P^\text{Ess}^\text{Neg}$ to the energy storage operator. Each transaction request by default is a plaintext visible to the public. We will subsequently conceal the transaction records.

(4) **Receipts:** The recipient of a transaction can attach a receipt on the ledger, which may include additional information for further verification and auditing by a third-party. In VNM settlement, the grid operator will need to audit the amount of energy that a user can be offset from energy storage service, which can be verified from the receipts associated with transaction records.

Note that there may be a negative flow of payment in egalitarian cost-sharing, such that $val_i < 0$. Hence, we need to ensure the corresponding transaction on a blockchain still functions correctly.

**Theorem 3.** Consider a multi-transaction $\text{mtx} = (ad_i, ad_{ess}, val_i)_{i=1}^N$, where $val_i$ may be negative. Namely, every $ad_i$ pays to the energy storage operator $ad_{ess}$, if $\sum_{i=1}^N val_i > 0$, then $\text{mtx}$ can be handled on a blockchain by the following transaction operations:

$$\text{Bal}(ad_i) \leftarrow \text{Bal}(ad_i) - val_i, \text{ for all } i$$  \hspace{1cm} (18) \\
$$\text{Bal}(ad_{ess}) \leftarrow \text{Bal}(ad_{ess}) + \sum_{i=1}^N val_i$$  \hspace{1cm} (19)

See Appendix A for the proof.

3.5 Security & Threat Models

In the previous sections, we have not considered privacy protection. We define privacy protection in our problem. We assume synchronously authenticated communications among the parties, including users, blockchain, energy storage operator and grid operator, where the protocols proceed in several rounds and the parties can authenticate each other properly so that there is no man-in-the-middle attack.

3.5.1 Security Requirements.

Our system aims to satisfy the following security requirements:

3. (We skip some practical issues of a blockchain transaction, like nonce to prevent replay attack, account-locking against front-running attack, etc. But our model can easily incorporate the solutions from the security literature (e.g., [6]) to address these issues.

5.1 **Demand Concealment:** The user’s demand $(a_i(t))_{t=1}^T$ is private information, which should not be revealed to other users or energy storage operator in energy storage service scheduling, cost-sharing and payment. But the parameters, such as $p_s, e_c, e_d, r_c, r_d, (p(t), B(t))_{t=1}^T$, are publicly known to all users. We need to ensure the operations of scheduling, cost-sharing and payment can be achieved correctly without leaking any information about $(a_i(t))_{t=1}^T$ to others. Specifically, given $\left((a_i(t))_{t=1}^T\right)_{i=1}^N$, we need a privacy-preserving summation function for the aggregate demand:

$$\text{Sum}_{\text{prv}}\left[\left((a_i(t))_{t=1}^T\right)_{i=1}^N\right] = (a(t))_{t=1}^T.$$  

No user should learn any information from $\text{Sum}_{\text{prv}}\left[\cdot\right]$ other than her own inputs and the final outputs.

5.2 **Zero-knowledge Cost-Sharing & Payment:** With $(a(t))_{t=1}^T$, one can compute the energy storage service schedule $(x^+(t), x^-(t), y(t))_{t=1}^T$ by Theorem 1. Then, each $U_i$ can compute and make her payment $P_i = P^\text{Ess}_i$ by Theorem 2. Since $a_i(t)$ is only known to $U_i$, we need verifiable “zero-knowledge” proofs in the payment transactions to show the following properties without revealing $a_i(t)$ or $P_i$:

5.2.1 **Non-negativity of user demands:** $a_i(t) \geq 0$ for all $t, s$

5.2.2 **Correctness of payment:** $P_i$ is computed correctly according to Theorem 2 for each $U_i$.

5.2.3 **Sufficient balance of payment:** $\text{Bal}(ad_i) \geq P_i$, where $ad_i$ is the account address of $U_i$.

5.2.4 **Budget balance of energy storage service:** $\sum_{i=1}^N P_i = \text{Cost}_{\text{ess}}$. These zero-knowledge proofs will be crucial to safeguard against dishonest users in cost-sharing payments.

5.3 **Auditing for Virtual Net Metering:** The grid operator needs to verify the agreed energy flows from the energy storage operator to users, namely, $\left(\left(x^+(t)_{t=1}^T\right)_{i=1}^N, \left(x^-(t)_{t=1}^T\right)_{i=1}^N\right)_{t=1}^T$. To enable auditing, the energy storage operator needs to provide a receipt for each $U_i$ to certify her corresponding schedule $\left(x^+(t)_{t=1}^T\right)_{i=1}^N$, but without the knowledge of $\left(x^+(t)_{t=1}^T\right)_{i=1}^N$.

We emphasize that privacy protection is considered throughout the integrated process of scheduling, cost-sharing, payment and VNM auditing, without requiring a trusted third-party.

3.5.2 Threat Model.

Any users may be dishonest, who may try to cheat by paying less to energy storage service or claim more in VNM than what they ought to. These dishonest users may collude to coordinate their actions. We aim to ensure the privacy of honest users and the correctness of scheduling, cost-sharing and payment in the presence of an adaptive adversary who may corrupt a majority of up to $N - 2$ dishonest users. The adaptive adversary model provides a stronger security guarantee than the static one, where the adversary may corrupt users at any time during the protocols rather than before the protocols. A malicious adversary is more challenging than a classical semi-honest user due to her ability of deliberately deviating from the protocols for prying into others’ privacy or sabotaging the protocols.
In case of any dishonest actions being detected, our system will abort and notify all the users.

Note that our system is not required to identify individual dishonest user and it is fundamentally impossible [3] to identify a dishonest user in multi-party computation with a majority of dishonest users. There are secure multi-party computation protocols [12] that can identify a dishonest user, but requiring a majority of honest users and considerable computational overhead. On the other hand, we can impose further measures to mitigate dishonesty. For example, requiring proper user authentication to prevent shilling. Or, we can require each user to pay a deposit in advance, which will be forfeited if any dishonesty is detected.

4 CRYPTOGRAPHIC COMPONENTS

Our privacy-preserving solution relies on several basic components from cryptography. We briefly explain them in this section. More details can be found in a standard cryptography textbook [e.g., (5)].

Denote by $\mathbb{Z}_p = \{0, \ldots, p - 1\}$ the set of integers modulo $p$, for encrypting private data. For brevity, we simply write “$x + y$” and “$x \cdot y$” for modular arithmetic without explicitly mentioning “$\mod p$.”

We consider a usual finite group $G$ of order $p$. We pick $g, h$ as two generators of $G$, such that they can generate every element in $\mathbb{G}$ by taking proper powers, namely, for each $e \in \mathbb{G}$, there exist $x, y \in \mathbb{Z}_p$, such that $e = g^x \cdot h^y$. The classical discrete logarithmic assumption states that given $g^x$, it is computationally hard to obtain $x$, which underlies the security of many cryptosystems.

4.1 Cryptographic Commitments

A cryptographic commitment allows a user to hide a secret (e.g., to hide the balances and transactions on a blockchain). We use Pedersen commitment, which is perfectly hiding (i.e., a computationally unbounded adversary cannot unlock the secret) and computationally binding (i.e., an adversary cannot associate with another secret in polynomial time). To commit secret value $x \in \mathbb{Z}_p$, a user first picks a random number $r \in \mathbb{Z}_p$ to mask the commitment. Then, the user computes the commitment by:

$$Cm(x, r) = g^x \cdot h^r \mod p,$$

(20)

where $g$ is a generator of a multiplicative group $\mathbb{Z}_p^*$, $h = g^k \mod p$, $k$ is a secret value and $p$ is a large prime number.

Note that Pedersen commitment satisfies homomorphic property: $Cm(x_1 + x_2, r_1 + r_2) = Cm(x_1, r_1) \cdot Cm(x_2, r_2)$.

In a zero-knowledge proof of knowledge, the prover always can convince the verifier if knowing the secret), soundness (i.e., the prover cannot convince a verifier if not knowing the secret) and zero-knowledge (i.e., the verifier cannot learn the secret).

4.2 Zero-knowledge Proofs (ZKP)

In a zero-knowledge proof (of knowledge), a prover convinces a verifier of the knowledge of a secret without revealing the secret. For example, to show the knowledge of $(x, r)$ for $Cm(x, r)$ without revealing $(x, r)$. A zero-knowledge proof of knowledge should satisfy completeness (i.e., the prover always can convince the verifier if knowing the secret), soundness (i.e., the prover cannot convince a verifier if not knowing the secret) and zero-knowledge (i.e., the verifier cannot learn the secret).

4.2.1 $\Sigma$-Protocol

$\Sigma$-Protocol is a general approach to construct zero-knowledge proofs. Given a computationally non-invertible function $f(\cdot)$ that satisfies homomorphic property $f(a + b) = f(a) + f(b)$ and $f(x) = y$, one can prove the knowledge of the concealed $x$:

1. First, the prover sends a commitment $y' = f(x')$, for a random $x'$, to the verifier.
2. Next, the verifier replies with a random challenge $\beta$.
3. The prover replies with $z = x' + \beta \cdot x$ (which does not reveal $x$).
4. Finally, the verifier checks whether $f(z) = y' + \beta \cdot y$.

4.2.2 $\Sigma$-Protocol Based Zero-knowledge Proofs

Next, we present four crucial instances of zero-knowledge proofs based on $\Sigma$-protocol:

- **ZKP of Commitment** ($zkpCm$): Given $Cm(x, r)$, a prover can convince a verifier of the knowledge of $x$ without revealing $(x, r)$. Denote the corresponding zero-knowledge proof by $zkpCm[x]$.
- **ZKP of Summation** ($zkpSum$): Given a set of commitments $(Cm(x_i, r_i))_{i=1}^n$ and $y$, a prover can convince a verifier of the knowledge of $y = \sum_{i=1}^n x_i$ without revealing $(x_i)_{i=1}^n$. Denote the corresponding zero-knowledge proof by $zkpSum[y, (x_i)_{i=1}^n]$.
- **ZKP of Membership** ($zkpMbs$): Given a set $X = \{x_1, \ldots, x_n\}$ and $Cm(x, r)$, a prover can convince a verifier of the knowledge of $x \in X$ without revealing $x$. Denote the corresponding zero-knowledge proof by $zkpMbs[x, X]$.
- **ZKP of Non-Negativity** ($zkpNN$): Given $Cm(x, r)$, a prover can convince a verifier of the knowledge of $x \geq 0$ without revealing $x$. Denote the corresponding zero-knowledge proof by $zkpNN[x]$.

The detailed constructions of these zero-knowledge proofs can be found in Appendix B.

4.3 Non-interactive Zero-knowledge Proofs

An interactive zero-knowledge proof that requires a verifier-provided challenge can be converted to a non-interactive one by Fiat-Shamir heuristic to remove the verifier-provided challenge.

Let $H(\cdot) : \mathbb{Z}_p \rightarrow \mathbb{G}$ be a cryptographic hash function. Given a list of commitments $(Cm_1, \ldots, Cm_r)$, one can map to a single hash value $H(Cm_1[\ldots][Cm_r])$, where the input is the concatenated string of $(Cm_1, \ldots, Cm_r)$. In a $\Sigma$-protocol, one can set the challenge by $\beta = H(Cm_1[\ldots][Cm_r])$, where $(Cm_1, \ldots, Cm_r)$ are all the commitments generated by the prover prior to the step of verifier-provided challenge (Step 2 of $\Sigma$-protocol). Hence, the prover does not wait for the verifier-provided random challenge, and instead generates the random challenge himself. The verifier will generate the same challenge following the same procedure for verification. We denote the non-interactive versions of the previous zero-knowledge proofs by $nzkpCm$, $nzkpSum$, $nzkpMbs$, $nzkpNN$, respectively.

4.4 Public-Private Key Signatures

Cryptographic signatures are a standard tool to verify the authenticity of some given data. Suppose that a signer has a pair of public and private keys $(K_p, K_s)$ for an asymmetric key cryptosystem (e.g.,
To reveal $x$, each party $i$ broadcasts $x_i$ to other parties. Then each party can reconstruct $x = \sum_{i=1}^n x_i$. See an illustration in Figure 2.

Multiplications can also be computed by SPDZ, and the detailed description can be found in Appendix D. With additions and multiplications, one can construct a large class of computation functions (including comparison and branching conditions).

However, some parties may be dishonest, who may not perform the correct local computation. To safeguard against dishonest parties, an information-theoretical message authentication code (MAC) can be used for verification. Every secretly shared number is encoded by a MAC as $\gamma(x)$, which is also secretly shared as $(\gamma(x))$. The basic idea is that if a dishonest party wants to modify her share $x_i$, then he also needs to modify $\gamma(x_i)$ consistently. This allows dishonesty to be detectable by checking the corresponding MAC in the final output. The detailed description of MAC can be found in Appendix D. In the following, we write $\langle x \rangle$ meaning that both $(x)$ and the respective MAC $(\gamma(x))$ are secretly shared among users.

5.2 Overview of SPDZ Protocol

The SPDZ consists of three phases, as outlined as follows:

1. **Pre-processing Phase**: In this phase, a collection of shared random numbers will be constructed to mask the private input numbers. For each private input number of party $i$, there needs a shared random number $\langle r^i \rangle$, where $r^i$ is revealed to party $i$ only, but not to other parties.

2. **Online Phase**: To secretly shares a private input number $x^i$ using $\langle r^i \rangle$, without revealing $x^i$, it proceeds as follows:
   1. Party $i$ computes and reveals $z^i = x^i - r^i$ to all parties.
   2. Every party sets $\langle z^i \rangle \leftarrow z^i + \langle r^i \rangle$ (see A3).

Any computation circuit with additions or multiplications can be computed by local computations (e.g., A1-A3). The MACs are updated accordingly to preserve the consistency.

3. **Output and Validation Phase**: All MACs will be revealed for validation. If there is any inconsistency in MACs, then abort.

The details of SPDZ protocol can be found in Appendix C.

6 PRIVACY-PRESERVING SOLUTION FOR ENERGY STORAGE SHARING

This section presents an integrated solution for privacy-preserving energy sharing, based on blockchain, zero-knowledge proofs and multi-party computation protocol SPDZ.

6.1 Privacy-Preserving Ledger

First, we incorporate privacy protection to hide the transaction records on the ledger, while still allowing proper verifications for cost-sharing and VNM. As in other privacy-preserving blockchain platforms (e.g., Zether [6]), we conceal the balances and transaction values in the ledger by the respective cryptographic commitments instead of plaintext values. The accounts in the ledger will become as follows:

<table>
<thead>
<tr>
<th>ad</th>
<th>K^2</th>
<th>\text{Cm(Bal(ad_j))}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad</td>
<td>K^2</td>
<td>\text{Cm(Bal(ad_j))}</td>
</tr>
</tbody>
</table>

... ... ...

Table 2. Accounts in the privacy-preserving ledger

ACM SIGENERGY Energy Informatics Review

Volume 1 Issue 1, November 2021
A multi-transaction will be concealed as $\text{mtx} = (ad_i, ad_j, \text{Cm}(\text{val}_i))_{i=1}^N$. Because of concealed balances and transaction values, each user must provide $\text{nzkpNN}[\text{Bal}(ad_i) - \text{val}_i]$ along with each transaction request to prove the non-negativity of the resultant balance. Otherwise, the transaction request will be denied by the ledger without the correct $\text{nzkpNN}$.

6.2 Privacy-Preserving Protocol $\Pi_{pess}$

We design a protocol, denoted by $\Pi_{pess}$, to coordinate the users for privacy-preserving energy storage service scheduling, cost-sharing, payment and VNM settlement. We denote the energy storage operator and grid operator by $\text{Opres}$ and $\text{Opgr}$ respectively.

Before presenting the details of the protocol, we first outline some high-level ideas:

1. First, the users need to secretly share private individual demands $\langle a_i(t) \rangle$. Then, they can compute aggregate demand $\langle a(t) \rangle$ via SPDZ in a privacy-preserving manner.

2. To enable subsequent verification of the payment transactions and VNM, $U_i$ also needs to announce commitment $\text{Cm}(a_i(t))$ to each other. However, a dishonest user may use inconsistent commitment $\text{Cm}(a_i(t))$ with respect to the secretly shared $\langle a_i(t) \rangle$. To show the consistency between $\text{Cm}(a_i(t))$ and $\langle a_i(t) \rangle$, all users need to create zero-knowledge proof of commitment $\text{zkpCm}(a_i(t))$ via SPDZ using secretly shared $\langle a_i(t) \rangle$. If $\text{zkpCm}(a_i(t))$ is verified to be correct, then $\text{Cm}(a_i(t))$ and $\langle a_i(t) \rangle$ are consistent. Each user also creates a zero-knowledge proof of non-negativity $\text{nzkpNN}[a_i(t)]$.

3. After verifying $\text{zkpCm}[a_i(t)]$ and $\text{nzkpNN}[a_i(t)]$, the users reveal $\langle a(t) \rangle$ and verify the corresponding MAC to ensure the integrity of $a(t)$. Then, the users compute the energy storage service schedule with the knowledge of $\langle a(t) \rangle_{t=1}^T$.

4. Next, $U_i$ can make her payment $P_i = P_{i}^{\text{pp}}$ or $P_{i}^{\text{val}}$ by Theorem 2. The users jointly compute the total payments $\sum_{i=1}^{N} P_i$ via SPDZ in a privacy-preserving manner to ensure that the difference between Cost$_{pess}$ and $\sum_{i=1}^{N} P_i$ is within a negligible rounding error $\epsilon$, such that $|\text{Cost}_{pess} - \sum_{i=1}^{N} P_i| < \epsilon$. The users agree and set Cost$_{pess} = \sum_{i=1}^{N} P_i$.

5. To make cost-sharing payments for energy storage service on the ledger, the users need to create a zero-knowledge proof that $\sum_{i=1}^{N} P_i = \text{Cost}_{pess}$ via SPDZ. $U_i$ also creates $\text{nzkpNN}[\text{Bal}(ad_i) - P_i]$ locally. Then, the users submit a multi-transaction request with relevant zero-knowledge proofs to the ledger.

6. After the completion of multi-transaction of payments, the energy storage service schedule is executed. Afterwards, $\text{Opres}$ signs $\text{Cm}\langle x_i(t) \rangle$ as a receipt on the ledger for each user. Note that $\text{Cm}\langle x_i(t) \rangle$ can be generated based on $\text{Cm}(a_i(t))$ and the energy storage service schedule.

7. The users request VNM settlement with $\text{Opgr}$, who will verify $x_i(t)$ from the signed $\text{Cm}\langle x_i(t) \rangle$ on the ledger.

Next, we present the details of the privacy-preserving protocol $\Pi_{pess}$, consisting of four stages (Initialization, Pre-operation Scheduling, Cost-sharing Payment & Operation and Post-operation VNM Settlement), as follows:

Stage 0: Initialization.
In this stage, the system parameters are chosen and the pre-processing phase of SPDZ is executed among the users. See Appendix C for detailed SPDZ pre-processing phase.

Initialization:

1. Choose and announce a multiplicative group $\mathbb{Z}_p^*$, two generators $g, h \in \mathbb{Z}_p^*$ and hash function $H(\cdot) \rightarrow \mathbb{Z}_p$ as public information to all users. Note that $g$ and $h$ can be obtained via a coin-tossing protocol [36] among the users such that $\log_g h$ is unknown due to the hardness of discrete logarithm.

2. The energy storage operator announces $p_{\text{es}}, e_{\text{es}}, e_{\text{d}}, r_{\text{c}}, r_{\text{a}}, p(t), B(t)$ as public information to all users.

3. Initialize SPDZ pre-processing phase among all users.

Stage 1: Pre-operation Scheduling.
In this stage, the users will compute their aggregate day-ahead demands via SPDZ. The users also need to make commitments of their individual demands $\langle a_i(t) \rangle_{t=1}^T$, which will be used for auditing in VNM. We ensure that the individual demands shared via SPDZ match the ones being committed. This can be accomplished by computing zero-knowledge proof of commitment $\text{zkpCm}[a_i(t)]$ using the secretly shared $\langle a_i(t) \rangle$. Next, the users will compute the optimal energy storage service schedule in (P2) based on aggregate demands $\langle a(t) \rangle_{t=1}^T$.

Protocol $\Pi_{pess}^{(1)}$:

1. $U_i$ commits $C_i(t) = \text{Cm}(a_i(t), r_i(t))$ for all $t$ and announces $(C_i(t))_{t=1}^T$ to all users with $(\text{nzkpNN}[a_i(t)])_{t=1}^T$, where $r_i(t)$ is a random masking number. All users verify $\text{nzkpNN}[a_i(t)]$. If verification of $\text{nzkpNN}[a_i(t)]$ fails, announce Abort.$t$.

2. $U_i$ secretly shares $\langle a_i(t) \rangle$ and $\langle r_i(t) \rangle$ via SPDZ for all $t$.

3. To show the equality of $a_i(t)$ in $\langle a_i(t) \rangle$ and $C_i(t)$, $U_i$ constructs an $\text{zkpCm}[a_i(t)]$ distributedly via SPDZ:

   a. $U_i$ randomly generates $\langle a_i(t) \rangle = \langle a_i(t) \rangle$ and secretly shares as $\langle a_i(t) \rangle$ and $\langle r_i(t) \rangle$. Then $U_i$ announces $C_i(t) = \text{Cm}(a_i(t), r_i(t))$ to all users. Note that all the users must complete this step before proceeding to the next step to produce a common random challenge $\beta(t)$.

   b. All the users conduct a coin-tossing protocol to obtain a random challenge $\beta(t)$. Firstly, each user announces a commitment $C_i^{\beta}(t)$ of a randomly generated number $r_i^{\beta}(t) \in \mathbb{Z}_p$. Then all the users reveal $r_i^{\beta}(t)$ and compute a random challenge $\beta(t) = \sum_{i=1}^{N} r_i^{\beta}(t)$.

   c. All users compute and reveal $\langle z_{a_i(t)} \rangle = \langle a_i(t) \rangle + \beta(t) \cdot \langle a_i(t) \rangle$ and $\langle z_{r_i(t)} \rangle = \langle r_i(t) \rangle + \beta(t) \cdot \langle r_i(t) \rangle$ for $U_i$.

   d. This creates $\text{zkpCm}[a_i(t)] = (C_i^{\beta}(t), z_{a_i(t)}, z_{r_i(t)})$. All users verify $\text{zkpCm}$ by checking

   \[ g^{z_{a_i(t)}} \cdot h^{z_{r_i(t)}} \equiv C_i^{\beta}(t) \cdot \text{Cm}(a_i(t), r_i(t))^{\beta(t)} \]

   e. If the verification of $\text{zkpCm}$ fails, announce Abort.$t$.

4. The users compute $\langle a(t) \rangle \leftarrow \sum_{i=1}^{N} \langle a_i(t) \rangle$ for all $t$ via SPDZ. Reveal $\langle a(t) \rangle$ to all users. Check MAC of $\langle a(t) \rangle$. If the MAC check fails, announce Abort.$t$. 
Stage 2: Cost-sharing Payment & Operation.

In this stage, the users will split the cost of energy storage service based on proportional or egalitarian cost-sharing scheme via SPDZ. The users compute the payment commitments and verify the validity of Cost$_{ess}$ by comparing with $\sum_{i=1}^{N} P_i$. Before issuing the multi-transaction, the users compute nzkpSum[$Cost_{ess}, (P_i)_{i=1}^{N}$] to satisfy Theorem 3. The users then make energy storage service payments via privacy-preserving blockchain. After receiving the payments, the energy storage operator will issue verifiable receipts on the ledger.

Protocol $\Pi^{(2)}_{pess}$:

1. $U_i$ computes $P_i = p^{pp}_i$ (or $p^{pgr}_i$) by Theorem 2, and announces commitment $\mathcal{Cm}(P_i, r_i)$ to all users, and secretly shares $\langle r_i \rangle$ via SPDZ.

2. The users also compute the total payments via SPDZ by

$$\sum_{i=1}^{N} P_i = \sum_{i=1}^{N} \left( \text{Cost}_{ess} \cdot \tilde{p}(i) \cdot \frac{N}{\text{Cost}_{org} - \text{Cost}_{ess}} \right)$$

and check if satisfying $[\text{Cost}_{ess} - \sum_{i=1}^{N} P_i] < \varepsilon$, where $\varepsilon$ is a small fault-tolerant factor, which restricts the rounding error, arising from computing $\text{Cost}_{org}$ or $\tilde{p}(i)$. If satisfied, the users let $\text{Cost}_{ess} = \sum_{i=1}^{N} P_i$. Otherwise, announce Abort.

3. The users compute nzkpSum[$Cost_{ess}, (P_i)_{i=1}^{N}$] via SPDZ distributively:

   a. $U_i$ randomly generates and secretly shares $\langle r'_i \rangle$ in $\mathbb{Z}_p$ before announcing $\mathcal{Cm}(0, r'_i)$.

   b. All users compute $C' = \prod_{i=1}^{N} \mathcal{Cm}(0, r'_i)$ and obtain a random challenge $\beta = H(C')$. Then all users compute $\tilde{C}_i = \sum_{i=1}^{N} \langle r'_i \rangle + \beta \cdot \sum_{i=1}^{N} \langle r_i \rangle$. Then reveal $\langle r_i \rangle$.

   c. This creates nzkpSum[$Cost_{ess}, (P_i)_{i=1}^{N}$] = $\{C', \tilde{C}_i\}$. All users verify nzkpSum by checking:

$$g^{\beta \cdot \text{Cost}_{ess} \cdot \tilde{C}_i} \equiv C' \cdot \prod_{i=1}^{N} \mathcal{Cm}(P_i, r_i)^{\beta}$$

If the verification of nzkpSum fails, announce Abort.

4. $U_i$ computes nzkpNN[Ba1(ad$_i$) - $P_i$] based on $\mathcal{Cm}(P_i)$.

5. Let the account address of $Opr_{es}$ be ad$_{ess}$. The users submit a multi-transaction request

$$\text{mtx} = \{ad_i, ad_{ess}, \mathcal{Cm}(P_i)\}_{i=1}^{N}$$

to the ledger, along with

$$\text{nzkpSum}[Cost_{ess}, (P_i)_{i=1}^{N}]$$

and nzkpNN[Ba1(ad$_i$) - $P_i$]$_{i=1}^{N}$

6. The ledger verifies nzkpSum and nzkpNN before proceeding the transaction. If the verification fails, announce Abort.

Stage 3: Post-operation VNM Settlement.

In this stage, $Opr_{es}$ will sign the receipts of individual energy storage service $(x^{-}_i(t))_{t=1}^{T}$. The receipts will be stored on the ledger. When the users request VNM settlement with $Opr_{gr}$, $Opr_{gr}$ will verify their claims by the receipts on the ledger.

Protocol $\Pi^{(3)}_{pess}$:

1. The users upload $(\mathcal{Cm}(a_i(t))_{t=1}^{T})_{i=1}^{N}$ (from Stage 1) to the ledger.

2. $Opr_{es}$ computes commitment $\mathcal{Cm}(x^{-}_i(t)) = \mathcal{Cm}(a_i(t))^{\frac{\tilde{x}_i(t)}{t_{pp}}}$ (according to Theorem 2) for all $t$ and $i$.

3. Let the public-private keys of $Opr_{es}$ be $(K^p_{ess}, K^a_{ess})$. $Opr_{es}$ signs $\text{sign}_{K^a_{ess}}[\mathcal{Cm}(x^{-}_i(t))]$ along with $\mathcal{Cm}(x^{-}_i(t))$ to be stored on the ledger.

4. $Opr_{es}$ prepares VNM and provides energy export profile $(x^{-}_i(t))_{t=1}^{T}$ to $Opr_{gr}$.

5. $U_i$ submits a claim for reimbursement by referring to $\mathcal{Cm}(x^{-}_i(t))$ and $\text{sign}_{K^a_{ess}}[\mathcal{Cm}(x^{-}_i(t))]$ on the ledger. $U_i$ also reveals $x^{-}_i$ to $Opr_{gr}$ to prove the validity.

6. $Opr_{gr}$ verifies $\text{sign}_{K^a_{ess}}[\mathcal{Cm}(x^{-}_i(t))]$ by public key $K^p_{ess}$, and compares the energy demand profile $(a_i(t))_{t=1}^{T}$ with $(x^{-}_i(t))_{t=1}^{T}$. If the verification is consistent, $Opr_{gr}$ will deduct the amount $\sum_{t=1}^{T} x^{-}_i(t) \cdot p(t)$ from $U_i$’s total payment.

Remarks: In protocol $\Pi_{pess}$, each user is required to input her privacy demand $\langle a_i(t) \rangle_{t=1}^{T}$ in two privacy-preserving ways: (1) commitment $\mathcal{Cm}(a_i(t))$, and (2) secretly shared value in SPDZ $\langle a_i(t) \rangle$. While the commitment $\mathcal{Cm}(a_i(t))$ is used to generate other zero-knowledge proofs for payments and VNM, the secretly shared $\langle a_i(t) \rangle$ is used to compute service scheduling and cost-sharing. Both inputs should be consistent (i.e., checked by zkp$\mathcal{Cm}(a_i(t))$ that is constructed via SPDZ). Also, $\mathcal{Cm}(a_i(t))$ can be used to construct $\mathcal{Cm}(P_i)$ and $\mathcal{Cm}(x^{-}_i(t))$ without the knowledge of $a_i(t)$, because of its homomorphic property for these constructions.

See Appendix D for the security analysis of $\Pi_{pess}$ for satisfying security requirements S1-S3.

7 EVALUATION

In this section, we present an evaluation study of our solution, including the effectiveness of energy storage sharing, multi-party computation protocol performance and the incurred cost of smart contract implementation on a practical blockchain platform.

7.1 Energy Storage Service Scheduling

We first evaluate the effectiveness of energy storage sharing. We selected 120 users from the Smart* microgrid dataset [2]. We consider a single 24-hour period, from midnight to next midnight.

In particular, we present the temporal data trace of scheduled energy storage services for 4 users. We observe that each of the user can utilize energy storage discharging during peak-hour. Most users
acquire energy from the grid during off-peak-hour, and partially during shoulder-hour. Next, we study the saving ($A_i$) of each user. In Figure 3, we study different cost-sharing schemes of energy storage service for 4 users. We observe that proportional cost-sharing gives each user at the same percentage of saving of 28.78% when energy storage $B$ is 400 kWh, whereas different user has different percentage of saving with egalitarian cost-sharing, but with the same amount of saving of $37.90$.

7.2 SPDZ Performance

Next, we evaluate the performance of SPDZ in stages 1-2 of $\Pi_{pess}$. We skip stage 3 due to its negligible performance compared with stages 1-2. We consider 144 time slots in a single 24-hour period. All the results were averaged over 20 instances.

7.2.1 Computational Overhead. We scaled the number of users from 5 to 25. Figure 4a displays the average running time incurred at each user in the stage 1. The running time shows a linearly growing trend with the increased number of users. The running time starts from about 0.91 seconds with 5 users to around 4.32 seconds with 25 users. We skip the displaying of the stage 2 due to the negligible computational overhead of only several milliseconds.

7.2.2 Communication Overhead. Figure 4b shows the average total volume of the transmission data in stages 1 and 2. It is evident that the total transmission amount scale linearly with the growing number of users in both stages. The data volume increases from 3.62 MB with 5 users to about 26.89 MB with 25 users in stage 1. In contrast, the data volume in stage 2 starts from a merely 5 KB with 5 users to about 40 KB with 25 users. Thus, the total data volume in stage 1 dominates the entire protocol.

7.3 Ethereum Smart Contract Gas Costs

We implemented the payment systems as a smart contract on real-world Ethereum blockchain platform. The smart contract is specified by Solidity programming language [27]. We outline some implementation components as follows:

(1) Pedersen. This component aims to realize the underlying Pedersen commitment scheme.

(2) ESToken. We created a Ethereum-based cryptocurrency ESToken for our energy sharing scenario, whereby users are able to pay the energy cost without revealing the true payments.

(3) MultiSignature. This component allows the users to submit a multi-transaction request, where the transaction will proceed unless all the involved users validate the transaction. There are three main methods: submitTransaction(), confirmTransaction() and executeTransaction().

Distributed miners will execute the compiled bytecode of the smart contracts in Ethereum Virtual Machine. Miners will charge additional Ether/ETH (Ethereum native cryptocurrency) called gas costs, because the extra computational tasks incurred by smart contracts will be broadcast throughout the blockchain. Gas costs are used to measure the amount of computational resources to execute the operations required by a transaction. We measured the incurred gas costs by our smart contracts and used a 255-bit prime number $q$ for Pedersen commitment since Solidarity supports at most 256-bit numbers. We employed Truffle Suite [40] as the Ethereum development framework to test and measure the average gas costs of Multi-Signature methods, shown in Table 3, where $N$ is the number of users and $N_b$ indicates the number of bits to represent the plaintext payment in the $nzkpMbs$. A transaction initiator must pay sufficient amount of gas costs to the miner, who creates transaction blocks on the network. The gasPrice in a transaction allows the transaction initiator to set the gas price that she is willing to pay. The higher the gas prices, the higher probability the transaction will be chosen by the miner in a block. We use the standard gas price $54$.

*The actual gas costs may vary based on the random generated parameters of the zero-knowledge proofs. Here, we show the average gas costs.
Gwei and Ether price\(^5\) $4068 USD/ETH to estimate the equivalent transaction cost in Ether (see [39] and [11]).

Table 3. Table of gas costs for multi-signature methods

<table>
<thead>
<tr>
<th>Input (bytes)</th>
<th>Gas Cost</th>
<th>Ether</th>
<th>USD (as on 25 Oct 21)</th>
</tr>
</thead>
<tbody>
<tr>
<td>submit()</td>
<td>106k</td>
<td>0.0057</td>
<td>$ 23.2</td>
</tr>
<tr>
<td>confirm()</td>
<td>3600k</td>
<td>0.1944</td>
<td>$ 790.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>128 + 180 xN-p</td>
</tr>
</tbody>
</table>

Figure 4c presents the gas cost of executeTransaction() function. We observe that the gas cost is linearly proportional to the number of involved users, starting from 1437k (0.0776 ether, $ 315.7) with 5 users to 5986k (0.3232 ether, $ 1314.8) with 25 users, since the verification of nzkpSum depends on the number of users. Overall, we observe only moderate incurred costs by our smart contract, which are comparable to other privacy-preserving smart contract studies in the literature.

8 EXTENSIONS AND LIMITATIONS

In this section, we discuss some possible extensions to enhance our privacy-preserving energy storage sharing solution. We also discuss some limitations of our current solution.

8.1 Flexible Demands and Hour-ahead Scheduling

Our current energy storage service scheduling is designed to operate in a day-ahead manner. There is a limitation that users are not supposed to alter their requested demand (at least, for the part of demand that is allocated to be satisfied by scheduled energy storage discharging). A more flexible approach is desirable to incorporate both day-ahead and hour-ahead scheduling processes, which is called dual service scheduling. The hour-ahead scheduling process allows the users to request energy storage service in a much shorter time-window.

The dual service scheduling can be achieved by dividing the capacity of energy storage into two parts: (1) \(B_d(t)\) as the capacity for day-ahead service scheduling and (2) \(B_h(t)\) for hour-ahead service scheduling, such that \(B_d(t) + B_h(t) = B(t)\). Both service scheduling processes can run separately, with hour-ahead scheduling running before each hour. Our privacy-preserving service scheduling process can be extended straightforwardly to operate the two parts of service scheduling in tandem.

8.2 Virtual Net Metering Service Fees

In our current privacy-preserving service scheduling process, we assume that the energy storage operator can transfer the credits as a result of exporting energy through discharging energy storage via virtual net metering to the users, with no additional service fees from the grid operator. Namely, the energy storage operator can always transfer the full credits of \(x^{-1}(t) \cdot p(t)\) to compensate the users’ consumption at each timeslot \(t\).

However, the grid operator may charge additional fees to attain virtual net metering, in particular, when the energy storage operator is not located in the same local grid network as the users. For instance, the grid operator charges fees as \(s_{vnm}\) percent of the credits transferred in virtual net metering. Namely, when the energy storage operator transfers the credits of \(x^{-1}(t) \cdot p(t)\) to the users, the users actually receives \((1 - s_{vnm}) \cdot x^{-1}(t) \cdot p(t)\).

One possible approach to incorporate in our privacy-preserving service scheduling process is to modify Cons. (2) as

\[
b(t + 1) - b(t) = e_c x^t(t) - e_d \frac{1}{1-s_{vnm}} \left( \sum_{i=1}^{N} x_i^t(t) \right)
\]

That is, we reduce the actual amount of discharged energy from the energy storage as to incorporate the additional service fees for virtual net metering from the grid operator.

8.3 Reducing Gas Cost on Ethereum

Although our current solution incurs a moderate gas cost on Ethereum, this is still considerable when the Ether price has increased significantly in the recent years. High Ether price has deterred many real-world smart contract projects from operation. We seek to improve the gas cost of our smart contract solution. There are a number of possible improvements. First, rather than storing the receipts on the ledger, which can take up a considerable storage space and costs extra gas cost, we can store a small hash pointer instead on the ledger. The users can verify the contents of the receipts by matching with the hash pointer. Second, there should be more efficient zero-knowledge proofs that can be executed on smart contract. One possible option is Bulletproofs [7], which are more succinct zero-knowledge proofs than \(\Sigma\) protocol. Bulletproofs have been employed in certain privacy-preserving blockchain platforms (e.g. Monero, Zether [6, 34]). Third, even though we implemented our solution as a smart contract on permissionless Ethereum blockchain platform, our solution can also be implemented on a permissioned blockchain platform, on which the gas cost is not a major concern.

9 CONCLUSION

In this paper, we provide a novel approach to support third-party energy storage sharing without compromising the privacy of individual users. In our privacy-preserving solution, an energy storage operator is only revealed the minimal information to schedule energy storage operations, without knowing users’ private demands. At the same time, the users can divide the cost of energy storage service fairly among themselves without knowing each other’s demands. Our solution can effectively safeguard against a majority of dishonest users, without requiring trusted third-parties. We implemented our solution as a smart contract on Ethereum blockchain platform, which incurs moderate overhead and gas costs in practice.

In future work, we will support robustness against potentially dishonest energy storage operators. For instance, we can require an energy storage operator to prove that her energy export profile matches the service schedules in order to receive the payments from users. We will also explore the support for peer-to-peer energy storage sharing by distributing service scheduling and cost-sharing computation among the end users themselves.
REFERENCES


APPENDIX

A PROOFS

THEOREM 1. If \((x^0(t), y(t))_{t=1}^T \) is an optimal solution of (P2), then \((x^0(t), y(t))_{t=1}^T \) is an optimal solution of (P1).

PROOF. Since \((x^0(t), y(t))_{t=1}^T \) is an optimal solution of (P2), it satisfies the condition \(x^0(t) + y(t) = a(t)\).

We let \(x^0_i(t) = a(t) - x^0(t)\) and \(y_i(t) = a(t) - y(t)\), then \((x^0_i(t), y_i(t))_{t=1}^T \) satisfies \((x^0_i(t) + y_i(t)) = a(t)\) for all \(i \in \{1, \ldots, N\}\) and \(t \in \{1, \ldots, T\}\), and hence, is a feasible solution of (P1).

Next, we argue that \((x^0(t), y(t))_{t=1}^T \) is a feasible solution of (P1) by contradiction. Suppose that there exists a better solution \((x^1(t), y^1(t))_{t=1}^T \) with a lower total cost in (P2). Then \((x^1(t), y(t))_{t=1}^T \) is not an optimal solution, because we can find another better solution by considering \((x^0(t), y(t))_{t=1}^T + (x^1(t), y^1(t))_{t=1}^T\) instead, which is also a feasible solution of (P2). This will violate the optimality of \((x^0(t), y(t))_{t=1}^T\).

THEOREM 2. If \((x^0(t), x^0(t))_{t=1}^T \) is an optimal solution of (P2) and let \(x^0_i(t) = a(t) - x^0(t)\) and \(y_i(t) = a(t) - y(t)\), then proportional and egalitarian cost-sharing schemes are individually rational.

Let \(\rho(t) \triangleq x^0(t) \frac{p(t)}{a(t)}\) and \(\text{Cost}_{\text{org}}^0 \triangleq \sum_{t=1}^T x^0(t) \cdot \rho(t)\). The proportional and egalitarian cost-sharing payments are given as follows:

\[
\begin{aligned}
&\frac{p(t)}{a(t)} = \text{Cost}_{\text{org}}^0 + \sum_{t=1}^T a(t) \cdot \hat{\rho}(t) = \sum_{t=1}^T a(t) \cdot \hat{\rho}(t) - \text{Cost}_{\text{org}}^0
\end{aligned}
\]
PROOF. For proportional cost sharing, $U_i$’s saving will be

\[
\Delta^\text{pp}_i = \sum_{t=1}^{T} x^*_t(t) \cdot p(t) - \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t) - \sum_{t=1}^{T} x^*_t(t) \cdot p(t)
\]

\[
= \left( \sum_{t=1}^{T} x^*(t) \cdot p(t) - \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t) \right) \cdot \frac{\sum_{t=1}^{T} x^*_t(t) \cdot p(t)}{\sum_{t=1}^{T} x^*(t) \cdot p(t)}
\]

(24)

(25)

Next, we show $\sum_{t=1}^{T} x^*(t) \cdot p(t) \geq \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t)$ by contradiction. Suppose $\sum_{t=1}^{T} x^*(t) \cdot p(t) < \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t)$, then $(x^*(t), x^*(t))_{t=1}^T$ is not an optimal solution of (P2) because one can always find a better solution by not charging energy storage according to $x^*(t)$. Instead, drawing energy at the time it is needed will only cost $\sum_{t=1}^{T} x^*(t) \cdot p(t)$, which is cheaper than the cost of charging and subsequently discharging from energy storage $(\sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t))$. Hence, we conclude that $\sum_{t=1}^{T} x^*(t) \cdot p(t) \geq \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t)$ and $\Delta^\text{pp} \geq 0$.

For egalitarian cost sharing, $U_i$’s saving will be

\[
\Delta^\text{egs}_i = \sum_{t=1}^{T} x^*_t(t) \cdot p(t) - \sum_{t=1}^{T} (p(t) + p_x) \cdot x^*(t) + \frac{\sum_{t=1}^{T} x^*_t(t) \cdot p(t)}{N}
\]

(26)

Following a similar approach by contradiction, we can similarly show that $\Delta^\text{egs} \geq 0$.

\[
\text{Theorem 3. Consider a multi-transaction mtx = (ad}_i, ad_{egs}, val_i)_{i=1}^N, \text{ where val}_i \text{ may be negative. Namely, every ad}_i \text{ pays to the energy storage operator ad}_{egs}. \text{ If } \sum_{i=1}^{N} \text{ val}_i > 0, \text{ mtx can be handled on a blockchain by the following transaction operations:}
\]

Bal(ad$_i$) $\leftarrow$ Bal(ad$_i$) - val$_i$, for all $i$

Bal(ad$_{egs}$) $\leftarrow$ Bal(ad$_{egs}$) + $\sum_{i=1}^{N}$ val$_i$

(27)

(28)

PROOF. It is straightforward to see that Eqn. (27) applies to the case when val$_i$ is negative. As long as $\sum_{i=1}^{N} \text{ val}_i > 0$, there is no net out-going payment from ad$_{egs}$. Therefore, mtx can be handled properly.

B ZERO-KNOWLEDGE PROOFS OF KNOWLEDGE

B.1 Zero-knowledge Proof of Commitment (zkpCm)

Given Cm(x, r), a prover wants to convince a verifier of the knowledge of (x, r). We can apply $\Sigma$-protocol as follows:

1. The prover randomly generates $(x', r') \in \mathbb{Z}_p^2$ and sends the commitment Cm$(x', r')$ to the verifier.
2. The verifier sends a random challenge $\beta \in \mathbb{Z}_p$ to the prover.
3. The prover replies with $z_x = x' + \beta \cdot x$ and $z_r = r' + \beta \cdot r$.
4. The verifier checks whether $g^{z_x} \cdot h^{z_r} \overset{?}{=} \text{Cm}(x', r') \cdot \text{Cm}(x, r)^{\beta}$.

Denote a zero-knowledge proof of commitment for Cm(x, r) by zkpCm[x].

B.2 Zero-knowledge Proof of Summation

Given commitments $\{\text{Cm}(x_1, r_1), \ldots, \text{Cm}(x_n, r_n)\}$ and $y$, a prover wants to convince a verifier of the knowledge of $y = \sum_{i=1}^{n} x_i$ without revealing $(x_1, \ldots, x_n)$. We can apply $\Sigma$-protocol as follows:

1. The prover randomly generates $r' \in \mathbb{Z}_p$ and sends the commitment Cm$(0, r')$ to the verifier.
2. The verifier sends a random challenge $\beta \in \mathbb{Z}_p$ to the prover.
3. The prover replies with $z_r = r' + \beta \cdot \sum_{i=1}^{n} r_i$.
4. The verifier checks whether $g^{z_r} \cdot h^{z_r} \overset{?}{=} \text{Cm}(0, r') \cdot \prod_{i=1}^{n} \text{Cm}(x_i, r_i)^{\beta}$

Denote a zero-knowledge proof of summation for $\{\text{Cm}(x_i, r_i)\}_{i=1}^{n}$ by zkpSum[y, (xi, ri)$_{i=1}^{n}$].

B.3 Zero-knowledge Proof of Membership

Given a set $X = \{x_1, \ldots, x_n\}$ and Cm(x, r), a prover wants to convince a verifier of the knowledge of $x \in X$ without revealing x. We can apply $\Sigma$-protocol as follows:

1. Suppose $x = x_i \in X$. The prover first randomly generates $(x'_j, r'_j) \in \mathbb{Z}_p^2$ and computes the commitment Cm$(x'_j, r'_j)$ for all $j \in \{1, \ldots, n\}$. Then, the prover randomly generates $\beta_j \in \mathbb{Z}_p$ for each $j \in \{1, \ldots, n\} \setminus \{i\}$, and computes $z_{x_i} = \begin{cases} x'_i + (x_i - x_j) \beta_j, & \text{ if } j \in \{1, \ldots, n\} \setminus \{i\} \\ x'_i, & \text{ if } j = i \end{cases}$

Next, the prover sends $(\text{Cm}(x'_j, r'_j), z_{x_i})_{j=1}^{n}$ to the verifier.
2. The verifier sends a random challenge $\beta \in \mathbb{Z}_p$ to the prover.
3. The prover sets $\beta_i = \beta - \sum_{j \neq i} \beta_j$, then computes $z_{x_i} = r'_i + r \cdot \beta_j$ for all $j \in \{1, \ldots, n\}$, and sends $(\beta_j, z_{x_i})_{j=1}^{n}$ to the verifier.
4. The verifier checks whether $\beta \overset{?}{=} \sum_{i=1}^{n} \beta_j$ and $g^{z_{x_i}} \cdot h^{z_r} \overset{?}{=} \text{Cm}(x'_j, r'_j) \cdot \left( \frac{\text{Cm}(x, r)}{g^{r_j}} \right)^{\beta_j}$ for all $j \in \{1, \ldots, n\}$.

Denote a zero-knowledge proof of membership for $x \in X$ by zkpMbs[x, X].

B.4 Zero-knowledge Proof of Non-Negativity

Given Cm(x, r), a prover wants to convince a verifier of the knowledge of $x \geq 0$ without revealing x. Suppose $x < 2^m$. We aim to prove there exist $(b_1, \ldots, b_m)$ such that $b_i \in \{0, 1\}$ for $i \in \{0, \ldots, m\}$ and $\sum_{i=0}^{m} b_i \cdot 2^{i-1} - r \in \mathbb{Z}$. We can apply $\Sigma$-protocol as follows:

1. The prover sends $(\text{Cm}(b_j, r_j))_{j=1}^{m}$ to the verifier, and provides zkpMbs[b$_j$, [0, 1] for each b$_j$ to prove that b$_j \in \{0, 1\}$. Also, the prover randomly generates $r' \in \mathbb{Z}_p$ and sends the commitment Cm$(0, r')$ to the verifier.
2. The verifier sends a random challenge $\beta \in \mathbb{Z}_p$ to the prover.
3. The prover replies with $z_r = r' + \beta \cdot \sum_{j=1}^{m} r_j \cdot 2^j - r$.
4. The verifier checks whether $h^{z_r} \overset{?}{=} \text{Cm}(0, r') \cdot \text{Cm}(x, r)^{-\beta} \cdot \prod_{i=1}^{m} \text{Cm}(b_j, r_j)^{\beta} \cdot 2^{i-1} - r$.

Denote a zero-knowledge proof of $x \geq 0$ by zkpNN[x].

B.5 Security Proof

It is straightforward to prove the completeness of these protocols. We will provide detailed proofs on their soundness and honest-verifier zero-knowledge properties below.
B.5.1 Soundness Proof. Proving soundness is equivalent to showing that there exists a knowledge extractor who makes the prover successfully answer two random given challenges $\beta_1$ and $\beta_2$:

- (zkpCom) For $x$, we have $\left( z'_x = x' + \beta_1 \cdot x, z''_x = x' + \beta_2 \cdot x \right)$. For $r$, we have $\left( z'_r = r' + \beta_1 \cdot r, z''_r = r' + \beta_2 \cdot r \right)$. Then we obtain $x = z'_x - z''_x$ and $r = z'_r - z''_r$. Finally, we can check that $g^x h^r = \text{CM}(x, r)$. We write $\pi_f$ such that each party

- (zkpSum) Let $z_1 = r' + \beta_1 \cdot \sum_{i=1}^{N} r_i$ and $z_2 = r' + \beta_2 \cdot \sum_{i=1}^{N} r_i$. Then we have $\sum_{i=1}^{N} r_i = z_1 - z_2$. Finally, we can check that $g^{y z'_1 \sum_{i=1}^{N} r_i} = \prod_{i=1}^{n} \text{CM}(x_i, r_i)$ to prove that $y = \sum_{i=1}^{N} x_i$.

- (zkpMbs) Let $z'_r = r' + \beta_1 \cdot r$ and $z''_r = r' + \beta_2 \cdot r$. Then we have $r = z'_r - z''_r$. Finally, we can check for each $j$ that $g^{z'_r_j} h^r_j = \text{CM}(x, r)$ to prove that $x \in X$.

B.5.2 Honest-verifier Zero-knowledge Proof. It suffices to show that there exists a simulator who can produce another set of zero-knowledge proofs that are computationally indistinguishable from a given set of zero-knowledge proofs:

- (zkpCom) $\text{CM}(x', r') = g^{x'} h^{r'} \cdot \text{CM}(x, r)^{-\beta}$.
- (zkpSum) $\text{CM}(0, 0) = g^0 h^0 \cdot \prod_{i=1}^{N} \text{CM}(x_i, r_i)^{-\beta}$.
- (zkpMbs) $\text{CM}(z'_j, r'_j) = g^{z'_j} h^{r'_j} \cdot \left( \frac{\text{CM}(x, r)}{g^{r'_j}} \right)^{-\beta}$ for all $j \in \{1, \ldots, n\}$.
- (zkpNN) $\text{CM}(0, r') = h^{r'} \cdot \text{CM}(x, r)^{\beta} \cdot \prod_{i=1}^{n} \text{CM}(b_i, r_i)^{-\beta} 2^{i-1}$.

C SPDZ PROTOCOL

In the following, we present a simplified version of SPDZ for the clarity of exposition. The full version can be found in [13, 14].

There are three phases in SPDZ protocol: (1) pre-processing phase, (2) online phase, and (3) output and validation phase. We write $(x)$ as a secretly shared number, meaning that there is a vector $(x_1, \ldots, x_n)$, such that each party $i$ knows only $x_i$. To reveal secretly shared number $(x)$, each party $i$ broadcasts $x_i$ to other parties. Then each party can reconstruct $x = \sum_{i=1}^{n} x_i$. We write $(\langle x \rangle)$ meaning that both $\langle x \rangle$ and the respective MAC $(\langle y(x) \rangle)$ are secretly shared.

C.1 Online Phase

In the online phase, the parties can jointly compute an arithmetic circuit, consisting of additions and multiplications with secretly shared input numbers.

C.1.1 Addition

Given secretly shared $(x)$ and $(y)$, and a public known constant $c$, the following operations can be attained by local computation at each party, and then the outcome can be assembled from the individual shares:

A1) $(x) + (y)$ can be computed by $(x_1 + y_1, \ldots, x_n + y_n)$.
A2) $c \cdot (x)$ can be computed by $(c \cdot x_1, \ldots, c \cdot x_n)$.
A3) $c + (x)$ can be computed by $(c + x_1, x_2, \ldots, x_n)$.

C.1.2 Multiplication

Given secretly shared $(x)$ and $(y)$, computing the product $(x) \cdot (y)$ involves a given multiplication triple. A multiplication triple is defined by $((a, b, c))$, where $a, b$ are some unknown random numbers and $c = a \cdot b$, are three secretly shared numbers already distributed among the parties. The triple is assumed to be prepared in a pre-processing phase. To compute $(\langle x \rangle \cdot \langle y \rangle)$, it follows the below steps of operations (A4):

A4.1 Compute $(\langle e \rangle = \langle x \rangle - \langle a \rangle$ by (A1)). Then, reveal $(\langle e \rangle$, which does not reveal $x$.
A4.2 Compute $(\langle d \rangle = \langle y \rangle - \langle b \rangle$. Then, reveal $(\langle d \rangle$.
A4.3 Finally, compute $(\langle (x) \cdot (y) = (\langle c \rangle + e \cdot \langle b \rangle + d \cdot \langle a \rangle + e \cdot \langle d \rangle$ (by A1-A3).

C.1.3 Message Authentication Code

To safeguard against dishonest parties, who may perform incorrect computation, an information-theoretical message authentication code (MAC) can be used for verification. We write a MAC as a global number $\bar{a}$, which is unknown to the parties, and is secretly shared as $(\bar{a})$. Every secretly shared number is encoded by a MAC as $y(x) = \bar{a}x$, which is secretly shared as $(\langle y(x) \rangle)$. For each $(x)$, each party $i$ holds a tuple $(x_i, y(x_i))$, and $\bar{a}_i$, where $x = \sum_{i=1}^{n} x_i$, $\bar{a}_i = \sum_{i=1}^{n} \bar{a}_i$, and $y(x) = \bar{a}x = \sum_{i=1}^{n} y(x_i)$. If any party tries to modify her share $x_i$ uncoordinatedly, then he also needs to modify $y(x_i)$ accordingly. Otherwise, $y(x)$ will be inconsistent. However, it is difficult to modify $y(x_i)$ without coordination among the parties, such that $\bar{a} x = \sum_{i=1}^{n} y(x_i)$. Hence, it is possible to detect incorrect computation (possibly by dishonest parties) by checking the MAC.

To check the consistency of $x$, there is no need to reveal $(\bar{a})$. One only needs to reveal $(\langle e \rangle$, and then reveals $\bar{a}_i - x \cdot y(x_i)$, from each party $i$. One can check whether $\sum_{i=1}^{n} (\bar{a}_i - x \cdot y(x_i)) = 0$ for consistency. To prevent a dishonest party from modifying her share $x_i$ after learning other party’s $x_j$. Each party needs to commit her share $x_i$ before revealing $x_j$ to others.

To maintain the consistency of MAC for operations A1-A4, the MAC needs to be updated accordingly as follows:

B1) $(\langle x + y \rangle$: Update MAC by $(\langle y(x_1) + y(y_1), \ldots, y(x_n) + y(y_n))$.
B2) $(\langle c \cdot x \rangle$: Update MAC by $(\langle c \cdot y(x_1), \ldots, c \cdot y(x_n))$.
B3) $(\langle c + x \rangle$: Update MAC by $(\langle c \cdot y(x_1), \ldots, c \cdot y(x_n) + y(x_n))$.
B4) $(\langle x \cdot y \rangle$: Update MAC at each individual step of A4.1-A4.3 accordingly by B1-B3.

The additions and multiplications of $(\langle x \rangle$ and $(\langle y \rangle$ follow A1-A4 and the MACs will be updated accordingly by B1-B4.

To verify the computation of a function, it only requires to check the MACs of the revealed values and the final outcome, which can be checked all efficiently together in a batch at the final stage by a technique of called “random linear combination”.
C.2 Pre-processing Phase
In the pre-processing phase, all parties need to prepare a collection of triplets \((a, b, c)\) where \(c = a \cdot b\), each for a required multiplication operation. Assume that the parties hold secretly shared numbers \(a = \sum_{i=1}^{N} a_i\) and \(b = \sum_{i=1}^{N} b_i\) (which has been generated by local random generation). Note that \(a \cdot b = \sum_{i=1}^{N} a_i b_i + \sum_{i=1}^{N} \sum_{j=1}^{N} a_i b_j\), and \(a b_j\) can be computed locally. To distribute \(a b_j\), one can use partial homomorphic cryptosystems, with encryption function \(\text{Enc}\cdot\) and decryption function \(\text{Dec}\cdot\) using party \(i\)’s public and private \(K_p^i, K_s^i\). First, party \(i\) sends \(\text{Enc}_{K_p^i}[a_i]\) to party \(j\), who responds by \(C_i = b_j \text{Enc}_{K_p^j}[a_i] = \text{Enc}_{K_p^j}[\hat{c}_j]\), where \(\hat{c}_j\) is a random share generated by party \(j\) and is encrypted by party \(i\)’s public key \(K_p^i\). Then party \(i\) can obtain \(\hat{c}_j = \text{Dec}_{K_s^j}[C_i]\). Hence, \(a b_j = \hat{c}_i + \hat{c}_j\), which are secret shares \(a b_j\). The above generation assumes honest parties. To prevent cheating by dishonest parties, one would need to use proper zero-knowledge proofs before secret sharing [13, 14].

To generate a random mask \(\langle r^i \rangle\), each party \(j\) needs to generate a random share \(r_j^i\) locally. Then the parties follow the similar procedure of triplet generation to compute the secretly shared product \(\langle y(r^i) \rangle\), where \(y(r^i) = \hat{a} r^i\).

C.3 Output and Validation Phase
We describe random linear combination for batch checking. To check the MACs of a number of secretly shared numbers \(\langle x^1 \rangle, ..., \langle x^m \rangle\) in a batch, first generate a set of random \(\langle r^1 \rangle, ..., \langle r^m \rangle\), then reveal \(\langle x^1 \rangle, ..., \langle x^m \rangle\). Each party \(i\) computes \(\sum_{j=1}^{m} r^j (\tilde{a}_i - x^j \cdot y(x^j))\) and reveals it. All parties check whether \(\sum_{i=1}^{N} \sum_{j=1}^{m} r^j (\tilde{a}_i - x^j \cdot y(x^j)) \neq 0\) for consistency in a batch checking.

C.4 Protocol
We summarize the SPDZ protocol as follows:

1. **Pre-processing Phase**: In this phase, a collection of shared random numbers will be constructed that can be used to mask the private input numbers. For each private input number of party \(i\), there is a shared random number \(\langle r^i \rangle\), where \(r^i\) is revealed to party \(i\) only, but not to other parties. All parties also prepare a collection of triplets \(\langle (a), (b), (c) \rangle\) where \(c = a \cdot b, e\) each for a required multiplication operation.
2. **Online Phase**: To secretly shares a private input number \(x^i\) using \(\langle r^i \rangle\), without revealing \(x^i\), it proceeds as follows:
   1. Party \(i\) computes and reveals \(z^i = x^i - r^i\) to all parties.
   2. Every party sets \(\langle x^i \rangle \leftarrow z^i + \langle r^i \rangle\).

   To compute an arithmetic circuit, implement the required additions or multiplications by \(A1\)-\(A4\) and the MACs are updated accordingly by \(B1\)-\(B4\).
3. **Output and Validation Phase**: All MACs will be checked for all revealed numbers and the final output value. It can check all in a batch using random linear combination. If there is any inconsistency in the MACs, then abort.

Note that SPDZ cannot guarantee abort with fairness – dishonest parties may learn some partial values, even when the protocol aborts. However, this is a fundamental problem for any multi-party computation protocol with a majority of dishonest users, where dishonest parties are not identifiable when the computation is aborted.

D SECURITY ANALYSIS
We adopt the most common approach of security analysis in cryptography, based on the **Ideal-Real-Model Simulation paradigm** to prove and formalize the security achieved by our protocols. We next briefly describe the simulation paradigm. The detailed explanation can be found in the tutorial [29].

In the ideal model, all the parties send their private inputs to a trusted third party, who performs the prescribed computations and outputs the results to each party. The security requirements are already satisfied in the ideal model. The real model represents the realistic view of the privacy-preserving protocol. The security is defined by comparing what an adversary can learn in the real model to that in the ideal model. If what can be learned by an adversary in the real world can be totally simulated in the ideal world, then the adversary cannot learn more information in the real world than in the ideal world, we can say that a protocol \(\Pi\) is as secure as its corresponding ideal functionality \(F\). We give a formal definition of the security of our protocol as below:

**Theorem 4.** Assuming the discrete logarithm problem underlying the Pedersen commitment scheme is hard and the non-interactive zero-knowledge proofs are secure with access to a random oracle, in the \(\mathcal{F}_{\text{prep}}\)-hybrid model [14], the protocol \(\Pi_{\text{pess}}\) securely implements \(\mathcal{F}_{\text{pess}}\) with abort in the presence of an adaptive, active adversary in a dishonest-majority setting, if for every probabilistic polynomial-time (PPT) adversary \(A\) in the real model, there also exists a PPT adversary \(S\) in the ideal model, such that for each \(i \in N:\)

\[
\{\text{IDEAL}_{\mathcal{F}_{\text{prep}}, \Pi_{\text{pess}}, S} \}^\text{comp} \cong \{\text{REAL}_{\mathcal{F}_{\text{prep}}, \Pi_{\text{pess}}, A} \}
\]

where IDEAL and REAL respectively refer to the views and outputs of the corrupted and honest users in both ideal and real worlds.

We sketch the proof of the above theorem. Our aim is to demonstrate a simulator in the ideal model that can create a computationally indistinguishable view from that of the adversary in the real model. Even with a different set of honest-users’ inputs, the adversary should still be unable to tell computationally indistinguishable differences between the views. The simulator \(S\) externally interacts with the ideal functionality and internally runs a copy of the protocol \(\Pi_{\text{pess}} \circ \mathcal{F}_{\text{prep}}\) feeding messages to the adversary \(A\). However, it is a not trivial task for a simulator to emulate an adaptively malicious adversary, who is able to corrupt users at any time during the protocol. The challenge lies in the difficulty that it must produce a consistent view of the corrupted users throughout the protocol without knowing their inputs. Firstly, we define an ideal functionality \(F^{(1)}(a_t)\) in the stage 1 Pre-operation Scheduling computing the total energy demands \(a(t)\) before presenting the corresponding simulator \(S^{(1)}(a_t)\).

**Input:** On input \(\langle \text{input}, U_i, a_t(i) \rangle\), the functionality stores \(a_t(i)\).
Output: On input (output) from all honest users, the functionality computes and outputs \( a(t) = \sum_{i=1}^{N} a_i(t) \) to all the users.

Abort: On input (abort), the functionality outputs 0.

Initialize: The simulator S first calls \( F_{\text{prep}} \) to generate a sufficient number of multiplication triples and random numbers. Note that S has access to all the shares of the MAC key, random numbers and multiplication triples as it knows the decryption keys of public-key cryptosystem in the preprocessing phase. The adversary \( \mathcal{A} \) firstly corrupts a set of users, denoted by \( C \). Then the adversary may adaptively make corruptions on other users during the protocol.

Next, S produces \( g, h = g^k \in \mathbb{Z}_p^* \), where \( k = \log_q h \) is the trapdoor to Pedersen commitment, with which S is able to find out two pairs \((m, r), (m', r')\), such that \( Cm(m, r) = Cm(m', r') \).

Simulator \( S^{(1)}_{\text{at}}(t) \):

1. For honest users \( i \notin C \), S will simply generates dummy inputs \( \hat{a}_i(t) = 0, \hat{r}_i(t) \in \mathbb{Z}_p \) and reveals a commitment \( \hat{C}_i(t) = Cm(\hat{a}_i(t), \hat{r}_i(t)) \) with an nzkp\( \mathbb{NN}\{a_i(t)\} \). For the corrupted users \( i \in C \), S can extract their inputs with the knowledge of all the shares of \( \langle r \rangle \) and verifies the nzkp\( \mathbb{NN}\{a_i(t)\} \).

   Remarks: From the perspective of the adversary, the inputs of the honest users are indistinguishable from those in the real world due to the information-theoretically hiding properties of SPDZ secret-sharing (unless all the N shares are collected, the inputs cannot be reconstructed), and of the Pedersen commitment.

2. S firstly calls \( F^{(1)}_{\text{at}}(t) \) to obtain the output \( a(t) \). As S already computed an output \( \hat{a}(t) \) using dummy inputs of the honest users, it can respectively modify the share and MAC of a random honest user by adding \( a(t) - \hat{a}(t) \) and \( \sigma(a(t) - \hat{a}(t)) \) with the MAC key \( \alpha \) initialized in the preprocessing phase. Then S can perform the MAC check to evaluate and open \( a(t) \). If the check passes, S calls \( F^{(1)}_{\text{at}}(t) \) to output \( a(t) \) to all the users. Otherwise, S sends Abort to \( F^{(1)}_{\text{at}}(t) \).

   Remarks: No matter what inputs the adversary generates for the corrupted users, S can always create a computationally indistinguishable output distribution in the ideal model from that in the real model from the view of the adversary \( \mathcal{A} \). For the evaluation of \( a(t) \), each i-th share \( a_i(t) \) appears uniformly random to the adversary, which has exactly the same distribution in both ideal and real models.

After the simulator provided the simulated input \( \hat{a}_i(t), \hat{r}_i(t), \hat{C}_i(t) \) for \( i \notin C \), the adversary can corrupt an honest user \( U_i \) at any time. As aforementioned, the simulator must reveal its entire internal states, including the inputs, shares of inputs and random values that are consistent with the commitment \( \hat{C}_i(t) \) to simulate an adaptive adversary. It is easy to obtain the input \( a_i(t) \) from \( F^{(1)}_{\text{at}}(t) \). Regarding the random value, the simulator will take advantage of the trapdoor \( k \) of the Pedersen commitment to obtain \( r_i(t) = r_i(t) + (\hat{a}_i(t) - a_i(t)) \cdot k^{-1} \), such that \( \hat{C}_i(t) = Cm(a_i(t), r_i(t)) = Cm(\hat{a}_i(t), r_i(t)) \). Moreover, \( U_i \)’s share of her initial dummy input \( \hat{a}_i(t) \) is \( \hat{a}_i(t) + r_i - r \). Thus, it is trivial for the simulator to reveal the share \( a_i(t) + \hat{a}_i(t) - r + r_i \) by adding \( a_i(t) \). (See online phase in Section C.4).

Next, we give a brief description of the SPDZ-based zero-knowledge proofs \( \text{zkpCM}(a_i(t)) \) in \( \Pi_{\text{pess}} \) and \( \text{zkpSumCost}_{\text{ess}}(P_1(N)) \) in \( \Pi_{\text{pess}} \). For \( \text{zkpCM}(a_i(t)) \) and \( \text{zkpSumCost}_{\text{ess}}(P_1(N)) \) are collectively computed by all users and will be evaluated via MAC check to prove their correctness. A similar simulator to \( S^{(1)}_{\text{at}}(t) \) can be constructed to emulate the ideal functionality computing \( z_k(a_i(t)) \) and \( z_k(r_i(t)) \). The challenge \( \beta(t) \) is uniformly random independent of the prover’s input as it is obtained by summing the random values generated by all the users. Thus, this zero-knowledge proof is secure given the proof of completeness, soundness, zero-knowledge properties in Section B.5.

We skip the details for the simulator \( S^{(2)}_{\text{Cost}^{\text{ess}}} \) emulating ideal functionality computing the total payment \( \text{Cost}_{\text{ess}} = \sum_{t=1}^{N} P_t \) as it is similar to \( S^{(1)}_{\text{at}}(t) \) except using a different input \( P_t \).

E. ETHEREUM BLOCKCHAIN PLATFORM & SMART CONTRACTS

In this section, we provide a brief description of Ethereum blockchain platform and Solidity programming language as well as the details on the implementations of the smart contracts in our protocols.

E.1 Background

Bitcoin was the first widely adopted digital currency on a permissionless distributed ledger. Bitcoin relies on a tampering-resistant ledger based on cryptographic signatures. Tampering-resistance ensures integrity when the ledger is maintained by a network of peer-to-peer systems called “miners”. The miners are incentivized by cryptocurrency rewards for updating and validating the transaction records. Since the distributed ledger can be modified by multiple systems simultaneously, it is crucial to ensure consistency by a distributed consensus protocol among untrusted peer-to-peer systems, based on proof-of-work (by solving computational puzzles) or proof-of-stake (by demonstrating ownership of digital assets).

Subsequently, Ethereum was built on the Bitcoin ideas by expanding its functions to support general computing as smart contracts along with transactions. Bitcoin operates using a transaction-output-based system, called unspent transaction outputs (UTXOs), whereas Ethereum operates using accounts and balances in a manner called state transitions. Smart contracts, which are code programmed in high-level logic, will be compiled into byte code and executed in the virtual machine of miners. Miners will charge additional cryptocurrency payments called gas costs, because the extra computational tasks incurred by smart contracts will be broadcast throughout the blockchain. Smart contracts are implemented in a high-level programming language, such as Solidity [27].

It is worth noting that Bitcoin and Ethereum were only supposed to enable decentralization, but do not ensure privacy. In fact, the transaction histories of many cryptocurrencies are visible to the public. There are certain high-profiled prosecution of darknet operators based on the evidence of Bitcoin transactions. Supporting privacy in blockchain is a crucial on-going research topic.
E.2 Smart Contract Implementation

We next explain how Multi-Signature smart contract can achieve the step (5) and (6) of the stage Cost-sharing Payment by the following methods:

1. `submitTransaction()`. This method allows each user to submit the `zkpSum[Cost_{ess}, (P_i)_{i=1}^N]` that they have agreed upon off the chain. The method will compare whether users have submitted the same `zkpSum`.

2. `confirmTransaction()`. On one hand, this method allows each user to confirm that the stored `zkpSum` in the smart contract is the one that they have agreed upon off the chain. On the other, each user is required to submit a `nzkpNN[Bal(ad_i) - P_i]_{i=1}^N`, which will be validated to prove that there is sufficient balance in his account to pay for the energy cost.

3. `executeTransaction()`. This method can only be executed by the operator unless all the users have already confirmed the transaction. The method will validate the `zkpSum` before calling `ESToken` smart contract to credit `Cost_{ess}` to the operator’s account and debit the corresponding payment from each user’s account.

4. `secretlyJointTransfer()`. This method, defined within the `ESToken` smart contract is invoked by `executeTransaction()`, which actually performs the real transfer between multiple accounts.
Public review for

Towards Online Optimization for Power Grids

Deming Yuan, Abhishek Bhardwaj, Ian Petersen, Elizabeth Ratnam, Guodong Shi

The note provides a timely overview of distributed online optimization methods for power system application. It starts by presenting main methods for distributed optimization, and then reviews recent approaches of online convex optimization that have found many applications in machine learning. Finally it explains how they can be combined into distributed algorithms for online convex optimization and discuss opportunities and challenges for their application in power flow optimization.

This is a very timely topic for the decarbonization of our energy systems, for three reasons. First, as new technologies, such as energy storage and electric vehicles, are increasing deploy that create temporal dependence, independent static optimization at each time instance will become inadequate and sequential decisions must be optimized across time. Second, the increasing uncertainty in energy supply and demand makes precise forecast difficult, motivating online optimization. Finally, as smart distributed energy resources proliferate, it will become necessary, and possibly, to implement distributed online solutions.

Public review written by

Steven Low

California Institute of Technology, USA
Towards Online Optimization for Power Grids

DEMING YUAN, School of Automation, Nanjing University of Science and Technology, Jiangsu 210094, China
ABHISHEK BHARDWAJ, School of Engineering, Australian National University, ACT 2600, Australia
IAN PETERSEN, School of Engineering, Australian National University, ACT 2600, Australia
ELIZABETH L. RATNAM†, School of Engineering, Australian National University, ACT 2600, Australia
GUODONG SHI*, Australian Center for Field Robotics, The University of Sydney, NSW 2006, Australia

In this note, we discuss potential advantages in extending distributed optimization frameworks to enhance support for power grid operators managing an influx of online sequential decisions. First, we review the state-of-the-art distributed optimization frameworks for electric power systems, and explain how distributed algorithms deliver scalable solutions. Next, we introduce key concepts and paradigms for online optimization, and present a distributed online optimization framework highlighting important performance characteristics. Finally, we discuss the connection and difference between offline and online distributed optimization, showcasing the suitability of such optimization techniques for power grid applications.

1 DISTRIBUTED OPTIMIZATION FOR POWER GRIDS

1.1 The evolving power grid

The rise of distributed and renewable energy resources including wind and solar backed by energy storage technologies, is accelerating the evolution of the electric power grid [31]. The evolution is supported by recent advances in communication [15], sensor technologies [32], data processing [1], and operational technologies [7, 9, 10, 33], enabling prosumers to generate and deliver surplus renewable energy back to the power grid [4]. As the complexity of operating the evolving power grid continues to increase, with renewable technologies becoming increasingly distributed and spatially diverse, scalable approaches are needed to manage electricity flows to and from millions of energy prosumers. Distributed optimization frameworks that support scalability in managing renewable energy flows in transmission [23] and distributions networks [18, 29], that additionally supporting solutions to integrate energy storage [21, 30], including electrical vehicles [13, 26, 27], potentially enhance both the operation and resilience of the evolving power grid.

1.2 Distributed optimization frameworks

The key promise in distributed optimization is a dramatic improvement in scalability to accommodate data and decisions scattered in physically decentralized locations. See [24] for an in-depth survey.

Example 1. (Economic Power Dispatch [9]) Consider $N$ generators indexed in $\mathcal{V} = \{1, \ldots, N\}$. At a fixed time there is a total power demand $P$ that needs to be met by these $N$ generators. Let generator $i$ be allocated a power $x_i \in [x_i^{\min}, x_i^{\max}]$, leading to a cost $\ell_i(x_i)$. The economic power dispatch problem:

$$\min_{x} \sum_{i=1}^{N} \ell_i(x_i)$$

s.t. $x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, \ldots, N$

$$x_1 + \cdots + x_N = P.$$  \hfill (1)

Here $\ell_i(\cdot)$ is a function mapping from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$. An optimal decision on the $x_i$ for all $i$ should minimize the total generation cost.

Example 2. (Optimal Power Flow [10]) Consider an electrical network with $N$ nodes indexed in $\mathcal{V} = \{1, \ldots, N\}$. Let $v_i \in \mathbb{C}$ and $i_j \in \mathbb{C}$ be the voltage and inflow current at node $i$. The network structure is captured by an admittance matrix $A \in \mathbb{C}^{n \times n}$. Then $x_i := \text{Re}(v_i i_j^\dagger)$ defines the active power at node $i$, where $^\dagger$ is the complex conjugate. Let $\ell_i(x_i)$ denote the cost associated with the power at node $i$. An optimal power flow problem is given in the following form:

$$\min_{x} \sum_{i=1}^{N} \ell_i(x_i)$$

s.t. $x_i = \text{Re}(v_i i_j^\dagger), \quad i = 1, \ldots, N$  \hfill (2)

$$v_i i_j^\dagger = v_i \sum_{j=1}^{n} A_{ij} v_j^\dagger, \quad i = 1, \ldots, N.$$
1.3 Distributed optimization algorithms

There are many algorithms for distributed optimization. In power systems, the Alternating Direction Method of Multipliers (ADMM) has been popular.

1.3.1 The ADMM. Given an optimization problem

\[
\begin{align*}
\min_{x,z} & \ f(x) + g(z) \\
\text{s.t. } & \ Ax + Bz = c, \\
x & \in \mathbb{R}^n, \ z \in \mathbb{R}^m.
\end{align*}
\]

(3)

ADMM proceeds by first defining the augmented Lagrangian

\[L_\alpha(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \alpha/2 \|Ax + Bz - c\|^2\]

dual variable \(y\). Then the algorithm runs recursively, where in each round there are updates in the decision variable for \(x, z, y\) that are arranged sequentially.

**ADMM Algorithm [3]**

Define an initial point \((x(0), z(0), y(0))\), smoothing parameter \(\alpha\), and iteration limit \(n\).

For \(k = 0, \ldots, n\) DO

(i) Update the first variable as

\[
x^{(k+1)} = \arg \min_{x \in \mathbb{R}^n} L_{\alpha} \left( x, x^{(k)}, y^{(k)} \right);
\]

(ii) Update the second variable as

\[
z^{(k+1)} = \arg \min_{z \in \mathbb{R}^m} L_{\alpha} \left( x^{(k+1)}, z, y^{(k)} \right);
\]

(iii) Update the dual variable as

\[
y^{(k+1)} = y^{(k)} - \alpha \left( Ax^{(k+1)} + Bz^{(k+1)} - c \right).
\]

1.3.2 Distributed ADMM. The original ADMM algorithm was proposed in the 1970s [12, 14], and regained its popularity in recent years due to its suitability for large-scale distributed computing problems [3]. If we write \(f(x) = \sum_{i=1}^N \ell_i(x_i)\) for the cost functions in (1) and (2) with \(x = (x_1, \ldots, x_N)\), and suitable auxiliary decision variables \(z\) from the constraints, problems in the form (1) and (2) can be written in the standard ADMM form (3).

**Example 3.** The problem (1) with \(x^*_{l_{\min}} = -\infty\) and \(x^*_{l_{\max}} = \infty\) can be written as (Chapter 7, [3])

\[
\begin{align*}
\min_{x,z} & \ f(x) + g_p \left( \sum_{i=1}^N z_i \right) \\
\text{s.t. } & \ Ax_i - z_i = 0, \ i = 1, \ldots, N,
\end{align*}
\]

(4)

where \(g_p(a) = 1\) if \(a = P\) and \(g_p(a) = +\infty\) otherwise. Then, due to the separable nature of the function \(f\) and the constraints in (4), the resulting ADMM algorithm can be naturally decomposed into parallel computations at the agents along each primal variable \(x_i\) and dual variable \(y_i\) for Step (i) and Step (iii). The Step (ii) of the ADMM algorithm relies on all \(x_i\) and \(y_i\) for \(i = 1, \ldots, N\), and can be implemented in a distributed fashion with the help of the communication graph \(\mathcal{G}\).

1.3.3 Distributed consensus optimization. Another powerful approach in distributed optimization focuses on the following consensus optimization [25]:

\[
\min_x \sum_{i=1}^N \ell_i(x) \\
\text{s.t. } \ x \in \mathcal{X},
\]

(5)

Here the nodes hold local cost functions \(\ell_i(\cdot)\), but share a common decision variable \(x\). We may associate a doubly stochastic matrix \(W \in \mathbb{R}^{N \times N}\) (i.e., \(1^T W = 1^T\) and \(WW = 1\)) with the graph \(\mathcal{G}\) such that \(W_{ij} > 0\) if and only if \((j, i) \in \mathcal{E}\). In general, for strongly connected graph \(\mathcal{G}\), one can always find such an \(W\). Then the following iteration, termed average consensus algorithm,

\[
x_i(k+1) = \frac{1}{N} \sum_{j=1}^N W_{ij} x_j(k)
\]

drives all \(x_i(k)\) to \(\sum_{k=1}^N x_i(0) / N\) [22].

The insight of distributed consensus optimization lies in the ability to utilize the consensus algorithm as an information aggregation subroutine in decentralized optimization. This leads to the distributed gradient algorithm [25]

\[
x_i(k+1) = \mathbb{P}_\mathcal{X} \left( [\sum_{j=1}^N W_{ij} (x_j(k) - \gamma(k) \nabla \ell_j(x_j(k)))] \right)
\]

where \(\mathbb{P}_\mathcal{X}\) is the projection onto \(\mathcal{X}\), \(\gamma(k)\) is the step-size, and the gradient \(\nabla \ell_j(x_j(k))\) may also be replaced by sub-gradients. With proper convexity on the \(\ell_i\) and well selected step-size, the algorithm drives all \(x_i(k+1)\) to a common state as a solution to (5). Distributed consensus optimization can also be used to solve problems in the form of (1) and (2).

**Example 4.** Consider the setting of Economic Power Dispatch as in Example 1, and recall the total power demand \(P\). Introduce \(P_i\) for node \(i\) with \(\sum_{i=1}^N P_i = P\). Define

\[
g_i(\lambda) : = \min_{x_i \in [x^*_{l_{\min}}, x^*_{l_{\max}}]} \ell_i(x_i) - \lambda (x_i - P_i).
\]

Then from the Lagrange duality, the problem (1) can be equivalently written as

\[
\max_{\lambda} \sum_{i=1}^N g_i(\lambda) \\
\text{s.t. } \ \lambda \in \mathbb{R}.
\]

(6)

Here equivalence means that any optimal solution \(\lambda^*\) for (6) will lead to \(x^*_{l_{\min}}, \ldots, x^*_{l_{\max}}\) as optimal solutions to (1), when \(\lambda^*\) is substituted into each \(g_i(\lambda)\) and the corresponding optimal \(x^*_{l}\) is found [34].

1.3.4 Other distributed optimization approaches. There are many other distributed optimization methods. Some popular algorithms based on the augmented Lagrangian technique are analytical target cascading [8], auxiliary problem principle [5], dual decomposition method [3], and proximal message passing [19]. One can also move away from the augmented Lagrangian, and use optimality condition decomposition [6], or dynamic programming [2].
2 ONLINE CONVEX OPTIMIZATION

2.1 Online optimization

The online optimization paradigm applies a robust optimization perspective for sequentially arriving data and costs that are too complex to be efficiently modeled. With its roots in classical ideas of sequential decisions in multi-armed bandit problems from the 1930s, online optimization has recently emerged as a prominent tool in machine learning, solving problems ranging from recommender systems to spam filtering [16, 28]. Online optimization portrays decisions for optimizing time-varying cost functions as a feedback process, where one learns from experience as time evolves. Performance is considered with respect to a static optimal decision taken in hindsight. Formally, the procedure of online optimization may be described as a game between a learner and an adversary played across a finite time horizon $t = 1, \ldots, T$.

### Online Optimization Paradigm [28]

For $t = 1, \ldots, T$, DO

(i) The adversary selects a cost function $\ell_t(\cdot) : \mathcal{X} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ and keeps it to itself;
(ii) The learner makes a decision $x_t \in \mathcal{X}$;
(iii) The learner suffers a loss $\ell_t(x_t)$, and receives the cost function $\ell_t(\cdot)$ (full information), or just value of the loss $\ell_t(x_t)$ (bandit information).

In sharp contrast to the view of classical optimization (i.e. a classical learner), where the loss function $\ell_t(\cdot)$ is revealed before the learner attempts to minimize it, online optimization acknowledges the difficulty in knowing $\ell_t(\cdot)$ or even a model of it before decisions are made. The information that the learner receives about $\ell_t(\cdot)$ may be the whole function, a scenario referred to as full information; or the learner only experiences losses at selected decisions, and in this case, we talk about bandit information. The loss functions $\ell_t(\cdot)$ are generally assumed to be arbitrary (but chosen from a given function class). Hence, it is impossible for the learner to infer $\ell_t(\cdot)$ before the decisions are made. As a result, it is sensible for the learner to identify $x_1, \ldots, x_T \in \mathcal{X}$ so that the regret, i.e.,

$$\text{Reg}(T) := \sum_{t=1}^{T} \ell_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x)$$

is minimized. From the definition, $\min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x)$ is the minimal accumulative loss of an oracle making a static decision to whom all $\ell_t(\cdot)$ are known before $t = 1$. Therefore, Reg$(T)$ represents the difference between the actual accumulative loss experienced by the learner compared to that of such an oracle, i.e., the regret.

2.2 Impact of feedback

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a compact convex set containing the origin, for which $P_{\mathcal{X}}$ is the projection onto $\mathcal{X}$. Simple yet effective algorithm for the online learner is gradient descent implemented sequentially. The standard online gradient descent algorithm for solving the online optimization problem with full information is described below where $\alpha_t$ is the stepsize.

### Online Gradient Descent: Full Information Feedback [16]

For $t = 1, \ldots, T$, DO

(i) The adversary selects a cost function $\ell_t(\cdot) : \mathcal{X} \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ and keeps it to itself;
(ii) The learner makes a decision $x_t = P_{\mathcal{X}}(x_{t-1} - \alpha_t \nabla \ell_{t-1}(x_{t-1}))$;
(iii) The learner suffers a loss $\ell_t(x_t)$, and receives $\ell_t(\cdot)$.

With bandit information, the learner only experiences losses and the loss function $\ell_t(\cdot)$ (and its gradient) is still unknown. Denote $\mathcal{X}_S = \{x : \frac{1}{\|x\|} x \in \mathcal{X}\}$. Let $\mathcal{S}$ be the unit sphere in $\mathbb{R}^d$ under standard Euclidean norm. Then one can build unbiased gradient estimates from experienced losses to replace the true gradients in online gradient descent, leading to the following online bandit optimization algorithm.

### Online Bandit Optimization: Bandit Information Feedback [11, 16]

Initialize $y_0 = 0$. For $t = 1, \ldots, T$, the learner DO

(i) Draw $u_t \in \mathcal{S}$ uniformly at random.
(ii) Play $x_t = y_t + u_t$; receive loss $\ell_t(x_t)$.
(iii) Build gradient estimate $g_t = d \nabla \ell_t(x_t) u_t / \delta$; Update $y_{t+1} = P_{\mathcal{X}_S}(y_t - \eta g_t)$.

It is immediately clear that in both (7) and (8), feedback is taking place. The promise of these online optimization algorithms lies in the fact that, when the stepsizes are selected as some suitable learning rates, the algorithms will produce sub-linear regrets $\lim_{T \to \infty} \text{Reg}(T)/T = 0$, for a suitably regular classes of convex cost functions. The regret averaged over time is close to zero for sufficiently long time horizon: it is as if all the $\ell_t(\cdot)$ are known before the whole play starts and the learner decides to play a static optimal decision. With careful classification of the function classes for the loss functions, refined upper bounds on Reg$(T)$ can be established at $O(\log T)$, $O(\sqrt{T})$, $O(T^{3/4})$, etc [16].

2.3 Distributed online optimization

In practice, the loss $\ell_t(\cdot)$ might represent a system-level loss, scattered across a number of subsystems indexed in $V = \{1, \ldots, N\}$, such that $\ell_t(\cdot) = \sum_{i=1}^{N} \ell_{ti}(\cdot)$. The overall system forms a network, where a directed graph $\mathcal{G} = (V, \mathcal{E})$ describes the communication structure of the network. Then the question arises on whether in the case that subsystems may only talk to their neighbors over the graph $\mathcal{G}$, this will enable distributed online learning throughout the network. Following the distributed optimization framework, a distributed online optimization paradigm can be described as follows [37] (see also [17, 35, 36, 38]).

---

1For bandit feedback, the regret is technically $\mathbb{E} \text{Reg}(T)$ where the expectation is taken over the randomness in the gradient estimate.
Distributed Online Convex Optimization [37]

Initialize $\mathcal{X}$ as a convex subset of $\mathbb{R}^d$ defined by a family of inequalities: $\mathcal{X} = \{ x \in \mathbb{R}^d | c_s(x) \leq 0, s = 1, \ldots, p \}$.

For $t = 1, \ldots, T$, agents in $\mathcal{V}$ DO
- Each agent $i \in \mathcal{V}$ selects $x_i(t) \in \mathcal{X}$, and a local adversary chooses $\ell_{i,t}(\cdot) : \mathbb{R}^d \to \mathbb{R}$ as a convex cost function;
- Each agent experiences a loss $\ell_{i,t}(x_i(t))$;
- The function $\ell_{i,t}$ is revealed to agent $i$; The decisions of the neighbors of the agent $i$ are also revealed to $i$, i.e., $x_j(t)$ for $j \in N_i := \{(j, i) \in \mathcal{E}\}$.

The decision set $\mathcal{X}$ implies that in each time-step each agent should be able to perform a projection onto $\mathcal{X}$, which can be computationally expensive. Instead, one can only require that the constraints are satisfied in the long run, i.e., that $\sum_{t=1}^T \sum_{j \in \mathcal{V}} c_s(x_i(t)) \leq 0$. An effective distributed online learning algorithm, then should aim to minimize the accumulated system-wide loss. The system-level regret is defined as the worst possible regret for all agents:

$$\text{SReg}(T) := \max_{i \in \mathcal{V}} \left[ \sum_{t=1}^T \sum_{s=1}^p \ell_{i,t}(x_i(t)) - \sum_{t=1}^T \sum_{s=1}^p \ell_{i,t}(x^*) \right]$$

(9)

where $x^* = \arg\min_{x \in \mathcal{X}} \sum_{t=1}^T \sum_{j \in \mathcal{V}} \ell_{j,t}(x)$ is the system-level decision by a static optimal oracle. The performance of the algorithm is further characterized by the so-called cumulative absolute constraint violation defined by

$$\text{CACV}(T) := \sum_{t=1}^T \sum_{i=1}^N \sum_{j=1}^p \left[ c_s(x_i(t)) \right]_+$$

(10)

where $[a]_+ = \max\{0, a\}$.

2.4 Distributed online primal-dual gradient algorithm

In [37], distributed online primal-dual gradient algorithms with full and bandit information feedback were proposed. The Lagrange dual function for each node $i$ at time $t$ is defined as

$$\mathcal{L}_{i,t}(x, \lambda) := \ell_{i,t}(x) + \sum_{s=1}^p [\lambda_s]_+ c_s(x) - \frac{\eta_t}{2} \|\lambda\|^2.$$ 

(11)

Here $\lambda = (\lambda_1, \ldots, \lambda_p)^T \in \mathbb{R}^p$ is the vector of Lagrangian multipliers with $|\lambda_s|_+$ being associated with the inequality constraint $c_s(x) \leq 0$. We present the pseudo code for the full information feedback algorithm below. The bandit algorithm is an extension of the full information algorithm, where the gradient information is replaced by estimations from the experienced losses.

Distributed Online Primal-Dual Gradient Algorithm: Full Information Feedback [37]

Initialize $x_i(1) = 0, \lambda_i(1) = 0, \forall i = 1, \ldots, N$.

For $t = 1, \ldots, T$, agents in $\mathcal{V}$ DO

(i) Agent $i \in \mathcal{V}$ commits to a decision $x_i(t)$, and then after receiving $\ell_{i,t}(\cdot)$, computes

$$y_i(t) = x_i(t) - \beta_t \left[ \nabla \ell_{i,t}(x_i(t)) + \sum_{s=1}^p [\lambda_i(t)_s]_+ c_s(x_i(t)) \right].$$

(ii) Agent $i$ communicates $y_i(t)$ to its neighbors and updates its decision as

$$x_i(t + 1) = \mathbb{A}_B \left( \sum_{j=1}^N W_{ij} y_j(t) \right)$$

where $\mathbb{A}$ is a ball that contains $\mathcal{X}$.

(iii) Agent $i$ updates its dual variable:

$$\lambda_i(t + 1) = \arg\max_{\lambda} \mathcal{L}_{i,t}(x_i(t + 1), \lambda).$$

Here in Step (ii) of the algorithm, the projection is made onto $\mathcal{X}$ instead of onto $\mathcal{X}$, therefore reducing the computational complexity. With full information feedback, upper bounds on $\text{SReg}(T)$ and $\text{CACV}(T)$ at scales of $O(T^{(1-c)/3})$ and $O(T^{1-c/2})$ can be established for convex losses; these regret bounds can be improved to $O(\log(T))$ and $O(\sqrt{T} \log(T))$ for strongly convex losses [37]. With bandit information feedback, regret and constraint violation are bounded above by $O(d^2 T^{(1-c)/3})$ and $O(d T^{1-c/2})$ for any $c \in (0, 1)$ for convex losses, which turn out to be $O(d^2 T^{2/3} \log(T))$ and $O(d \sqrt{T} \log(T))$ in the case of strongly convex losses [37]. The user-defined parameter $c$ tunes the trade-off between $\text{SReg}(T)$ and $\text{CACV}(T)$.

We illustrate the performance of the algorithms using a simple experiment. The code for reproduction of the numerical results can be found at https://github.com/dyuan21/EIR_DOCCO.

Example 5. Consider a distributed online linear regression problem over a network, where $\ell_{i,t}(x) = (a_i(t)^T x - b_i(t))^2/2$. The constraints are described as

$$c_{m}(x) = L - x_m \leq 0, \quad m = 1, \ldots, d,$$

$$c_{d+m}(x) = x_m - U \leq 0, \quad m = 1, \ldots, d.$$ 

(12)

(13)

Every entry of $a_i(t)$ and $b_i(t)$ is generated uniformly at random within the interval $[-1, 1]$ and $[0, 1]$, respectively, independently for each time $t = 1, \ldots, T$. Throughout the experiments, we implement the distributed online primal-dual gradient algorithms proposed in [37] with full information or bandit information feedback.

System Setup. The graph $G$ is randomly generated and selected as depicted in Fig. 1. The weighting matrix associated with the network in Fig. 1 is generated according to the maximum-degree weights: $W_{ij} = \frac{1}{d_{\max}} (i, j) \in \mathcal{E}; W_{ii} = 1 - \frac{d_i}{d_{\max}}; W_{ij} = 0, (j, i) \notin \mathcal{E}$. Here $d_{\max} = \max_{i \in \mathcal{V}} \{d_i\}$ is the maximum degree of $\mathcal{G}$ with $d_i$ denoting the degree of node $i$. We set the parameters as follows: $N = 20,$
\[ d = 2, L = -\frac{1}{2}, U = \frac{1}{2}, \text{ and } R_X = U \sqrt{d}. \] The performance of the algorithm is averaged over 10 runs.

**Fig. 1.** A randomly generated network of 20 nodes.

**Performance.** We run the algorithms and plot the Average System Regret (ASR, for short) defined as \( \frac{\text{SReg}(T)}{T} \) and the Average Constraint Violations (ACV, for short) defined as \( \frac{CACV(T)}{T} \), as a function of the time horizon \( T \) in Fig. 2. Clearly both the ASR and ACV converge to zero.

**Fig. 2.** ASR and ACV vs. time for the distributed online optimization algorithms with full information and bandit information in [37].

**Decision Stationarity.** At time \( t \), let the system-level objective function \( \sum_{i=1}^{N} \frac{1}{2} (a_i(t)^T x - b_i(t))^2 \) yield an optimal decision \( x^*_i(t) \). In Fig. 3, we plot the first entry of the repeated offline optimizer \( x^*_1(t) \), and the first entry of the distributed sequential online optimizer \( x_1(t) \) and \( x_2(t) \) for agent 1 and agent 2, respectively. It can be seen that the online agent decisions demonstrate significantly reduced fluctuations compared to the repeated system-level offline decisions.

**Fig. 3.** System-level optimal decisions from repeated offline optimization vs. distributed agent decisions from sequential online optimization.

## 3 PERSPECTIVES ON ONLINE OPTIMIZATION FOR THE POWER GRID

### 3.1 Offline vs. online optimization

For problems such as the economic dispatch and optimal power flow as in Example 1 and Example 2 over a time horizon \( t = 1, \ldots, T \), there may be two paradigms.

**Distributed (Repeated Offline) Optimization [DRO-O].** For each time \( t \), independently treat the corresponding problem (1) and (2); apply distributed optimization algorithms until suitable convergence is guaranteed for time \( t \); implement the optimal decision \( x^*_i(t) \) for \( i = 1, \ldots, N \) repeatedly at the respective time \( t \).

**Distributed Online Optimization [DO-O].** Employ the Distributed Online Convex Optimization paradigm outlined in Section 2.3; apply distributed online optimization algorithms throughout the horizon \( t = 1, \ldots, T \); implement the decision \( x_i(t) \) for \( i = 1, \ldots, N \) sequentially for \( t = 1, \ldots, T \).

The potential in developing online optimization frameworks for problems in power grids has drawn attention in the literature. Online convex optimization has been adopted in [40] for the control of distributed energy sources in the context of social welfare maximization. Under a similar social welfare maximization paradigm, [39] considers control of distributed energy sources with both continuous and discrete constraints. Moreover, [20] provides a unified framework for economic dispatch and unit commitment and proposes a centralized and distributed online convex optimization method for exploring such a framework.

Next, we would like to offer a few perspectives towards the strengths, challenges, and possible future direction for online optimization in power grids.

### 3.2 Perspectives between [DRO-O] and [DO-O]

First, it is worth mentioning that the key difference between [DRO-O] and [DO-O] goes far beyond the respective classes of algorithms.
Underpinning the two frameworks are fundamentally different views about the system:

- In [DRO-O], the time-varying cost functions are known before decisions;
- In [DO-O], the time-varying cost functions are experienced after decisions.

As a result, conceptually the [DRO-O] algorithms are optimizers, while the [DO-O] algorithms are learners. Therefore, [DO-O] suits systems that are uncertain or unpredictable.

Next, the strength of [DO-O] lies in guaranteed sub-linear regret against adversaries. In practice, the adversaries represent the worst-case scenarios. Remarkably, the aforementioned regret bounds of orders $O(\log T)$, $O(\sqrt{T})$, $O(T^{3/4})$ may be valid even for feedback adversaries, where the cost function $\ell_t(\cdot)$ depends on the past experiences. Moreover, the online decisions in [DO-O] tend to converge to a static optimal decision with respect to the cumulative cost over the entire horizon, while decisions in [DRO-O] tend not to converge as they are tracking real-time optimal decisions for time-varying cost functions. This is shown in Example 5, where online decisions indeed are more stationary compared to repeated offline optimal decisions.

In the context of power grids, [DO-O] might be more suited in problems related to wind or solar energy grid-integration, and energy storage applications including residential batteries and EVs. Such applications involve significant uncertainty regarding the weather, network impedance and topology, real-time price volatility, and user preferences including when, where and for how long an EV will require charging. Importantly, for problems with known grid and user information, [DRO-O] is a more sensible choice as the performance of [DO-O] is much more conservative.

Finally, we remark that while the long-term constraint violation metric reduces the computational complexity associated with distributed optimization, it also allows the constraints to be violated during the transient. Such violations may cause safety concerns or lead to decisions that are not realizable in practical power grids. As a result, it is important to evaluate the feasibility of the constraints in terms of hard of soft constraints, before deciding on using the long-term constraint violation metric or not. One can remedy this in the online primal-dual gradient algorithms by projecting onto the feasible set $X$. This modification will preserve the bounds on the regret $\mathcal{R}(T)$, of course now at a higher computational cost.

### 3.3 Future directions

Towards establishing practical online optimization frameworks for problems in power grid, there are a few possible directions. First, the notion of regret needs to be taken into account for online optimization of power grid problems. Existing regret bounds for online optimization are for classes of convex, smooth, or strongly convex functions, etc. Cost functions in power grid problems and constraints are certainly more structured (despite being unknown before decisions), and thus refined regret bounds might exist. Second, the uncertain nature of online optimization needs to be carefully matched to practice. The characterization of cost functions should also be evaluated in the power system context. Third, hybrid decision frameworks that combine the strength of [DRO-O] and [DO-O], where the information and uncertainty of the cost functions can be jointly treated, would be of significant value for power grid applications.

### ACKNOWLEDGMENTS

We wish to thank Professor Alexandre Proutiere for insightful feedback and suggestions. This work was supported by the Australian Research Council under Grants DP180101805 and DP190103615.

### REFERENCES


Public review for

Online Peak-Aware Energy Scheduling with Untrusted Advice

Russell Lee, Jessica Maghakian, Mohammad Hajiesmaili, Jian Li, Ramesh Sitaraman, Zhenhua Liu

This paper studies an online energy scheduling problem in a hybrid pricing model with volume pricing and peak pricing. The demand of energy customers is uncertain. A main contribution of the paper is an online algorithm with an add-on machine learning (ML) advice. Specifically, energy can be either locally generated or drawn from the grid. The paper uses a break-even point to balance the local generator costs and grid costs and absorbs the possible choices on the local generator and the grid. The peak price of the grid is unknown, and the paper uses ML to predict, with a prediction-robustness factor for quantification; this makes the ML advised online algorithm analyzable. The paper evaluates the algorithms using real-world data center traces from Akamai.

The paper is rigorous and clearly written. Online energy scheduling is an important problem and the way that this paper makes the online algorithm with ML advice analyzable is interesting. It may lead to follow up studies in the future.

Public review written by

Dan Wang

Hong Kong Polytechnic University, Hong Kong
Online Peak-Aware Energy Scheduling with Untrusted Advice

RUSSELL LEE, UMass Amherst
JESSICA MAGHAKIAN, Stony Brook University
MOHAMMAD HAJIESMAILI, UMass Amherst
JIAN LI, Binghamton University-SUNY
RAMESH SITARAMAN, UMass Amherst and Akamai Tech
ZHENHUA LIU, Stony Brook University

This paper studies the online energy scheduling problem in a hybrid model where the cost of energy is proportional to both the volume and peak usage, and where energy can be either locally generated or drawn from the grid. Inspired by recent advances in online algorithms with Machine Learning (ML) advice, we develop parameterized deterministic and randomized algorithms for this problem such that the level of reliance on the advice can be adjusted by a trust parameter. We then analyze the performance of the proposed algorithms using two performance metrics: robustness that measures the competitive ratio as a function of the trust parameter when the advice is inaccurate, and consistency for competitive ratio when the advice is accurate. Since the competitive ratio is analyzed in two different regimes, we further investigate the Pareto optimality of the proposed algorithms. Our results show that the proposed deterministic algorithm is Pareto-optimal, in the sense that no other online deterministic algorithms can dominate the robustness and consistency of our algorithm. Furthermore, we show that the proposed randomized algorithm dominates the Pareto-optimal deterministic algorithm. Our large-scale empirical evaluations using real traces of energy demand, energy prices, and renewable energy generations highlight that the proposed algorithms outperform worst-case optimized algorithms and fully data-driven algorithms.

CCS Concepts: • Theory of computation → Online algorithms; Linear programming; • Hardware → Energy generation and storage; Power estimation and optimization.

Additional Key Words and Phrases: Machine learned advice, online algorithms, renewable generation, Pareto-optimality

1 INTRODUCTION

The electricity bill for large energy customers is usually based on the following two terms: (1) the volume pricing, which is the aggregate energy usage over the cycle, i.e., \( \sum_t p(t)e(t) \), where \( p(t) \) is the real-time unit price at \( t \), and (2) the peak pricing, which is the peak demand drawn over the cycle, i.e., \( \max_{t \in [T]} e(t) \), multiplied by \( p_m \) as the peak price. The contribution of peak pricing in the electricity bill is usually substantial. The peak price is often more than 100 times higher than the maximum spot price, e.g., 118x for PG&E or 227x for Duke Energy Kentucky. Hence, the contribution of peak charging in the energy bill for large energy customers can be considerable, e.g., from 20% to 80% for Google data centers [31].

A promising approach to reduce the contribution of peak charges in the final electricity bill is to install on-site generation units that can “shave the peak” by covering that portion of the demand [34]. A notable example is Microsoft’s plan to add 72 new generators at its Quincy, Washington data center campus [14]. The global market for on-site generators is growing and expected to reach a revenue of around $5 billion in 2023 [2]. With an on-site generator, one can schedule its generation such that part of the total energy demand is satisfied by the local generator, hence, the peak net demand from the grid is reduced over the billing cycle.

However, peak-aware energy generation scheduling of local generators is a challenging problem due to the uncertainty of the demand of energy customers, especially in data centers. For data centers the energy demand is highly unpredictable because user demand for internet services is variable. For instance, a data center serving videos to users can experience an unexpected flash crowd of users for a popular video release. Furthermore, sophisticated optimization algorithms are being used in Google data centers to improve the energy efficiency of data center’s internal operations [10], which can further increase the variability of energy demand. In geographical load balancing schemes [17, 20], a global load balancer could move user demand into or out of the data center, resulting in unexpected changes in the energy patterns. Lastly, the integration of renewables into data centers provides even more uncertainty, since the production level of renewables is uncertain and intermittent [11].

The peak-aware energy generation scheduling problem (henceforth PAES) has been tackled using the competitive online framework [34]. More specifically, two deterministic and randomized algorithms have been proposed that can achieve the best competitive ratio as the well-established performance metric for online algorithms [4]. Competitive ratio is defined as the ratio between the cost of an online algorithm and that of the offline optimal for the worst-case over all feasible instances to the problem. The competitive online framework, however, aims to be provably efficient
motivation is two-fold: (1) to keep the core competency of online algorithms, i.e., performance guarantee against the worst-case; and (2) to achieve a provably improved performance if the accuracy of ML-predictor is satisfactory. The two motivations could be analyzed for learning-assisted online algorithms [22, 25] by introducing the notions of (1) robustness that characterizes the first motivation; and (2) consistency that characterizes the second one.

Specifically, suppose that \( \mathcal{A} \) is a learning-assisted online algorithm that leverages an ML-predictor in decision making. The algorithm \( \mathcal{A} \) is \((\alpha, \gamma)\) -competitive where \( \alpha \) and \( \gamma \) represent the robustness and consistency of \( \mathcal{A} \), respectively. That is, the competitive ratio of \( \mathcal{A} \) is always less than \( \alpha \) regardless of the error in ML-predictor. Also, \( \mathcal{A} \) is \( \gamma \)-consistent if with perfect predictions it achieves the competitive ratio of \( \gamma \). Robustness measures how well the algorithm does in the worst-case of poor predictions, while consistency measures how well the algorithm does under perfect predictions. In this framework, the performance of an algorithm is evaluated using two criteria, i.e., robustness and consistency. Hence, investigating the optimality of an algorithm naturally leads to the consideration of Pareto optimality. Therefore, the eventual goal in this setting is to design an algorithm \( \mathcal{A} \) that is Pareto-optimal, meaning that there is no other algorithm that can achieve a better consistency (resp., robustness) than \( \mathcal{A} \) without sacrificing the robustness (resp., consistency).

With this analytical framework, one is able to achieve “the best of both worlds” paradigm from the perspective of learning-assisted competitive algorithms. While it might slightly degrade the robustness against worst-case, or ideally maintain the worst-case guarantee, it resolves the fundamental drawback of competitive analysis of pessimistic decision making by incorporating ML predictions. More importantly, unlike classic prediction-based competitive designs [5, 7, 8, 13], the framework used in this paper leverages a trust parameter that determines how much the algorithms trust the predictors, enabling the full spectrum coverage from pure worst-case to fully prediction-based decision making.

1.3 Summary of Contributions

Inspired by the above direction of learning-assisted algorithm design, we develop deterministic and randomized algorithms for PAES that take into account advice from an ML model in decision making. This paper makes the following contributions, with a summary of theoretical results outlined in Table 1.

First, we propose \( \text{OnMLEng} \), a deterministic algorithm parameterized by a trust parameter \( \lambda \in [0, 1] \), that achieves a competitive ratio of \( 1 + (1 - \beta)/\lambda \), where \( \beta \in (0, 1) \) is a problem-specific parameter that determines the ratio between the unit price of the grid and local generator. We show that \( \text{OnMLEng} \) is \((1 + (1 - \beta)/\lambda)\)-robust and \((1 + \lambda(1 - \beta))\)-consistent. The trust in ML prediction is interpreted as follows. Greater trust in ML prediction is achieved by setting \( \lambda \) close to 0, which means that \( \text{OnMLEng} \) is 1-consistent, i.e., it achieves the optimal performance with perfect advice. On the other hand, less trust in ML advice is achieved by setting \( \lambda \) close to 1, and the robustness result guarantees the optimal online competitive ratio of \( 2 - \beta \) as in [34]. More importantly, we show that \( \text{OnMLEng} \) is...
We consider the scenario where the energy demand can be covered by Akamai data centers. These distribution functions are customized based on Yao’s principle (NYISO). The results show the improved performance of our proposed online algorithms with ML advice as compared to the purely online algorithm. We also investigate the impact of several parameters and provide insights that reveal the practical benefits of learning-assisted algorithms.

2 PROBLEM STATEMENT

2.1 System Model

We consider the scenario where the energy demand can be covered by either local generators or the external grid. The peak-aware energy scheduling problem (PAES) aims to prudently choose the source of energy, so that the energy demand can be met at each time step while the total cost is minimized.

We focus on one billing cycle $T = \{1, \cdots, T\}$ with $T$ discrete time slots of uniform length. The billing cycle is usually one month and the length of each slot is 5 minutes. Let the energy demand in slot $t$ be $e(t)$ and $e = \{e(t)\}_{t \in T}$. We consider an online scenario in which the values of demand are unknown for future slots. The demand can be covered by two sources, the local generator and external grid. The local generator can satisfy at most $C \geq 1$ KWs of demand in each slot, with cost $p_g$. In reality, some traditional generators [21] have maximum ramp-up and ramp-down constraints that limit the change of output between two adjacent slots. In this paper, we focus on “fast-responding” generators that can ramp up and down without any limit. In experiments (§6.3.3), we investigate the impact of ramp constraints.

We consider a typical energy cost model for industrial energy customers that follows a hybrid charging model that has both total usage (a.k.a. energy charge) and peak usage (a.k.a. demand charge) components. The energy cost is the sum of the following two terms: (1) the usage-based pricing, which is the total energy usage over the cycle, and (2) the peak pricing, which is the peak demand drawn over the cycle. Following the dynamics of the energy market, the grid provides electricity with a spot price $p(t)$ at time $t$, where we assume $p(t) \geq p_{\text{min}} > 0$. In reality, the unit cost of local generators $p_g$ is usually higher than that of external grid, i.e., $p_g \geq p(t)$. Otherwise, it is always optimal to use local generators as much as possible for both online and offline algorithms. However, the expensive local generator can shave the peak demand (peak charge) of the external grid. In addition, $p_m$ is the peak charge price that is known and fixed over the billing cycle. Note that $p_m$ is usually more than 100 times larger than $p(t)$. For ease of exposition, denote $\beta \triangleq \frac{p_{\text{min}}}{p_g} < 1$ as the ratio between the minimum grid price and the unit cost of local generation. We characterize the performance of our algorithms as a function of $\beta$.

2.2 Problem Formulation

Let $v(t)$ and $u(t)$ be the optimization variables that determine the amount of electricity procured from the external grid and local generator, respectively. For the grid, its cost consists of volume charge and peak charge. The volume charge is the sum of the maximum single-slot power and peak price $p_m$, i.e., $p_m \max_T {v(t)}$ [31, 34]. The cost of using local generators, is $\sum_t p_g u(t)$. Therefore, with $u = \{u(t)\}_{t \in T}$ and $\alpha = \{\alpha(t)\}_{t \in T}$, the PAES problem is defined as

$$\text{PAES}: \quad \min_{u, \alpha} \sum_{t \in T} p(t) v(t) + p_m \max_T v(t) + \sum_{t \in T} p_g u(t)$$

s.t.,

$$u(t) + v(t) \geq e(t), \quad t \in T,$$

$$u(t) \leq C, \quad t \in T,$$

$$v(t) \geq 0, u(t) \geq 0, \quad t \in T,$$

where the first constraint ensures that the demand is satisfied, and the second constraint is due to the generator capacity limitation. We note that in our algorithm design we focus on a basic version of PAES, where the demand $e(t)$ only takes binary values 0 or 1. Our algorithms and competitive analysis, however, could be extended to the general case as discussed in Section 4.6.

PAES with $e(t)$ and $p(t)$ values known in advance is a linear program. Hence, it can be solved using any linear programming algorithm. However, in practice $e(t)$ and $p(t)$ are unknown in advance and hard to predict, hence an online approach is required. We
use a recently proposed algorithmic framework [22, 25] for devising online algorithms with advice, and provide a brief overview of the framework in the following section.

3 ALGORITHMIC FRAMEWORK

In this section, we give an overview of the recently proposed framework for designing competitive algorithms with ML advice [22, 25].

3.1 Online Algorithms with ML Advice

In this framework the goal is to utilize ML advice to improve the performance of online algorithms, both in theory and practice. Toward this, it is assumed that there is advice from an untrusted ML model that might be subject to error or even vulnerable to malicious activities. The goal is to develop online algorithms that are able to determine the level of trust in the ML advice.

**Trust.** The trust parameter determines how much the algorithm trusts the ML advice. More formally, let \( \lambda \in (0, 1] \) be a trust parameter that indicates the level of trust that we place on the advice. In our algorithms, setting \( \lambda \to 0 \) represents full trust in ML advice, and \( \lambda \to 1 \) indicates no trust at all, i.e., making worst-case decisions similar to the classic competitive framework. Any value in between indicates partial trust in ML advice.

**Robustness and Consistency.** The performance of an algorithm in this framework is captured using two metrics that reflect two extreme cases when the advice is inaccurate and when the advice is fully accurate. Specifically, suppose that \( \mathcal{A}_\lambda \) is an online algorithm that leverages ML advice in decision making with the trust parameter \( \lambda \). Let \( \epsilon \) be the error of the ML advice, which is the absolute difference between the advice and actual outcome. Denote \( ALG(\epsilon, \lambda) \) as the cost of \( \mathcal{A}_\lambda \) given \( \lambda \) as the trust parameter and \( \epsilon \) as the error of the ML advice, and \( OPT \) as the offline optimum, respectively.

**Definition 1.** (Robustness) \( \mathcal{A}_\lambda \) is \( \alpha \)-robust if \( ALG(\epsilon, \lambda) \leq \alpha \cdot OPT \) for all \( \epsilon \) and feasible instances to the problem.

**Definition 2.** (Consistency) \( \mathcal{A}_\lambda \) is \( \gamma \)-consistent if \( ALG(0, \lambda) \leq \gamma \cdot OPT \) when the ML advice is accurate (\( \epsilon = 0 \)) and for all feasible instances to the problem.

Note that \( \sigma \) and \( \gamma \) could be functions of the problem parameters as well as \( \lambda \) and \( \epsilon \). Intuitively, robustness measures how well the algorithm does in the worst-case of poor advice, and consistency measures how well the algorithm does with perfect advice.

**Pareto Optimality.** In the traditional framework with competitive ratio as the performance metric, the notion of optimality refers to an online algorithm that achieves the best possible competitive ratio. In the new framework with ML advice, the performance of algorithms is measured by two criteria: robustness and consistency. The superiority of an algorithm in a bi-criteria setting against an alternative can be measured using the notion of dominance and Pareto optimality.

**Definition 3.** (Dominance) For comparing two online algorithms \( A \) and \( B \), we say that \( A \) dominates \( B \) if it is better in both criteria, i.e., \( \alpha_A \leq \alpha_B \) and \( \gamma_A \leq \gamma_B \).

With the trust parameter, we develop a class of online algorithms in which each instance of the algorithm refers to a specific value of trust parameter. Hence, for analyzing the dominance of the algorithms, our goal is to investigate the Pareto frontier properties of a class of algorithms.

**Definition 4.** (Pareto Optimality) Let \( \mathcal{A} = \{ A_\lambda, \lambda \in (0, 1] \} \) be the class of online algorithms with trust parameter \( \lambda \). \( \mathcal{A} \) is Pareto-optimal if for any other online algorithm \( B \), there exists \( A_\lambda \in \mathcal{A} \) such that \( A_\lambda \) dominates \( B \).

In Section 4, we develop deterministic and randomized algorithms using the above framework and analyze their robustness, consistency, and Pareto-optimality in Section 5. Our proposed algorithms are built on top of existing fully online algorithms that do not use ML advice for decision making. We briefly introduce these algorithms in the following subsection.

3.2 Existing Online Algorithms without ML Advice

The idea of prior online algorithms (OnEng) [34] lies in constructing a break-even point that balances between the cost of using generators and the peak charge of using the grid. Specifically, break-even point \( \sigma \) is

\[
\sigma = \frac{1}{p_m} \sum_{t \in T} (p_g - p(t))e(t) \quad (1)
\]

The parameter \( \sigma \) plays a critical role in algorithm design. For an optimal offline algorithm, we have \( \sigma'(t) = e(t), \forall t \in T \), when \( \sigma > 1 \); and \( \sigma'(t) = 0, \forall t \in T \), otherwise. The optimal output of the local generator is then \( u'(t) = e(t) - \sigma'(t) \).

The value of \( \sigma \) can be calculated easily in an offline manner. However, with unknown price and demand values, this value cannot be fully computed online. The high-level idea of OnEng is make a decision based on a partially-calculated value of \( \sigma \) over the current and past slots. Specifically, OnEng keeps using the local generator initially and switches to the grid at the first time \( t \) such that \( \sum_{\tau=1}^{t}(p_g - p(t))e(t) \geq p_m \). The competitive ratio of OnEng is \( 2 - \beta \). The proof ideas are similar to the ski-rental problem and they show that the break-even point is the best balance between being aggressive (paying the one-time premium peak cost) and being conservative (using the local generator). Finally, the competitive ratio has been improved to \( e/(e - 1 + \beta) \leq 1.58 \), by developing a randomized algorithm (rOnEng), in which the algorithm starts purchasing grid electricity when \( \sum_{\tau} (p_g - p(\tau)) \geq s \cdot p_m \), where \( s \) is chosen randomly according to the following distribution

\[
f^*(s) = \begin{cases} 
\frac{e^s}{e - 1 + \beta}, & \text{when } s \in [0, 1]; \\
\frac{\beta}{e - 1 + \beta}, & \text{when } s = \infty; \\
0, & \text{otherwise}.
\end{cases}
\]

4 ALGORITHM DESIGN

In this section, we first introduce how the ML advice could be constructed for the PAES problem, then present a Pareto-optimal deterministic algorithm, and finally a randomized algorithm that dominates our proposed Pareto-optimal deterministic algorithm. We also highlight our technical results. The detailed derivation of the theoretical results are given in Section 5.
4.1 ML advice for the PAES problem

First, we introduce the ML advice. Assume that there is a learning model that predicts the future values of external grid prices, \( \hat{p}(t) \), and energy demand, \( \hat{e}(t) \). The key idea in our online algorithm design lies in constructing a break-even point using these two values so as to balance between the cost of using generators and the peak charge of using the grid. Given these two values, let \( \hat{\sigma} \) be the predicted break-even point using the learning model as

\[
\hat{\sigma} \triangleq \frac{1}{p_m} \sum_{t \in T} (p_g - \hat{p}(t))\hat{e}(t).
\]

Note that it is even possible that the ML model directly predicts the value of \( \hat{\sigma} \) based on historical break-even points. Hence, predicting individual values of \( \hat{p}(t) \) and \( \hat{e}(t) \) for the cycle is not needed.

4.2 A Simple Consistent and Non-Robust Algorithm

We first show that there exists an algorithm \( \text{Eng-dd} \) for PAES that naively uses the predicted break-even point and is 1-consistent, i.e., its competitive ratio is 1 when the advice is accurate. However, it is straightforward to show that this algorithm is not robust since the competitive ratio can be arbitrarily large in the case of incorrect predictions. We empirically compare the result of our robust and consistent algorithms with this simple algorithm in Section 6.

Algorithm 1 \( \text{Eng-dd} \)

\[
\begin{align*}
\text{if } \hat{\sigma} > 1 \text{ then} & \\
& \text{use the local generator entirely} \\
\text{else} & \\
& \text{use the grid entirely} \\
\end{align*}
\]

4.3 OnMLEng: A Deterministic Robust and Consistent Algorithm

We propose an online algorithm OnMLEng with ML advice that uses the trust parameter \( \lambda \in (0, 1) \) to determine the level of trust in advice as introduced in Section 3. OnMLEng makes decisions based on the values of \( \hat{\sigma} \) and \( \lambda \) as summarized in Algorithm 2. Note that in OnMLEng, \( \lambda \to 0 \) (full trust) is equivalent to running Eng-dd.

Algorithm 2 OnMLEng

\[
\begin{align*}
\text{if } \hat{\sigma} > 1 \text{ then} & \\
& s \leftarrow \lambda \\
\text{else} & \\
& s \leftarrow 1/\lambda \\
\end{align*}
\]

Use local generator first and commit to switching to the grid starting at the first time \( \tau \) where

\[
\sum_{t=1}^\tau (p_g - p(t))e(t) \geq s \cdot p_m.
\]

Theorem 1. The OnMLEng algorithm achieves the competitive ratio of \( 1 + (1 - \beta)/\lambda \), where \( \lambda \in (0, 1) \).

Algorithm 3 rOnMLEng

\[
\begin{align*}
\text{Denote } & \Phi_1 = \frac{1}{e^{x - 1 + \lambda} + \beta} \text{ and } \Phi_2 = \frac{1}{e^{\lambda} + \beta} \\
\text{if } \hat{\sigma} > 1 \text{ then} & \\
& \Phi_1 e^x, \quad s \in [0, \lambda]; \\
& \Phi_2 \hat{\sigma} \beta \delta(0), \quad s = \infty; \\
& 0, \quad \text{otherwise.} \\
\text{else} & \\
& \Phi_2 e^x, \quad s \in [0, \frac{1}{\lambda}]; \\
& \Phi_2 \hat{\sigma} \beta \delta(0), \quad s = \infty; \\
& 0, \quad \text{otherwise.} \\
\end{align*}
\]

end if

Pick a value \( s \) randomly according to probability distribution \( f_1^*(s) \) or \( f_2^*(s) \), and switch to grid electricity starting at the first time \( \tau \) where

\[
\sum_{t=1}^\tau (p_g - p(t))e(t) \geq s \cdot p_m.
\]

Corollary 1. OnMLEng is \((1 + (1 - \beta)/\lambda)\)-robust.

Corollary 2. OnMLEng is \((1 + \lambda (1 - \beta))\)-consistent.

Intuitively, \( s \in (0, \infty) \) is a function of \( \hat{\sigma} \) and \( \lambda \) that determines when OnMLEng switches to the grid. Setting \( \lambda = 1 \) will fix \( s = 1 \), which recovers the robustness competitive ratio of \( 2 - \beta \) from the optimal online algorithm OnEng [34]. This implies that with bad advice it suffices to completely distrust the advice to be robust against worst-case performance. On the other hand, setting \( \lambda \to 0 \) will result in \( s \to 0 \) or \( s \to \infty \), which means immediately switching to the grid or entirely staying with the local generator respectively. This results in a consistency of 1. Tuning the value of \( \lambda \) effectively adjusts the level of trust in advice by determining \( s \).

Next, in Theorem 2, we show that OnMLEng represents a family of Pareto-optimal algorithms specified by the trust parameter \( \lambda \), based on Definition 4.

Theorem 2. OnMLEng defines the Pareto frontier of robustness and consistency for the PAES problem, and is Pareto-optimal for all deterministic algorithms that solve PAES.

The above result shows that OnMLEng defines the Pareto frontier. In other words, there is no other family of deterministic algorithms that can achieve a better consistency (resp., robustness) than OnMLEng without sacrificing the robustness (resp., consistency).

Furthermore, we show that for any deterministic algorithm \( \mathcal{A} \) that solves PAES, it can be expressed by a deterministic algorithm with a switching parameter, i.e., OnMLEng is Pareto-optimal for any deterministic algorithm \( \mathcal{A} \).

4.4 rOnMLEng: A Randomized Robust and Consistent Algorithm

In randomized algorithms, the decision making is based on random variable draws from a proper probability distribution function. We develop a randomized algorithm, rOnMLEng, as summarized in Algorithm 3. In rOnMLEng we modify the probability distribution function of OnMLEng [34] based on \( \lambda \) and \( \hat{\sigma} \) as in Eq. (3).
Theorem 3. \( \text{rOnMLEng} \) achieves a competitive ratio of
\[
\max \{1+\Phi_1(1-\beta), 1+\Phi_2(1-\beta)\}\{(e^{1/\lambda} - 1 - 1/\lambda)/(1/\lambda - 1) + 1/\lambda^2\}\},
\]
where \( \lambda \in (0, 1] \), \( \Phi_1 = \frac{1}{e^{\frac{1}{1+\lambda^2\beta}} - 1 + \lambda^2\beta} \) and \( \Phi_2 = \frac{1}{e^{\frac{1}{1+\lambda^2\beta}} - 1 + \lambda^2\beta} \).

Corollary 3. \( \text{rOnMLEng} \) is \( \max \{1+\Phi_1(1-\beta), 1+\Phi_2(1-\beta)\}\{(e^{1/\lambda} - 1 - 1/\lambda)/(1/\lambda - 1) + 1/\lambda^2\}\)-robust.

Corollary 4. \( \text{rOnMLEng} \) is \( \max \{1+\Phi_1\lambda^2(1-\beta), 1+\Phi_2(1-\beta)\}\)-consistent.

Corollary 5. The consistency and robustness bounds of \( \text{rOnMLEng} \) are strictly better than those of \( \text{OnMLEng} \).

The probability distributions \( f_1^*(s) \), \( f_2^*(s) \) are designed to satisfy two critical conditions. First, setting \( \lambda = 1 \) retains the original distribution function of the optimal randomized online algorithm \( \text{rOnEng} \), and therefore would recover its competitive ratio of \( e/(e - 1 + \beta) \). Second, setting \( \lambda \to 0 \) guarantees picking \( s = 0 \) or \( s = \infty \) depending on the advice driven break-even point \( \hat{\sigma} \). This will result in a competitive ratio of 1 for consistency, meaning matching optimal performance once the ML advice is accurate. Note that this follows the same selection of \( s \) in \( \text{OnMLEng} \) when \( \lambda \to 0 \).

Corollary 5 shows that the proposed randomized algorithm \( \text{rOnMLEng} \) dominates \( \text{OnMLEng} \), the Pareto optimal deterministic algorithm. Lastly, in Appendix B we show that the randomized algorithm that naïvely modifies the distribution function of \( \text{OnEng} \) (2) based on the guidelines in deterministic algorithm fails to achieve satisfactory robustness and consistency at the same time.

4.5 OnMLEng-dyn and rOnMLEng-dyn: Dynamic Break-even Point Algorithms

\( \text{OnMLEng} \) and \( \text{rOnMLEng} \) utilize a static predicted break-even point \( \hat{\sigma} \) that persists over the entire billing cycle, but our results can also be extended to utilizing a set of dynamic break-even points \( \hat{\sigma} = \{\hat{\sigma}_1, \hat{\sigma}_2, ..., \hat{\sigma}_T\} \). Having a dynamic break-even point captures a broad range of algorithms and allows a rich design space within \( \text{OnMLEng} \) and \( \text{rOnMLEng} \). For example, predictions can be adjusted and possibly improved according to observed values over time. Algorithms that use a sliding window of predictions also fit within this framework, since the break-even point is dynamically calculated according to the available predictions.

Define \( \text{OnMLEng-dyn} \) as the version of \( \text{OnMLEng} \) where the set of predictions at each time step may change over time. In other words, let \( \hat{\epsilon}_t = [\hat{\epsilon}_t(t)]_{t \in [T]}, \hat{p}_t = [\hat{p}_t(t)]_{t \in [T]} \), be the set of predictions for demand and grid price at time \( t \in [1, T] \). Then the advice is dynamically provided as
\[
\hat{\sigma}_t = \frac{1}{p_m} \left[ \sum_{t \in [T]} (p_g - \hat{p}_t(t))\hat{\epsilon}_t(t) \right],
\]
and the decision to switch from the local generator to the grid is made according to \( \hat{\sigma}_t \) and \( \lambda \). Similarly, define \( \text{rOnMLEng-dyn} \) as the version of \( \text{rOnMLEng} \) using a set of break-even points \( \hat{\sigma} \), with all else remaining the same.

Theorem 4. The robustness and consistency of \( \text{OnMLEng-dyn} \) and \( \text{rOnMLEng-dyn} \) are equivalent to those of \( \text{OnMLEng} \) and \( \text{rOnMLEng} \), respectively.

---

**Algorithm 4 OnMLEng-dyn**

Use local generator first and switch to the grid starting at the first time \( t \) where
\[
\sum_{t=1}^{T} (p_g - p(t))e(t) \geq s_T \cdot p_m,
\]
and \( s_T \) is defined by
\[
s_T = \begin{cases} 
\lambda, & \hat{\sigma}_1 > 1; \\
\frac{1}{1/\lambda}, & \hat{\sigma}_1 \leq 1.
\end{cases}
\]

**Algorithm 5 rOnMLEng-dyn**

Denote \( \Phi_1 = \frac{1}{e^{\frac{1}{1+\lambda^2\beta}} - 1 + \lambda^2\beta} \) and \( \Phi_2 = \frac{1}{e^{\frac{1}{1+\lambda^2\beta}} - 1 + \lambda^2\beta} \).

\[
\text{if } \hat{\sigma}_t > 1 \text{ then } \begin{cases} 
\Phi_1 e^{\hat{\sigma}_t}, & s \in [0, \lambda]; \\
\Phi_1 \lambda^2 \beta \delta(0), & s = \infty; \\
0, & \text{otherwise}.
\end{cases}
\]

\[
\text{else } \begin{cases} 
\Phi_2 e^{\hat{\sigma}_t}, & s \in [0, \frac{1}{\lambda}]; \\
\Phi_2 \lambda^2 \beta \delta(0), & s = \infty; \\
0, & \text{otherwise}.
\end{cases}
\]

end if

Pick a value \( s_1 \) randomly according to probability distribution \( f_1^*(s) \) and \( s_2 \) likewise from \( f_2^*(s) \).

Switch to grid electricity starting at the first time \( t \) where
\[
\sum_{t=1}^{T} (p_g - p(t))e(t) \geq s_T \cdot p_m.
\]
and \( s_T \) is defined by
\[
s_T = \begin{cases} 
s_1, & \hat{\sigma}_1 > 1; \\
s_2, & \hat{\sigma}_1 \leq 1.
\end{cases}
\]

While the theoretical bounds of dynamic break-even algorithms are the same for robustness, consistency, and Pareto optimality, these algorithms are of practical importance because they can capture scenarios such as a sliding window of available predictions or improved prediction quality over time. We empirically evaluate the performance of one such algorithm in Section 6.

4.6 Extending Algorithms for Energy Problem to the General Case

The proposed algorithms for PAES in the paper are analyzed for a basic version in which the demand takes binary values of 0 or 1. Also, the corresponding competitive analyses are dedicated to the basic setting. However, the results can be straightforwardly extended to the general problem of non-negative integer demand. This is done by dividing the integer demand \( e(t) \) into multiple sub-problems with binary demand. At a given layer \( i \), the layered demand at time \( t \) is 1 if \( e(t) \leq i \) and 0 otherwise. Then the result in [34, Theorem 3] can be applied. By using the layered sub-problems strategy, the competitive ratio of an algorithm which solves the sub-problem
with binary demand is an upper bound to the competitive ratio of an algorithm which solves the general integer demand problem. Similarly, the robustness and consistency of the binary demand setting provide an upper bound to the robustness and consistency of the general setting.

5 PROOFS OF MAIN RESULTS
In this section, we provide the main proofs for the algorithms. The proof for the competitive ratio of the randomized algorithm is given in Appendix A.

5.1 Proof of Theorem 1
We analyze the competitiveness of OnMLEng. Given the structure of Algorithm 2, we can parameterize any online algorithm by parameter $s$. Let $\mathcal{A}_s$ be an online algorithm with a specific parameter $s$, e.g., OnMLEng is in this category with the value of $s$ as in Algorithm 2. Let $h(\mathcal{A}_s, \sigma)$ be the ratio between the cost of algorithm $\mathcal{A}_s$ and that of an optimal offline algorithm given $\sigma$. The following proposition characterizes the closed-form value of $h(\mathcal{A}_s, \sigma)$, and facilitates the analysis of the proposed algorithm.

**Proposition 1.** [34] For any online algorithm $\mathcal{A}_s$, we have

\[
\begin{align*}
    h(\mathcal{A}_s, \sigma) &= \begin{cases} 
        1, & \text{if } s > \sigma; \\
        1 + \frac{1-\sigma \delta}{\delta} (1 - \beta), & \text{otherwise}.
    \end{cases}
\end{align*}
\]

When $\sigma > 1$, $h(\mathcal{A}_s, \sigma) = \begin{cases} 
    1 + \frac{1 + (1 - \beta)}{(1 - \beta) \beta_i}, & \text{if } s > \sigma; \\
    1 + \frac{1 + (1 - \beta) \beta_i}{\beta_i}, & \text{otherwise}.
\end{cases}$

We proceed to prove the robustness and consistency results. We first consider the robustness. The worst-case cost ratio for a general deterministic algorithm $\mathcal{A}_s$ with parameter $s$ is when $s = \lambda$, where the online algorithm pays for the peak charge premium but has no net demand to serve anymore. From Proposition 1, this worst case cost ratio $\max_\sigma h(\mathcal{A}_s, \sigma)$ is

\[
\max_\sigma h(\mathcal{A}_s, \sigma) = \begin{cases} 
    1 + \frac{1 + (1 - \beta)}{(1 - \beta) \beta_i}, & \text{if } s \leq 1; \\
    1 + \frac{1 + (1 - \beta) \beta_i}{\beta_i}, & \text{otherwise}.
\end{cases}
\]

We compute the competitive ratio of OnMLEng under two cases:

(i) $\dot{\sigma} > 1$: According to OnMLEng, $s = \lambda < 1$. From (5), we have $\text{CR}(\mathcal{A}_\lambda) = 1 + (1 - \beta)/\lambda$.

(ii) $\dot{\sigma} \leq 1$: According to OnMLEng, $s = 1/\lambda > 1$. From (5), we have $\text{CR}(\mathcal{A}_1/\lambda) = 1 + (1 - \beta)/\lambda$.

This means that OnMLEng is $(1 + (1 - \beta)/\lambda)$-robust. Note that setting $\lambda = 1$ recovers the competitive ratio of the optimal online algorithm.

Next, we consider the consistency. For consistency guarantees, we compute the competitive ratio assuming the predictions are correct. There are two cases to consider here:

(i) $\dot{\sigma} = \sigma > 1$, i.e., $s = \lambda$. From Proposition 1, when $\sigma > 1 \geq s = \lambda$, we have $\text{CR}(\mathcal{A}_1/\lambda) = 1 + \frac{(1 - \beta)}{(1 - \beta) \beta_i} \leq 1 + \lambda (1 - \beta)$.

(ii) $\dot{\sigma} = \sigma \leq 1$, i.e., $s = 1/\lambda$. From Proposition 1, when $\sigma \leq 1 \leq s = 1/\lambda$, the worst case occurs when $s = 1/\lambda = \sigma$. Then $\text{CR}(\mathcal{A}_1/\lambda) = 1 + \lambda (1 - \beta)$.

This means that OnMLEng is $(1 + \lambda(1 - \beta))$-consistent. Note that setting $\lambda \to 0$ results in a competitive ratio of 1, which means optimal performance with accurate predictions.

5.2 Proof of Theorem 2
First, we establish that OnMLEng is Pareto optimal for all deterministic algorithms with a switching parameter. From Theorem 1, OnMLEng is $(1 + \lambda(1 - \beta))$-consistent and $(1 + \frac{1}{2}(1 - \beta))$-robust. Note the consistency and robustness bounds as $\gamma_A = 1 + \lambda(1 - \beta)$ and $\alpha_A = 1 + \frac{1}{2}(1 - \beta)$.

Consider an arbitrary algorithm $A’$ that takes prediction-based advice with a consistency bound $\gamma_{A’}$ and robustness bound $\alpha_{A’}$. $A’$ switches at either $i \cdot p_m$ or $j \cdot p_m$ based on the advice. Without loss of generality, we assume $i \leq j$.

**Lemma 1.** $A’$ must be at least $\gamma_{A’} \geq \frac{1}{2}(1 - \beta)$-consistent and $\alpha_{A’} \geq \frac{1}{2}(1 - \beta)$-robust.

**Proof.** We consider a couple of cases for the true break-even point $\sigma$ utilizing Proposition 1: First, when $\sigma \leq 1$, $A’$ will either select $s = i$ or $s = j$ for competitive ratios of $1 + \frac{1}{2}(1 - \beta)$ or $1 + \frac{1}{2}(1 - \beta)$, respectively. We now consider the corresponding consistency and robustness:

(i) For consistency, $A’$ has perfect predictions and knows exactly that $\sigma \leq 1$. As a result, $A’$ will rationally pick $s = j$, since $i \leq j$ implies that $\frac{1}{2}(1 - \beta) \leq \frac{1}{2}(1 - \beta)$. Then $\gamma_{A’} \geq \frac{1}{2}(1 - \beta)$.

(ii) For robustness, $A’$ does not have perfect predictions and cannot have full certainty that $\sigma \leq 1$. Then $A’$ could rationally pick $s = i$ or $s = j$, and in the worst case $s = i$ will be chosen. Then $\alpha_{A’} \geq \frac{1}{2}(1 - \beta)$. \hfill $\Box$

Assume that $A’$ has a lower consistency bound than OnMLEng, i.e., $\gamma_{A’} \leq 1 + \lambda(1 - \beta)$. It follows that $1 + \frac{i(1 - \beta)}{\beta_i} \leq 1 + \lambda(1 - \beta)$, and subsequently $i \leq \lambda$. Applying this to the robustness bound for $\alpha_{A’}$ yields $\alpha_{A’} \geq 1 + \frac{1}{2}(1 - \beta) \geq 1 + \frac{1}{2}(1 - \beta) = \alpha_A$.

This concludes the proof that OnMLEng is Pareto optimal for all deterministic algorithms with a switching parameter, since $\gamma_{A’} \leq \gamma_A$ guarantees that $\alpha_{A’} \geq \alpha_A$ for any algorithm $A’$ with switching parameters $i, j$.

**Lemma 2.** Any deterministic algorithm for PAES can be expressed by a deterministic algorithm with a switching parameter.

The main idea for proving this lemma is that time slots assigned to the local generator and the grid can be reordered such that the assignment can be determined by a single parameter. The full details of the proof are given in Appendix C. Combining Lemma 1 and Lemma 2 concludes the proof of Theorem 2.

5.3 Proof of Theorem 4
The key observation is that Proposition 1 still holds with a dynamic break-even point, where we can characterize the competitive ratio of any online algorithm $\mathcal{A}_s$ with parameter $s$. Although the value of $s$ will change over time in OnMLEng-dyn with dynamic advice $\dot{s}_t$, the possible values of $s$ remain the same as the possible values in OnMLEng. The possible competitive ratios must be the same, and subsequently the robustness and consistency are the same. Similarly
Theorem 2. Thus the Pareto optimality of general demand setting.

5.4 Extending Results to the General Demand Setting
Consider an instance of PAES with integer demand. We can construct a binary demand instance PAES-a at the k-th layer by denoting \( e^k(t) = 1_{\{v(t) \geq k\}} \). The full details of decomposing PAES into PAES-a are inspired by [34], so we outline the necessary adaptions for robustness and consistency analysis.

Denote \( \hat{v}^k(t) \) and \( \hat{u}^k(t) \) the energy usage from the grid and local generator respectively from the k-th layer of binary demand. Note that \( \max_t \sum_k \hat{v}^k(t) = \sum_k \max_t \hat{v}^k(t) \), i.e. the overall peak grid utilization is the sum of the layered peak utilization. Similarly, \( u(t) = \sum_k \hat{u}^k(t) \), \( v(t) = \sum_k \hat{v}^k(t) \), i.e. the overall grid and generator usage is the sum of the layered grid and generator usage. Then we have

\[
\text{cost}(\text{PAES} - \text{ALG}) = \sum_k \text{cost}(\text{PAES-a} - \text{ALG})
\]

i.e. the cost of PAES is equal to the sum of the costs over the binary demand problems PAES-a.

5.5 Extending Consistency and Robustness Results
Let an algorithm which solves PAES-a be \( \alpha \)-robust and \( \gamma \)-consistent in the binary demand setting. Then we demonstrate that extending to the integer demand setting PAES is also \( \alpha \) robust and \( \gamma \) consistent. Consider the consistency scenario, where the integer demand predictions \( \hat{v}(t) \) are correct. Then each layer prediction \( \hat{v}^k(t) \) would also be correct. We can then use the \( \gamma \) consistency bound. Consider a binary demand layer \( k \):

\[
\text{cost}(\text{PAES-a} - \text{ALG}) \leq \gamma \text{cost}(\text{PAES-a} - \text{OPT}), \forall k
\]

Then summing over \( k \) gives:

\[
\text{cost}(\text{PAES} - \text{ALG}) \leq \gamma \text{cost}(\text{PAES} - \text{OPT})
\]

Now consider the robustness scenario, where the overall demand prediction \( \hat{v}(t) \) is not necessarily accurate. Then each consider a binary demand layer \( k \) will be \( \alpha \) robust:

\[
\text{cost}(\text{PAES-a} - \text{ALG}) \leq \alpha \text{cost}(\text{PAES-a} - \text{OPT}), \forall k
\]

Then summing over \( k \) gives:

\[
\text{cost}(\text{PAES} - \text{ALG}) \leq \alpha \text{cost}(\text{PAES} - \text{OPT})
\]

We can just substitute the respective consistency and robustness bounds of OnMLEng and rOnMLEng for \( \gamma \) and \( \alpha \). Thus the upper bounds on OnMLEng and rOnMLEng extend to the general demand setting.

5.6 Extending the Pareto Optimality of OnMLEng
Observe that PAES-a is a special case of PAES. Then it is impossible for an algorithm which solves PAES to dominate OnMLEng in the integer demand setting. If such an algorithm existed, then it would dominate OnMLEng in the binary demand setting, which contradicts Theorem 2. Thus the Pareto optimality of OnMLEng extends to the general demand setting.

6 EXPERIMENTS
We use real-world traces to experimentally evaluate the performance of proposed learning-assisted algorithms as compared to the pure online algorithms and the offline optimum. Our proposed algorithms characterize a class of algorithms that are determined by the choice of trust parameter. Our experiments consider such algorithms in both the worst-case performance and practical average-case performance scenarios. The results answer these questions:

(1) How does the OnMLEng algorithm compare to the pure online algorithm? Our results show that OnMLEng consistently achieves better average performance than the pure online algorithm, sometimes even achieving near-optimality.

(2) What is the effect of varying prediction quality via renewable penetration? Lower-quality predictions noticeably degrade the worst-case performance of OnMLEng instances that are too optimistic about advice, while the performance of OnMLEng instances with more cautious trust parameter selection is robust to poorer prediction quality.

(3) How do problem parameters such as peak price and grid capacity constraints affect the performance? The normalized cost of the best performing OnMLEng algorithm remains extremely close to optimal under four different varying parameters.

6.1 Data Traces and Comparison Algorithms
6.1.1 Data Center Energy Demands. For representing the energy demands of data centers, we use a dataset including the server load information for 300+ Akamai data centers across the United States, collected every 5 minutes [24]. Since some data centers are co-located with on-site renewable sources, we use wind data traces from [16] and inject renewable sources with 40% penetration in our experiments, unless the penetration level is otherwise stated. Two sample one week trajectories of energy demand for different locations in the United States are depicted in Figure 2. While we see a roughly diurnal pattern for the New York City energy demand, the pattern is less visible for Rochester. For both cities, the high unpredictability of renewable generation leads to comparable unpredictability in the net energy demand, regardless of diurnal patterns in energy demand. These observations show the importance of ML advice, as well as the possibility of tuning the level of trust in a principled manner.

6.1.2 Energy Pricing Data. We use the 2018 spot energy prices from the New York Electricity Market (NYISO). The value of spot prices changes in real-time over intervals of 5 minutes. As an example, the spot prices in April 2018 vary between $13.69/MWh and $64.62/MWh. The value of \( p_d \) is set to be roughly \( 100 \times \max_{t \in T} p(t) \), which is based on common practice by U.S. utilities such as PG&E and Duke Energy. The cost of local generation is set to \( p_g = \max_{t \in T} p(t) \). Finally, the capacity of the local generator is set to be roughly 60% of the energy demand.

6.1.3 Comparison Algorithms. Table 2 summarizes the acronyms for different algorithms in our experiments. Here, we use three approaches to determine the trust parameter: first, offline optimal selection of the trust parameter – this approach searches over all possible values of \( \lambda \) in a brute force manner as input to OnMLEng.
and selects the best performing choice of $\lambda$. Although selecting the best choice of $\lambda$ is not possible in online settings, the optimal hybrid algorithm serves to demonstrate the full potential of algorithms with ML advice. In experiments, the offline optimal algorithm OnMLEng-opt is used for PAES. Second, as a practical online selection, we choose the trust parameter based on the historical optimal value, that is the best $\lambda$ for the previous instance of the problem. The algorithm OnMLEng-hist is used for this scenario. To demonstrate time-varying predictions, algorithm OnMLEng-hist-dyn aligns the predictions used in OnMLEng-hist according to observed values in an online manner. Third, we use the adversarial Lipschitz algorithm in a full-information environment (ALF) [23] to learn the choice for $\lambda$ based on a history of previous instances. For this online learning setting, we compare against OnMLEng-static, the best static choice of $\lambda$ over the full history of instances.

In experiments, we report the normalized cost of different algorithms. The normalized cost is the ratio between the cost of the algorithm and the offline optimal cost (i.e., ENG-OPT for PAES). The normalized cost is always greater than or equal to 1. The lower the cost ratio of an algorithm, the better the performance. Finally, to show how online algorithms with ML advice achieve the best of both worlds, we compare their normalized cost to pure online algorithms (OnEng [34]) and naive data-driven algorithms that fully trust the advice (Eng-dd).

6.2 Large-scale Trace-Driven Evaluation

6.2.1 Analysis at a Single Renewable Penetration Level. We first evaluate the performance of the proposed algorithms over a large variety of locations and trials, with emphasis on demonstrating how the proposed algorithms are able to achieve the best of both worlds. In Figure 3, we report results for PAES over 338 locations and 30 trials at 30%, 40%, and 50% renewable penetration. To begin, we focus on the cumulative probability distribution of normalized cost at a 30% renewable penetration. Specifically, we observe that OnEng is strictly upper bounded by the theoretically guaranteed bound of approximately 1.85, but the majority (over 80%) of locations have normalized cost of greater than 1.6. The Eng-dd algorithm has comparatively better normalized cost for the majority of locations, but has a heavy tail. OnMLEng-opt and OnMLEng-hist clearly outperform OnEng and Eng-dd since they leverage advice for better decision making. Last but not least, the performance of OnMLEng and its variants largely depend on the level of uncertainty in energy demand.

6.2.2 Analysis at Multiple Renewable Penetration Levels. We now consider the effect of all three different renewable penetration levels in Figure 3. Overall, increasing the penetration degrades the accuracy of the predictions and subsequently drives the normalized cost higher. This is most prevalent in the worst case scenario, where the heavy tails beyond the theoretical guarantee increase from 20% to 40% for Eng-dd. On the other hand, the mean normalized cost is relatively robust degrading predictions via penetration level. For all algorithms, the mean normalized cost remains below the theoretical guarantee.

6.2.3 Evaluating a Dynamic Break-even Point Algorithm. A natural choice for a dynamic break-even point algorithm within OnMLEng-dyn is one that aligns the predictions with observed past and current values, i.e. once $p(t), e(t)$ are observed at time $t$ then the predictions for all current and future time slots $\tau \geq \tau'$ are set as
Fig. 3. Cumulative probability distribution of normalized cost of different algorithms at 30%, 40%, and 50% penetration levels. We consider some key observations from these plots. First, algorithms with ML advice almost strictly outperform OnEng in mean normalized cost, but careful selection of trust level is important as prediction quality decreases. Second, the worst-case performance in OnMLEng-hist is noticeably robust to degrading prediction quality when compared to Eng-dd, indicating that careful selection of trust level will restrict poor worst-case performance.

Fig. 4. Cumulative probability distribution of dynamic vs. static break-even advice algorithms at 50% penetration. The key observation is that OnMLEng-hist-dyn is slightly better than OnMLEng-hist in worst case scenarios, but equivalent in the practical average case scenario.
\( \hat{p}_{t \leq \tau}(t) = p(t), \hat{\xi}_{t \leq \tau}(t) = e(t) \). This type of algorithm is a natural middle ground between OnEng and OnMLEng, since the predicted break-even points are gradually aligning with observed values. We consider a variant of this algorithm OnMLEng-hist, which uses the best historical trust parameter, and compare it against OnMLEng-hist-dyn. From Figure 4, we see that OnMLEng-hist-dyn is slightly better than OnMLEng-hist in worst case and 99 percentile scenarios, but functionally equivalent in the average case scenario. This correlates closely with Theorem 4, as the two algorithms have equivalent theoretical robustness bounds.

### 6.3 Evaluation Results for PAES

In this section, we investigate the impact of different parameters on the performance of the proposed algorithms.

#### 6.3.1 The Impact of Trust Parameter

Introducing the trust parameter in the algorithm design allows effective usage of predictions in algorithmic actions. Specifically, setting \( \lambda \) close to 0 represents more trust in predictions, while \( \lambda \) close to 1 represents almost no trust in predictions. To scrutinize the impact of \( \lambda \) on the performance of OnMLEng, in Figure 5(a) we vary the value of \( \lambda \) from 0 to 1. We report the average normalized cost over several locations and trials, with ML advice in three regimes: (i) accurate denotes perfect ML advice, (ii) high error denotes poor ML advice, and (iii) previous with the values of the previous run of the algorithm as the ML advice. These three regimes represent a broad range of ML advice and the goal is to investigate the impact of level of trust in different algorithms. The notable observations are summarized as follows: (1) With accurate ML advice, and \( \lambda \leq 0.1 \), OnMLEng achieves the optimal performance. (2) With high error in ML advice, unfavorable values of \( \lambda \) (high trust on prediction) lead to even worse performance than the pure online algorithms. (3) Favorable setting of \( \lambda \), e.g., \( \lambda \geq 0.4 \) for OnMLEng, achieves better performance even with high error in ML advice. This experiment signifies the importance of setting right values for the trust parameter in order to outperform purely online algorithms without advice.

#### 6.3.2 The Impact of Peak Price

The peak price \( p_m \) is an important parameter that impacts the break-even point. Case studies show that the peak charge varies substantially in different geographical locations, ranging from 20% to 80% of the total electricity bill [31]. In this experiment, we investigate the impact of this parameter on different algorithms. We scale the value of peak price from \( 1x \) to \( 20x \) of its original value and report the average normalized cost values of 30 runs in Figure 5(b). The result shows that the normalized cost of OnMLEng with trust \( \lambda = 0.5 \) is constantly better than OnEng. OnMLEng-hist is always very close to OnMLEng-opt and is substantially better than OnEng. Interestingly, the normalized costs of all algorithms are better in the extremes of low and high peak prices. This is reasonable since with low peak prices it is natural to use the grid. At high peak prices, the optimal decision is clearly to fully utilize the generators. So, despite the uncertainty of the input, decisions in these two extreme regimes are trivial.

#### 6.3.3 The Impact of Ramp Constraints

The algorithms proposed in this paper work for fast-response generators. In practice, there are several generators that are slow-response and cannot switch their output level quickly. The proposed algorithms are easily modified to incorporate ramp constraints. Specifically, let \( R \) be the ramp constraints, so that \( |u(t) - u(t - 1)| \leq R \), \( \forall t \in T \), i.e., the changes in generator output level should be always less than \( R \). We can easily modify OnMLEng and OnEng, as explained in [35, Section 4], to reflect the ramp constraint. The idea is to first run the algorithm without the ramp constraints, and then, project the obtained values to the feasible region to respect the ramp constraints. In Figure 5(e), we vary the ramp to capacity ratio from 10% to 100%, and report the average normalized cost of OnEng and OnMLEng. The result shows that OnMLEng always achieves better performance than OnEng. Although the normalized costs for OnEng and OnMLEng increase as we relax the ramp constraints, OnMLEng-hist and OnMLEng-opt those values are robust.

#### 6.3.4 The Impact of Local Generation Capacity

A drawback of pure online algorithms such as OnEng is that they are too conservative in decision making. An example of such performance degradation is once the capacity of the generator is above 60% of the total energy demand (see Figure 5 in [34]). By leveraging ML advice, we can effectively prevent this performance degradation. To show this, we investigate the cost saving of different algorithms as the capacity of generator changes. We define \( p = C_{max}^e e(t) \) as the ratio between the capacity of generator and the maximum energy demand, and change this value from 10% to 100%. Figure 5(d) shows the normalized cost of different algorithms. To better illustrate the benefit of algorithms with ML advice, in Figure 5(e) we report the cost reductions as compared to a baseline without local generation. With \( p \leq 30\% \), all algorithms perform more or less similarly. However, with \( p > 40\% \) the performance of OnEng and OnMLEng with \( \lambda = 0.5 \) degrades substantially, while the cost reduction of OnMLEng-opt and OnMLEng-hist increases. We consider this observation as another critical motivation to use online algorithms with ML advice for tackling online problems.

### 6.4 Parameter Selection with Online Learning

The previous sections largely consider the normalized cost of one billing period as the evaluation metric for different algorithms. Another perspective is the online learning scenario, where we evaluate the performance of algorithms over a sequence of billing period instances. This is a useful scenario since online learning is an alternative method of learning the trust parameter \( \lambda \) over multiple instances, beyond the basic method of looking at the previous instance alone as in OnMLEng-hist.

Specifically, we consider the problem of selecting \( \lambda \) over \( N \) rounds. For each round \( n \in [1, N] \), we receive a problem instance PAES\(_n\) and must choose a trust parameter \( \lambda_n \) to use this round. Let the cost in round \( n \) be cost(PAES\(_n\), \( \lambda_n \)). The best single offline optimal static trust parameter \( \lambda_{static} \) satisfies

\[
\lambda_{static} = \arg \min_{\lambda} \sum_{n \in [1:N]} \text{cost(PAES}_n, \lambda) .
\]

Then the goal of online learning is to choose \( \lambda_n \) each round such that the regret is minimized compared to using \( \lambda_{static} \), i.e.,
Regret(N) = \sum_{n \in [1,N]} \text{cost}(\text{PAES}_n, \lambda_n) - \sum_{n \in [1,N]} \text{cost}(\text{PAES}_n, \lambda_{\text{static}}).

In our experiments for online learning, we use the adversarial Lipschitz algorithm [23] in a full-information setting to choose \( \lambda \). ALF initially samples \( \lambda_n \) from a uniform distribution, but gradually updates the probability distribution upon observing the cost of each setting of \( \lambda \) on the instance PAES\(_n\). In other words, choices of \( \lambda \) with low cost will be more likely to be sampled in the next round.

We report the results of online learning over 372 billing period instances. Figure 6(a) compares the average cumulative cost of OnMLEng-ALF and OnMLEng-static, and Figure 6(b) plots the regret of OnMLEng-ALF. Initially in early rounds, there is a large gap between OnMLEng-ALF and OnMLEng-static because ALF is still sampling \( \lambda_n \) for each round. As the number of rounds increases, ALF begins to settle on a choice of \( \lambda \) and the gap with OnMLEng-static narrows, with the gap appearing to become constant in later rounds. This convergence to the static offline optimal is also reflected in the regret plot, which shows the growth of regret slowing drastically after 50 rounds, and growing only marginally near the end of the learning process.
The framework of ML advice for online algorithms is recently proposed and has been utilized for several online problems, e.g., online caching [22, 27, 29], bin packing [1], ski rental [3, 25, 28, 30] and job scheduling [25, 30]. However, this work is the first that uses this framework in the context of energy scheduling. Traditional approaches incorporating predictions often assume the prediction or prediction error follows a particular distribution or stochastic process, which limits the generality and practicality of their prediction framework. Other works [8, 9] use prediction windows of limited size that do not provide any information about events further into the future. These works apply predictions directly into the optimization for decision making. In this paper, predictions are used to generate advice for decision making. From our theoretical analysis and numerical evaluation, using advice is more powerful because only high-level structure such as break-even points is needed. This is in contrast to requiring detailed prediction of each time slot and modelling the error structure of each prediction.

PAES is an extended version of the ski-rental problem [6], in which a skier is going to ski for an unknown number of days. For each day, the skier can either rent skis at unit price or buy them for a higher price of \( h > 1 \) and ski for free from then on. The best known deterministic algorithm for ski-rental problem is the break-even algorithm: rent the first \( h - 1 \) days and buy on day \( h \). In PAES, there is a rent-vs-buy dilemma in usage-based vs. peak-based decision making and the online algorithms in literature follow the break-even structure [34]. However, there is an additional unique challenge dedicated to PAES, namely that the “buy” option is not fully free and has an additional time-varying unit price. In our algorithm design with ML advice, we assume that an ML model provides an estimate of the break-even point to the problem. We do not assume any modeling from ML and treat it as a black-box that provides input to our algorithms.

8 CONCLUDING REMARKS

This paper improves the performance of classic competitive algorithms with ML advice in a principled manner for the peak-aware energy scheduling problem. Different from prior literature on using prediction for online algorithms, our algorithms are empowered with a parameter that determines the level of trust in the ML advice. For all algorithms we characterized the competitive ratio as a function of the trust parameter and showed that our algorithms are provably the best possible algorithms in this framework since they are Pareto optimal. By extensive large-scale experiments we showed the improved performance of the proposed algorithms against pure online algorithms as well as data-driven algorithms that naively trust the advice, verifying that our algorithms achieve the best of both worlds. While we focused on an energy scheduling problem, the rent-vs-buy nature and the category of break-even point algorithms appear frequently in broad application domains such as server on/off scheduling, TCP acknowledgment, and renting cloud servers, and a promising future direction is to extend the break-even point algorithms for those problems. Another promising direction is to incorporate the energy storage systems into the peak-aware energy scheduling problem.

ACKNOWLEDGMENTS

We acknowledge the support from National Science Foundation under grant numbers CNS-1908298, CNS-1763617, CNS-1717588, CNS-1730128, CNS-1919752, CNS-1839287, CNS-2106299, NGSDI-2105494, CAREER-2045641, CPS-2136199, and a Graduate Research Fellowship. In addition, this research is supported by an IBM Academic Award, and the U.S. DOE Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009341.

REFERENCES


cost is given by \( \int h(s, \sigma) f_2^2(s) ds \)

\[
= \Phi_1 \int_0^\lambda \left[ 1 + \frac{s(1 - \beta)}{(\sigma - 1)\beta + 1} \right] e^\sigma ds + \Phi_1^3 \left[ 1 + \frac{(\sigma - 1)(1 - \beta)}{(\sigma - 1)\beta + 1} \right] \frac{1}{\lambda^2} \beta
\]

\[
= 1 + (1 - \beta)\Phi_1^3 \frac{1}{\lambda^2} + \Phi_1^3 \frac{(e^{\lambda} - 1 - \lambda)(\lambda - 1)}{(\sigma - 1)\beta + 1}
\]

\[
\leq 1 + (1 - \beta)\Phi_1^3 \frac{2}{\lambda^2} + \Phi_1^3 \lambda^2(1 - \beta),
\]

where (a) holds true since we have \( 0 \leq e^{\lambda} - 1 - \lambda \) for \( \lambda \in (0, 1) \) from the discussions in Case (i), and \( \lambda - 1 \leq 0 \).

We now consider the cases where \( \hat{\sigma} \leq 1 \).

(iv) \( \hat{\sigma} \leq 1, 1 \leq \lambda < \sigma \). Note this is a worst case failed prediction scenario. The expected cost is given by \( \int h(s, \sigma) f_2^2(s) ds \)

\[
= \Phi_2 \int_0^\lambda \left[ 1 + \frac{s(1 - \beta)}{(\sigma - 1)\beta + 1} \right] e^\sigma ds + \Phi_2^3 \left[ 1 + \frac{(\sigma - 1)(1 - \beta)}{(\sigma - 1)\beta + 1} \right] \frac{1}{\lambda^2} \beta
\]

\[
= 1 + \Phi_2^3 \frac{1}{\lambda^2} + \Phi_2^3 \frac{(e^\lambda - 1 - \frac{\lambda}{2})(\lambda - 1)}{(\sigma - 1)\beta + 1}
\]

\[
\leq 1 + \Phi_2^3 \frac{1}{\lambda^2} + \Phi_2^3 \frac{(e^\lambda - 1 - \frac{\lambda}{2})(\lambda - 1)}{(\sigma - 1)\beta + 1}
\]

where (c) holds since \( (\sigma - 1)\beta + 1 \geq 1 \) and \( (e^\lambda - 1 - \frac{\lambda}{2}) \) is positive as shown in case (i) that \( f(1/\lambda) \) is increasing for \( \lambda \in [0, 1] \) so \( f(1/\lambda) \geq f(1) > 0 \).

\[\text{ACM SIGENERGY Energy Informatics Review}\]

Volume 1 Issue 1, November 2021
(v) \( \bar{\sigma} \leq 1, 1 \leq \sigma < \frac{1}{\lambda} \). Note this is a worst case 3 failed prediction scenario. The expected cost is given by 
\[
\int_0^\infty h(s, \sigma) f_2(s) \, ds \n\]
\[
= \Phi_2 \int_0^\infty \left[ 1 + \frac{s(1-\beta)}{(s-1)\beta + 1} \right] e^s \, ds + \Phi_2 \int_\sigma^\infty e^s \, ds + \Phi_2(1) \frac{1}{\lambda^2} \beta 
\]
\[
= 1 + \Phi_2(1-\beta) \left[ \frac{e^{\bar{\sigma}} - 1}{\lambda} \right] \left[ \frac{1}{\lambda^2} \right] \beta 
\]
where (e) is true since \( \sigma < \frac{1}{\lambda} \). The final step is true since inequality (e) is equivalent to inequality (b) from case (iv), so the competitive ratio of case (v) reduces to the competitive ratio of case (iv). So the same robustness bound from (iv) will also dominate OnMLEEng.

(vi) \( \bar{\sigma} \leq 1, \sigma \leq 1 < \frac{1}{\lambda} \). This is a correct prediction scenario. The expected cost is given by 
\[
\int_0^\infty h(s, \sigma) f_2(s) \, ds \n\]
\[
= \Phi_2 \int_0^\infty \left[ 1 + \frac{1-\sigma + s}{\sigma} (1-\beta) \right] e^s \, ds + \Phi_2 \int_\sigma^\infty e^s \, ds + \Phi_2(1) \frac{1}{\lambda^2} \beta 
\]
\[
= 1 + \Phi_2(1-\beta) \left[ \frac{e^{\bar{\sigma}} - 1}{\lambda} + \frac{1}{\lambda^2} \right] \beta 
\]
A.2 Proof of Corollary 5

Proof. We show that Corollary 5 is true for each of the above 6 cases. We begin with the cases where \( \sigma > 1 \).

(1) Consider case (i) above with bound 1 + \Phi_1(1-\beta). We show that this bound is better than that of the deterministic algorithm, i.e., 1 + \Phi_1(1-\beta) \leq 1 + \frac{1}{\lambda}(1-\beta) for \( \lambda \in (0, 1] \) and \( \beta \in [0, 1] \). In other words, we need to show \( \Phi \leq \frac{1}{\lambda} \), i.e., \( \Phi(\lambda) = e^\lambda - 1 - \lambda \). It is easy to check that \( f(\lambda) \) is increasing in \( (0, 1] \), hence \( f(\lambda) \geq f(0) = 0 \).

(2) The result for case (i) holds, since case (ii) is upper bounded by case (i).

(3) Consider case (iii) above with bound 1 + \lambda^2 \Phi_1(1-\beta). We show that this bound is better than that of the deterministic algorithm, i.e., 1 + \lambda^2 \Phi_1(1-\beta) \leq 1 + \lambda(1-\beta) for \( \lambda \in (0, 1] \) and \( \beta \in [0, 1] \). Given \( \Phi_1 \) we only need to show \( \frac{\lambda^2}{\lambda^2 - 1} \beta \leq \lambda \), i.e., \( \frac{1}{\lambda^2 - 1} \beta \leq \lambda \), which holds true from case (i).

So, cases (1) - (3) demonstrate that when \( \bar{\sigma} > 1 \), rOnMLEEng dominates the robustness and consistency bounds of OnMLEEng.

New, we consider the cases where \( \bar{\sigma} \leq 1 \). (4) Consider case (iv) above with bound 1 + \Phi_2(1-\beta)((e^{1/\lambda} - 1)(1 - \frac{1}{\lambda^2}) + \frac{1}{\lambda^2} \beta)). We show that this bound is better than that of OnMLEEng, i.e., 1 + \Phi_2(1-\beta)((e^{1/\lambda} - 1\beta)) \leq \frac{1}{\lambda^2}(\frac{1}{\lambda^2}) + \frac{1}{\lambda^2} \beta)).

To show this, we have
\[
\Phi_2((e^{1/\lambda} - 1)(1 - \frac{1}{\lambda^2}) + \frac{1}{\lambda^2} \beta) \leq \frac{1}{\lambda^2}(\frac{1}{\lambda^2}) + \frac{1}{\lambda^2} \beta) 
\]
\[
\Rightarrow (e^{1/\lambda} - 1)(1 - \frac{1}{\lambda^2}) + \frac{1}{\lambda^2} \beta \leq 1 \n\]
\[
\Rightarrow (e^{1/\lambda} - 1)^2 \leq \lambda^2 \beta 
\]
\[
\Rightarrow \frac{e^{1/\lambda} - 1}{\lambda^2} \beta \leq \frac{1}{\lambda^2} \beta 
\]
\[
\Rightarrow 0 - (e^{1/\lambda} - 1)^2 \leq 0 + \frac{1}{\lambda^2} \beta 
\]
\[
\Rightarrow (d) \leq 0 + \frac{1}{\lambda^2} \beta 
\]
where (d) holds true from \( (e^{1/\lambda} - 1 - \frac{1}{\lambda^2}) \). The inequality holds true, thus the robustness bound of rOnMLEEng dominates OnMLEEng when \( \bar{\sigma} \leq 1 \).

(5) The result for case (iv) holds, since case (v) is upper bounded by case (iv).

(6) Consider case (vi) above with bound 1 + \Phi_2(1-\beta). We show that this bound is better than that of OnMLEEng, i.e., 1 + \Phi_2(1-\beta) \leq 1 + \lambda(1-\beta), i.e., \( \Phi_2 \leq \lambda \). It is easy to show that e^{1/\lambda} - 1 \geq \frac{1}{\lambda^2}.

Then 1 + \lambda^2(1-\beta) \leq \frac{1}{\lambda^2}, i.e., \( \Phi_2(1-\beta) \leq \lambda \). Thus the competitive ratio dominates the consistency bound of OnMLEEng. So cases (3) - (6) demonstrate that the rOnMLEEng dominates the robustness and consistency bounds of OnMLEEng when \( \bar{\sigma} \leq 1 \).

B A RANDOMIZED ALGORITHM WITH DIRECT EXTENSION OF THE EXISTING RANDOMIZED ALGORITHM

The goal in this section is to show that a naive incorporation of the ML advice in designing a randomized algorithm lead to an algorithm that is neither robust nor consistent. Specifically, we show that a randomized algorithm that modifies the distribution function proposed in Equation (2) fails to achieve both robustness and consistency at the same time. In particular, a first attempt to change the distribution function is to naturally modify according to the enhancements in deterministic algorithms and obtain the following functions:

If \( \bar{\sigma} > 1 \):
\[
f_1^\star(s) = \begin{cases} 
\Phi_1 e^{s}, & s \in [0, \lambda]; \\
\Phi_1 \beta \delta(0), & s = \infty; \\
0, & \text{otherwise.}
\end{cases}
\]

If \( \bar{\sigma} \leq 1 \):
\[
f_2^\star(s) = \begin{cases} 
\frac{e^{s - 1}}{e^{1/\lambda} - 1} \beta, & s \in [0, 1/\lambda]; \\
\Phi_2(1-\beta)(1 - \frac{1}{\lambda^2}) \beta, & s = \infty; \\
0, & \text{otherwise.}
\end{cases}
\]
Our analysis below demonstrates that with these functions, rOnMLEng is large \( \min \left\{ \frac{1}{1-\beta}, \frac{1}{\beta} \right\} \), robust and (1/\beta)-consistent.

This means that with above distribution functions the consistency could be large as \( \beta \) approaches 0.

(i) \( \hat{\sigma} > 1, \sigma \leq \lambda < 1 \). This is an incorrect prediction scenario. The expected cost is given by

\[
\int_{h(s, \sigma)} f_1^*(s) ds = \int_{0}^{\sigma} \left[ 1 + \frac{1 - \sigma + s}{\sigma} (1 - \beta) \right] \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\lambda} \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\lambda} \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\sigma} \left[ \frac{1 - \sigma + s}{\sigma} (1 - \beta) \right] \frac{e^s}{e^1 - 1 + \beta} ds + \frac{1 - \beta}{e^1 - 1 + \beta} + \frac{\beta}{e^1 - 1 + \beta} = \frac{e^1}{e^1 - 1 + \beta} .
\]

(ii) \( \hat{\sigma} > 1, \lambda \leq \sigma < 1 \). This is an incorrect prediction scenario. The expected cost is given by

\[
\int_{h(s, \sigma)} f_1^*(s) ds = \int_{0}^{\lambda} \left[ 1 + \frac{1 - \sigma + s}{\sigma} (1 - \beta) \right] \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\sigma} \frac{1 - \sigma + s}{\sigma} (1 - \beta) \frac{e^s}{e^1 - 1 + \beta} ds + \frac{1 - \beta}{e^1 - 1 + \beta} + \frac{\beta}{e^1 - 1 + \beta} = \frac{e^1}{e^1 - 1 + \beta} .
\]

(iii) \( \hat{\sigma} > 1, \lambda < 1 < \sigma \). Note this is a correct prediction scenario. The expected cost is given by

\[
\int_{h(s, \sigma)} f_1^*(s) ds = \int_{0}^{\lambda} \left[ 1 + \frac{s(1 - \beta)}{(1 - \beta) + 1} \right] \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds + \frac{1}{e^1 - 1 + \beta} + \frac{\beta}{e^1 - 1 + \beta} = \frac{e^1}{e^1 - 1 + \beta} .
\]

(iv) \( \hat{\sigma} \leq 1, 1 \leq \frac{1}{\lambda} < \sigma \). This is an incorrect prediction scenario. The expected cost is given by

\[
\int_{h(s, \sigma)} f_2^*(s) ds = \int_{0}^{1/\lambda} \left[ 1 + \frac{s(1 - \beta)}{(1 - \beta) + 1} \right] \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{1/\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds + \frac{1}{e^1 - 1 + \beta} + \frac{\beta}{e^1 - 1 + \beta} = \frac{e^{1/\lambda}}{e^{1/\lambda} - 1 + \beta} \left[ 1 + \frac{(1/\lambda - 1)(1 - \beta)}{1 - \beta} \right] + \frac{1}{e^{1/\lambda} - 1 + \beta} + \frac{\beta}{e^{1/\lambda} - 1 + \beta} = \frac{1}{e^{1/\lambda} - 1 + \beta} .
\]

where (c) holds since \( (\sigma - 1)\beta + 1 \geq 1 \), and (d) is true since \( 0 \leq 1 - \beta \leq 1 \). However, note the following upper bound also holds:

\[
\int_{0}^{1/\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds \leq \frac{1}{e^{1/\lambda} - 1 + \beta} .
\]

Then the competitive ratio in this case is

\[
\int_{h(s, \sigma)} f_2^*(s) ds \leq \left\{ \frac{1}{\beta}, \frac{1}{\lambda} \right\} \cdot \frac{e^{1/\lambda}}{e^{1/\lambda} - 1 + \beta} .
\]

(v) \( \hat{\sigma} \leq 1, 1 \leq \frac{1}{\lambda} < \sigma \). This is an incorrect prediction scenario. The expected cost is given by

\[
\int_{h(s, \sigma)} f_2^*(s) ds = \int_{0}^{1/\lambda} \left[ 1 + \frac{s(1 - \beta)}{(1 - \beta) + 1} \right] \frac{e^s}{e^1 - 1 + \beta} ds + \int_{0}^{1/\lambda} \frac{s(1 - \beta)}{(1 - \beta) + 1} \frac{e^s}{e^1 - 1 + \beta} ds + \frac{1}{e^{1/\lambda} - 1 + \beta} + \frac{\beta}{e^{1/\lambda} - 1 + \beta} = \frac{e^{1/\lambda}}{e^{1/\lambda} - 1 + \beta} \left[ 1 + \frac{(1/\lambda - 1)(1 - \beta)}{1 - \beta} \right] + \frac{1}{e^{1/\lambda} - 1 + \beta} + \frac{\beta}{e^{1/\lambda} - 1 + \beta} = \frac{1}{e^{1/\lambda} - 1 + \beta} .
\]

where (d) is true since \( 1 \leq \sigma \leq 1/\lambda, \sigma - 1/\lambda < 0, \) and \( \sigma - 1 \geq 0 \).

Then the competitive ratio in this case is

\[
\int_{h(s, \sigma)} f_2^*(s) ds \leq \left\{ \frac{1}{\beta}, \frac{1}{\lambda} \right\} \cdot \frac{e^{1/\lambda}}{e^{1/\lambda} - 1 + \beta} .
\]
(vi) $\hat{\sigma} \leq 1, \sigma \leq 1 < 1/\lambda$. Note this is a correct prediction scenario. The expected cost is given by

$$
\int_0^\infty \left[ 1 - \frac{e^{s\lambda} + s}{\sigma (1 - \beta)} \right] e^{s\lambda - 1 + \beta} ds + \int_0^{1/\lambda} \frac{e^{s\lambda}}{e^{s\lambda - 1 + \beta}} ds(1) \frac{1}{e^{s\lambda - 1 + \beta}}
$$

Then the expected cost of the randomized algorithm is

$$
\int_0^{1/\lambda} \frac{e^{s\lambda}}{e^{s\lambda - 1 + \beta}} ds + \int_0^\infty \left[ 1 - \frac{e^{s\lambda} + s}{\sigma (1 - \beta)} \right] e^{s\lambda - 1 + \beta} ds
$$

We consider Algorithm 6 with $T_I, T_g$, and an arbitrary deterministic algorithm $A$ modeled by Algorithm 6 with $T_I, T_g$.

Let $t^*$ be the last timeslot before $|T_l|$ in the new demand ordering $e^*(t)$ with nonzero demand, i.e. $t^*$ is defined by

$$
t^* = \max_{t \leq |T_l|} e^*(t) = 1.
$$

We will show that the switching parameter algorithm will switch at time $t^*$. In other words, the timeslots chosen for local generator and the grid will be the same as $T_I$ and $T_g$ except for some timeslots from $t^*$ to $|T_l|$ with 0 demand. As a results, the cost is equivalent since timeslots with 0 demand contribute nothing to the cost.

Since $e^*(t) = 0$ for $t^* < t \leq |T_l|$, we have

$$
s = \frac{1}{p_m} \sum_{t=1}^{|T_l|} (p_g - p(t)) e^*(t)
$$

where (e) is from Algorithm rOnMLEng.

C PROOF OF LEMMA 2

We show that an arbitrary deterministic algorithm can be expressed by a deterministic algorithm with a switching parameter.

We first define two algorithms: (i) Generic-set-selection, a deterministic algorithm that is not limited by a switching parameter; and (ii) Converted-switching-parameter, rearranges price and demand $p(t), e(t)$ such that Generic-set-selection is replicated.

Algorithm 6 Generic-set-selection

Use local generator for a set of timeslots $T_I \subseteq T$, and use grid electricity starting for the set of timeslots $T_g = T - T_I$.
Algorithm 7 Converted-switching-parameter

Let the timeslots of \( T_l \) be specified \( T_l = \{l_1, l_2, \ldots, l_{|T_l|}\} \) and \( T_g \) be specified \( T_g = \{g_1, g_2, \ldots, g_{|T_g|}\} \).

Define a new ordering of price and demand \( p'(t), e'(t) \) according to:

\[
p'(t) = \begin{cases} 
    p(l_t), & \text{if } t \leq |T_l|, \\
    p(g_t), & \text{otherwise}.
\end{cases}
\]

\[
e'(t) = \begin{cases} 
    e(l_t), & \text{if } t \leq |T_l|, \\
    e(g_t), & \text{otherwise}.
\end{cases}
\]

Choose switching parameter \( s \) according to:

\[
\frac{1}{p_m} \sum_{t=1}^{|T_l|} (p_g - p'(t))e'(t) = s.
\]

Under the new ordering, use local generator first and switch to the grid electricity starting at the first time \( \tau \) where

\[
\sum_{t=1}^\tau (p_g - p'(t))e'(t) \geq s \cdot p_m.
\]

\[
e = \frac{1}{p_m} \sum_{t=1}^{t^*} (p_g - p'(t))e'(t) + 0,
\]

i.e., \( \sum_{t=1}^{t^*} (p_g - p'(t))e'(t) = s \cdot p_m \).

Similarly, since \( e'(t^*) = 1, p'(t) < p_g, \forall t \), we have

\[
s = \frac{1}{p_m} \sum_{t=1}^{t^*} (p_g - p'(t))e'(t)
\]

\[
= \frac{1}{p_m} \sum_{t=1}^{t^*-1} (p_g - p'(t))e'(t) + \frac{(p_g - p'(t^*))e'(t^*)}{p_m}
\]

\[
> \frac{1}{p_m} \sum_{t=1}^{t^*-1} (p_g - p'(t))e'(t),
\]

i.e., \( \sum_{t=1}^{t^*-1} (p_g - p'(t))e'(t) < s \cdot p_m \).

Therefore the first time \( \tau \) where \( \sum_{t=1}^\tau (p_g - p'(t))e'(t) \geq s \cdot p_m \) will be at \( \tau = t^* \).

Timeslots \( 1, \ldots, t^* \) are selected for the local generator. These are correctly assigned since \( t^* \leq |T_l| \). If \( t^* < |T_l| \), then timeslots \( t^* < t \leq |T_l| \) are incorrectly assigned to the grid. However, \( e'(t) = 0 \) for \( t^* < t \leq |T_l| \), which means there is no difference in cost. Timeslots \( |T_l|+1, \ldots, T \) are correctly assigned to the grid. Therefore switching at time \( t^* \) in the new ordering has an equivalent cost as assigning \( T_l \) and \( T_g \).
Bi-directional EV charging is an exciting direction in next-generation transportation systems that promises to bring many benefits, from improved economics and grid efficiencies to reduced carbon emissions. This paper presents a valuable overview of recent works in Grid-to-Vehicle (G2V) and Vehicle-to-Grid (V2G) control algorithms. Many recent research papers are discussed, including operations of single and multiple charging stations coupled with transportations and power networks, as well as energy trading in electricity market to provide services to the power grid.

Envisioning the future of transportation as the Internet of Electric Vehicles (IoEV), this paper introduces readers to the many potential applications of G2V and V2G services. A case study further highlights the economic benefit of joint optimization of routing and charging scheduling of multiple EVs in the IoEV. Potential areas for improvement include extending the case study to include other algorithms that jointly optimize charge scheduling and routing, as well as in-depth discussions on more complex experimental setups and scenarios.

We hope that this paper will help bring excitement to the area of G2V and V2G and highlight future research direction in IoEV.

/Public review written by
Xiaofan (Fred) Jiang
Columbia University, USA
A Holistic Review on Advanced Bi-directional EV Charging Control Algorithms

XIAOYING TANG, School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), and the Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
CHENXI SUN, School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), and the Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
SUZHI BI, Shenzhen University, China
SHUOYAO WANG, Shenzhen University, China
ANGELA YINGJUN ZHANG, The Chinese University of Hong Kong, China

The rapid growth of electric vehicles (EVs) has promised a next-generation transportation system with reduced carbon emission. The fast development of EVs and charging facilities is driving the evolution of Internet of Vehicles (IoV) to Internet of Electric Vehicles (IoEV). IoEV benefits from both smart grid and Internet of Things (IoT) technologies which provide advanced bi-directional charging services and real-time data processing capability, respectively. The major design challenges of the IoEV charging control lie in the randomness of charging events and the mobility of EVs. In this article, we present a holistic review on advanced bi-directional EV charging control algorithms. For Grid-to-Vehicle (G2V), we introduce the charging control problem in two scenarios: 1) Operation of a single charging station and 2) Operation of multiple charging stations in coupled transportation and power networks. For Vehicle-to-Grid (V2G), we discuss how EVs can perform energy trading in the electricity market and provide ancillary services to the power grid. Besides, a case study is provided to illustrate the economic benefit of the joint optimization of routing and charging scheduling of multiple EVs in the IoEV. Last but not the least, we will highlight some open problems and future research directions of charging scheduling problems for IoEVs.

CCS Concepts: • General and reference → Surveys and overviews.

Additional Key Words and Phrases: Electric Vehicle (EV), bi-directional charging control, V2G

1 INTRODUCTION

As an environmental friendly substitute for traditional fuel-powered vehicles, electric vehicles (EVs) lie at the heart of future sustainable and smart transportation systems. The rapid development of EVs and charging facilities is driving the evolution of Internet of Vehicles (IoV) to Internet of Electric Vehicles (IoEV). However, uncontrolled EV charging can result in expensive power generation cost, transmission congestion, and even cause security issues to the smart grid [32].

The recent development of smart grid technology provides a new set of tools that enable more secure and efficient EV charging. For instance, advanced charging facilities enable both Grid-to-Vehicle (G2V) and Vehicle-to-Grid (V2G) power flows such that EVs can act as not only electricity load consumers, but also energy providers that compensate for the power deficiency in peak load hours. Besides, EVs can act as mobile energy storage to transport excessive energy generated by remote renewable sources to the main grid [18]. Furthermore, the emerging Internet of Things (IoT) technology provides a platform to control various loads and manage the charging facilities [40]. By supporting rapid and secure data collection, distribution, and information exchange, IoT technology enables many advanced data processing and performance optimization technologies in power grids and city transportation systems that would significantly enhance the EV charging efficiency.

As shown in Fig. 1, IoT interconnects a massive number of EVs, charging facilities and other critical components that affect the performance of the IoEVs. For instance, different types of energy generators lead to different location-dependent electricity prices; the power consumptions of commercial and residential buildings cause time-varying electricity prices; energy storage can absorb the excess energy generated from renewables to meet energy deficiency of the EVs during peak hours. In particular, 4G/5G, IEEE 802.11p and other wired/wireless communication technologies enable real-time data collection and information exchange among different components, such as pricing and congestion conditions. For instance, EVs can operate in a vehicle-to-vehicle (V2V) mode, where IEEE 802.11p is used to exchange road conditions with neighboring EVs, or in a vehicle-to-infrastructure (V2I) mode, where LTE/4G is used to receive electricity charging prices updated and broadcasted by the system operator. Due to the resource constraints of IoEVs, data processing is often delegated to cloud servers with strong computational power. As such, mobile edge/cloud computing enhances the functionality of IoEVs in terms of data storing, processing, dissemination and fast computation.

The routing and charging behaviour of EVs leads to real-time interactions between IoEV, smart grid, charging stations, and transportation system. As shown in Fig. 2, IoEV can collect not only the information about price and waiting time from charging stations at different locations, but also the real-time traffic information from the intelligent transportation system. Based on the received information, each EV makes sequential decisions for charging/discharging and routing planning, and informs the charging station of interest about its status, such as charging demand, estimated arrival and
Fig. 1. Illustration of IoEVs and related energy systems. IoT interconnects a massive number of EVs, charging facilities and other critical components that affect the performance of the IoEVs.

Fig. 2. An overview of multi-layer system and interaction process of IoEV, smart grid, charging stations and transportation system. IoEV collect not only information from charging stations at different locations, but also real-time traffic from the intelligent transportation system. In the meantime, IoEV can send charging information to charging station of interest and update the routing planning to the transportation system.

parking time, battery capacity, etc. The EV may also update its information, such as location and route planning, to the transportation system. In the meantime, based on the received information from IoEV, charging stations can update the pricing or incentive schemes to maximize the system utilities, and update the electricity requests to the smart grid. In general, the challenges of bi-directional IoEVs charging control lie in two aspects. The first challenge is due to system randomness, including the random charging profiles of EVs (arrival, departure, charging demand, state of charge (SoC), etc.),
random future load demand in the main grid, random renewable energy generations and random electricity prices. The other challenge is due to the coupling effect between the smart grid and transportation network in the sense that an EV can only replenish or discharge its battery at charging stations in its route. As such, IoEV charging scheduling, charging station selection, and routing decisions are strongly coupled.

In the remainder of the paper, Section 2 presents the problem setting and charging control techniques for a single charging station and for multiple charging stations in coupled transportation and power networks. Section 3 describes how EVs can perform energy trading in the electricity market and provide ancillary services to the power grid. Section 4 provides a case study to illustrate the economic benefit of the joint optimization of routing and charging scheduling of multiple EVs in the IoEV. Section 5 highlights some open problems and future research directions of efficient charging control for IoEVs. Lastly, we draw conclusions in Section 6.

2 GRID-TO-VEHICLE (G2V)

2.1 Operation of a single charging station

A general interaction process of charging station, EVs and utility company is shown in Fig. 3. EVs arrive at the charging station randomly, and each EV expects the charging station to fulfill its charging demand within an expected time period. At each time slot, the charging station receives the current electricity price from the utility company and broadcasts a charging price to all arriving EVs. When an EV arrives, it attempts to minimize its charging cost by determining its charging demand according to the charging price, and then sends the demand request to the charging station. Based on the requested demand, the charging station decides whether to admit the EV in order to avoid excessive delay of admitted EVs. Once admitted, the EV enters the service zone where there are several parking lots and charging ports. Then, the charging station schedules the charging power/rate for each EV plugged in the charging ports and collects charging fees from the EV users accordingly. In some practical scenarios, the number of charging ports may be smaller than that of the admitted EVs, so that the EVs may have to wait for charging after being admitted. In this case, the waiting time of the admitted EVs is sent back to the controller of the charging station. The charging station has the following control methods to optimize the charging performance.

Pricing Scheme: Pricing is a type of demand-response mechanisms where EVs adjust their charging demands according to the charging price announced by the charging stations. Therein, the charging station can design the pricing scheme to control the charging demands of EVs to maximize its overall profit and system efficiency.

Admission Control: An admission control policy refers to the policy where the charging station selectively admits EV’s charging requests. Admission control is very useful to reduce the charging waiting time. Here, the charging waiting time is defined as the time between the arrival time of an EV and the time that the EV starts to receive service. The charging capacity of a charging station is limited by two factors: 1) the total charging power of a charging station bounded due to physical and security constraints of the distribution network; 2) the number of EVs that a charging station can accommodate limited by the hardware and space constraints.

A long charging waiting time degrades the users’ experience, and further negatively impacts the charging station’s long-term profit. In practice, a waiting time penalty should be considered in the system model. A naive admission control method is the queue-length based admission (QBA) policy, where a newly arrived EV is admitted only if the number of EVs waiting to be served at the station is below a specific threshold. However, it may perform poorly due to the neglection of the user demand differences [51]. Therefore, a good admission control design should be based on the actual charging demands brought by the EVs.

Scheduling: Charging scheduling refers to the sequential decisions made by charging stations on how much power to charge each admitted EV at each time. The decisions are generally made based on the past and current information of EVs that have already arrived. There exists some greedy charging scheduling methods that maximize the current revenue without considering the unknown future charging demands. These approaches may suffer a high penalty in the future, e.g., paying higher electricity price or penalty because of low service quality [44, 45]. In contrast, a good online charging scheduling decision should take into account the random future events, which include the demand, time of arrival and departure of EVs, elastic and inelastic load demand in the power system, renewable generations, realtime electricity prices and regulation service prices, etc.

In recent years, many charging scheduling algorithms have been developed under a variety of settings. [19] presented a flexible adaptive scheduling algorithm based on convex optimization and model predictive control and allows for significant over-subscription of electrical infrastructure. [26] studied the real-time operation of a public charging station providing charging service to large-scale Plug-in Electric Vehicles (PEVs). [53] introduced a new policy called least laxity ratio to achieve a suitable notion of proportional fairness. [39] proposed an operating model that can be used both for the day-ahead scheduling and for the intraday model predictive control (MPC) based adjustments, assuming that both the charging stations and the EV fleets belong to the same company. [27] proposed a new day-ahead co-optimization algorithm to reduce the detrimental effects of PEVs on the power system. [24] proposed an
optimal charging scheduling method that minimizes the operation cost by responding to the time-of-use (TOU) electricity price.

Some existing studies investigated the joint optimization of pricing and the charging scheduling schemes that benefit both EV users and the charging stations. In this case, both the charging rate and charging price are control variables of the charging stations. For example, [50] formulated the pricing and scheduling problem into an Markov decision process and proposed a reinforcement learning approach that maximizes the profit of a charging station.

There are also some studies that jointly optimize pricing and admission control schemes to maximize the total profit of charging station as well as minimize the waiting time of EV users. In this case, the control variables include the charging price, the number and total charging demands of admitted EVs, while generally simple charging schedule schemes are adopted, e.g. first come first serve with constant charging/discharging rates. The key point of jointly optimizing pricing and admission control schemes is to strike a good balance among the waiting time, admission probability, and charging port utilization. For example, [51] analyzed the EV queueing dynamics and derived the waiting time in closed-form, and accordingly proposed a novel multi-sub-process based admission control scheme in order to jointly optimize the profit of charging stations and the delay of EV users.

Some work [10, 52, 54] considered both admission control and charge scheduling strategy. [10] formulated a multi-stage stochastic programming model to minimize the expected total energy costs over the finite time horizon. [52] proposed a two-stage admission and scheduling mechanism to find the optimal tradeoff between accepting EVs and missing charging deadlines under several energy supply scenarios. [54] proposed an innovative station-level optimization framework to operate charging station with optimal pricing policy and charge scheduling.

Besides, in order to tackle the system dynamics and randomness of user behavior, data-driven model is another popular method used in energy management for a single charging station. For instance, [29] presented a new coordinated dynamic pricing model to reduce the overlaps between residential and charging station loads by inspiring the temporal PEV load shifting during evening peak load hours. [7] proposed a multi-agent multi-objective reinforcement learning architecture that aims at simultaneously minimizing energy costs and avoiding transformer overloads, while allowing EV recharging. [21] proposed a charging scheduling strategy using a safe deep reinforcement learning approach to minimize the charging cost as well as guarantee the EV can be fully charged. [48] proposed a two-stage energy management system for power grids with massive integration of EVs and renewable energy resources. In [37], a smart reservation system considering the behavior of EV users, parking slot availability, SoC value of EVs, and the parking lot usage history of EV users was proposed. [22, 23] demonstrated that utilizing real EV charging data, a method that combines reinforcement learning and predictive control can provide lower charging costs than MPC. This line of research utilized historical data such as load, usage history and smart meter measurements to develop effective models for charging station operation.

Fig. 4. An illustration of the IoEV system coupled with both the smart grid and transportation network

2.2 Operation of charging stations (in coupled transportation and power networks)

In this section, we introduce the joint optimization of routing and charging scheduling for IoEV operation control in transportation network equipped with multiple heterogeneous charging stations. Optimal routing is a classic problem in conventional transportation networks that aims to minimize travel time, traversed distance and/or energy consumption, etc. The optimal routing problem is mostly modeled as a shortest path problem in a graph, with some variations of edge weights to consider road congestions, regulations and user preferences. Commonly used methods include shortest path Dijkstra algorithms, A* based-search algorithm, Ant Colony optimization, Particle Swarm Optimization, and Tabu Search [1].

The conventional routing algorithms cannot be directly adopted to IoEV. Due to the coupling effect between the smart grid and transportation network, the route selection is also coupled with the charging station selection along the selected path, and thus related to the operation of the power system. Specifically, unlike gasoline price, electricity price can be significantly different at different charging facilities. For instance, as shown in Fig. 4, the electricity price at node B with renewable energy source is likely to be much cheaper than that at node A, which is powered by the main grid. Therefore, an EV user may have the incentive to take a detour to charge its battery at station B instead of taking the shortest path. Besides, the V2G technology allows an EV to sell energy back to the grid for profit. Under this condition, an EV user may consider the potential profit in route selection by first charging at stations with a lower electricity price (e.g., node B) and then discharging its battery at stations with a higher electricity buying price (e.g., node A). Moreover, the electricity price and availability of charging facilities are related to other electricity consumers, such as commercial users and households. For instance, renewable energy is scarce in urban areas with high household consumption but abundant in suburban areas. As a result, EV routing must be jointly optimized with charging station selection by taking into account all the elements in the system.
In Fig. 4, we use a simple example to show how the electricity prices and V2G technologies influence the routing and charging scheduling of an EV. Suppose that an EV travels from node S to node T. Node B is powered by renewables and node A can buy energy from the EVs. We assume that the initial battery level of a tagged EV is sufficient to complete both paths S – A and S – B. To maximize the profit of the EV, the optimal routing is S – B – A – T and the optimal charging scheduling is to fully charge the EV at node B and discharged (sell the extra electricity) at node A given that the EV can complete each road segment in the path. On the other hand, if V2G is not available at node A, the EV will choose the path S-B-T with the minimum energy cost. In addition, if renewable energy is not available at node B, the EV will select S-A-T with the shortest path and minimum energy consumption.

The problem becomes much more complicated when a large number of EVs plan their routes at the same time. Uncoordinated planning may lead to overwhelming charging demands at bottleneck charging stations. Therefore, it is necessary to coordinate the route selection and charging scheduling of EV users to maximize the system performance. In this case, a major challenge is to design the right incentive scheme, so that the EVs’ and system operators’ selfish decisions are also the maximizer of the social welfare. Besides, it is desirable to find scalable algorithms to solve the large-scale routing and charging scheduling problems with affordable communication and computation overheads.

One way to jointly optimize the routing and charging scheduling of a single EV is to model the problem as an extended transportation graph and find a shortest path [2]. In the case of multi-EV coordination, the decisions of individual EV users are coupled due to the constraint of limited traffic and charging station capacity. There are two types of control schemes for multi-EV coordination, namely, centralized schemes [6] and distributed schemes [5, 47]. Notice that centralized schemes require the EV users to submit their complete information, resulting in serious privacy concerns. In contrast, distributed algorithms only require little information exchange between the EVs and the system operator, thus significantly reducing the privacy leakage and the complexity of computation and communication compared with centralized schemes. For instance, [47] proposed a proximal method based distributed algorithm, where the EV users are not required to share their specific route selection with the system operator.

Some recent studies investigated the joint routing and charging problem from a social coordinator’s perspective [36, 41, 55] where the EV owners aimed to find their own optimal charging station and the social coordinator designed pricing strategies such as congestion tolls and locational marginal prices (LMP) to influence or guide EV charging and routing behaviors. In [36], a bi-level model was proposed to determine the optimal charging service fees for guiding EVs and minimizing the social cost that includes the total driving time, waiting and charging times in transportation network, and total generation cost in power network. [41] proposed an online recommendation and charging schedule algorithm with on-arrival commitment for sequential EV arrivals that aims to maximize the expected total revenue of a charging station network. [55] adopted an expanded transportation network model to describe transportation constraints and the AC power flow model to describe electrical constraints and proposed a second order cone programming model to minimize the total social cost that includes driving and charging time costs of PEV drivers and power supply costs.

Some studies considered a Charging Network Operator (CNO)’s perspective [9, 30, 31] where the EV owners cannot directly choose the charging station but are rather assigned to certain stations by a central controller based on the optimization objectives. In this setting, users can specify their desired SoC and their destinations to the CNO and the CNO will assign each EV to an optimal charging station based on the charging request. Specifically, [9] developed an EV assignment algorithm based on the Lyapunov optimization method that aims to minimize the average time spent from requesting the service to accessing it. [30] formulated an integer multi-objective optimization problem for optimal coordination of a fleet of cooperative EVs considering the objectives of EV owners, charging station owners, and power systems. [31] designed pricing and routing policies that ensure users reveal their true needs to the CNO and directly assigned them to a station on their path in order to manage their effects on the grid and ensure fair services. This line of work formulated the charging control of EVs in a charging station network as a decision problem of the CNO and focused on developing appropriate methods to find the optimal solution.

3 VEHICLE-TO-GRID (V2G)

3.1 Energy trading in the electricity market

With the implementation of Vehicle-to-Grid (V2G) technology, EVs can also provide energy to the grid [17]. Fig. 5 gives an illustration of EVs participating in the electricity market. Specifically, due to the limited battery capacity of each EV, EV aggregator (EVA) is normally required to coordinate a collection of EVs in order to participate in the electricity market with bids to purchase or sell electricity. There are two types of electricity/energy market: day-ahead market and real-time market. The day-ahead market lets market participants buy or sell electricity one day before the operating day, and real-time market which allows market participants to buy or sell electricity during the operating day.

Most existing studies [3, 12, 13, 25, 35] considered the EVA operation in day-ahead electricity market, as EVA should plan the charging scheduling for each EV beforehand. Specifically, [13] proposed an optimal operation strategy for an EVA, which performs energy arbitrage in the energy market and provides ancillary services from aggregated EVs, while providing charging services to EVs.
to maximize the profit in a future energy market. [35] proposed a hierarchical optimization approach to represent the decision-making of this aggregator in the day-ahead electricity market. [3] proposed a day-ahead market framework for congestion management in smart distribution networks considering collaboration among EVAs. In [12], a new distributionally robust optimization (DRO) via scenario wise ambiguity set is proposed to develop a collaborative bidding strategy for intermittent resources such as EV as the day-ahead energy market. [25] considered the EVA participation not only in the electricity market, but also reserve market. These studies investigated the optimal day-ahead operation strategy of EVA and the potential benefit of collaboration among EVAs.

3.2 Ancillary Service by EVA

Apart from energy trading, EVA can also provide ancillary service to the smart grid, such as frequency regulation [8, 33, 34, 38, 49], voltage control [11, 15, 28, 43], and reserve service [4, 14] etc. Ancillary services provide the resources the system operator requires to maintain the instantaneous and continuous balance between power generation and load demand in a reliable manner. Fig. 6 gives an illustration of EVs providing ancillary services via EVAs.

3.2.1 Frequency regulation. Frequency regulation is an ancillary service that aims to maintain the frequency of the grid around its nominal value (50 Hz or 60 Hz) by controlling the frequency variations caused by imbalances between power generation and load demand. Typically, frequency control contains three phases with different timescales. The primary frequency control, also known as droop control is usually triggered within a few seconds. The secondary frequency control, also known as automatic generation control (AGC) is triggered within minutes. The tertiary frequency control, namely economic dispatch is triggered within a few minutes if the frequency deviation event does not correct itself through primary or secondary frequency control mechanisms. Due to the fast response time and high ramp rates of Battery Energy Storage System (BESS), primary frequency control at load-side can be provided by single BESS directly or by multiple small-scale BESSs coordinated by a battery aggregator [59]. For example, [58] derived the optimal planning and control strategy for BESSs participating in the primary frequency control regulation market.

Recently, some studies investigated the potential of EVs to provide frequency regulation services when they are plugged into the grid. Specifically, regulation-down can be done by charging the PEV batteries from the grid, and regulation-up can be achieved by discharging the PEV batteries to the grid. For instance, [8] proposed an online rolling decoder-dispatch framework for the frequency management of electrical-grid-electric-vehicle systems. [49] proposed a state-space based EVA modeling and control method for frequency regulation. [33] proposed a control scheme to involve the aggregated EVs in frequency regulation by using a tube-based model predictive control in conjunction with a disturbance observer control. [34] considered an event-triggered mechanism (ETM) for multiple frequency services of electric vehicles (EVs) in smart grids. [38] proposed an event triggered control based switching approach for frequency regulation with EV participation. [56] proposed a hierarchical system model to jointly optimize power flow routing and V2G scheduling for providing regulation service.

3.2.2 Voltage control. Voltage control aims to keep the voltage magnitudes in the smart grid close to the nominal values through injection or digestion of reactive power. Conventionally, voltage control is performed in a centralized manner to determine the day-ahead dispatch of on-load tap changer (OLTC), voltage regulators or capacitor banks, which lack the fast-response capability and are ineffective to mitigate fast voltage violation in real time.

Recently, some work proposed several control mechanisms that utilize the dispatch of EVs [11, 15, 28, 43]. [43] proposed a three-layer hierarchical voltage control framework to mitigate fast voltage violation problems with the dispatch and control of EVs. [15] presented an optimization model to flexibly control available PEV battery charging/discharging power based on three-phase power flow and sensitivity approaches. [11] proposed a two-stage centralized approach to level the power mismatch between the demand forecast and the real time demand in medium voltage grids by means of fast charging stations. [28] proposed a novel optimal hybrid control framework to improve the voltage profile of highly unbalanced Distribution Grids by coordinating the injection of reactive power from multiple off-board EV chargers.

3.2.3 Reserve Service. An operating reserve (spinning reserve, supplemental reserve, replacement reserve) is a power source that can quickly be dispatched to ensure that there is sufficient energy generation to meet load in response to a major generator or transmission outage. Spinning reserves are power sources that are already online, synchronized to grid, and can rapidly increase their power output to meet fast changes in demand. Supplemental reserves can be offline and need not to respond immediately. Replacement reserves are used to restore spinning and supplemental reserves to their pre-contingency status.

There are some work that studied the potential benefits of EVs in providing reserve services [4, 14]. In [4], an optimization model and two operational management algorithms were described for supporting the participation of an EVA in the day-ahead energy and manual reserve market sessions. [14] simulated the potential monetary benefit that EV could generate by providing the regulation and reserve power to the Dutch market.
charging station is typically located in the central of city, which has good connection to the power grid but insufficient space to deploy PV panels. In Fig. 7, there are four type 1 charging stations, located in suburb areas, and five type 2 charging stations, located in downtown areas. Type 2 charging stations sell (buy) the electricity to (from) the EVs at a price of $18/kWh ($8/kWh). Type 1 charging stations sell renewable energy to the EVs at a very low price of $1/kWh. The total amount of renewable energy in each type 1 charging station is set to be 30kWh. Suppose that EV 1, 2, and 3 travel from source node s1, s2, and s3 to destination d1, d2, and d3, respectively, as shown in Fig. 7. For each EV, the battery capacity is set to 15kWh and the initial SoC at the source node is set to 0.5. The charging efficiency is 0.9. Due to the limited capacity of charging station, the route selections of all the EVs are coupled, necessitating a joint optimization across the system.

To reduce the computational complexity and information exchange, we proposed a distributed routing scheme in [47], where each EV user’s selfish behavior to maximize their own profits also leads to the maximum social surplus. Specifically, the distributed scheme in [47] optimizes the route selection of all the EVs given a set of $k$ shortest paths as candidate paths. In a special case when $k = 1$, the EVs have no choice but to travel through the shortest path from the starting point to the destination, which is equivalent to the conventional distance-based shortest path method. For each given path, the optimal charging scheduling of an EV is to charge its battery as much as possible whenever it encounters a type 1 charging station, and to sell the electricity back to the grid when it encounters a type 2 charging station under the constraint that the remaining energy is sufficient to reach the next type 1 charging station or the destination, whichever is closer. We plot in Fig. 8 the optimal total profits of all EVs under different value of $k$. It can be observed that the total profit increases when $k$ increases, as increasing $k$ enlarges the solution set of joint route selection and charging scheduling design that aims to maximize the total profit rather than just minimizing the traveling distance. The total profit also grows with the battery capacity. Specifically, when the battery capacity increases to 20kWh, the distributed scheme leads to 447.8491$ more profit when $k = 5$ compared to when $k = 1$, which demonstrates the superiority of joint route selection and charging scheduling design over conventional distance-based shortest path method in IoEV networks.

5 FUTURE DIRECTIONS

5.1 Leveraging data-driven approaches

5.1.1 For demand response. Most existing work on demand response scheme is based on mathematically convenient models that are often too simple to be practical. Recent success in machine learning holds significant potential in solving this issue by learning from massive real-world data. For instance, EV users may exhibit certain group patterns that can be identified by clustering methods [42] and Gaussian Mixture models [20]. The identified charging patterns and user preferences allow us to design more realistic models to handle user heterogeneity. The actual charging profiles of EVs can also help develop more accurate battery models, which enables the refinement on demand-response scheme design. In addition,
model-free reinforcement learning approaches have shown great success in solving demand response problems, as it can integrate user preference and adapt to the environment.

5.1.2 For charging and routing optimization. The EV routing optimization problem in [47] only considers one-stage routing optimization, i.e., one journey with a pair of starting and destination nodes for each EV. In practice, EVs need to complete multiple journeys. The future journeys are often coupled with the current routing decision and are random in general. The problem becomes even more challenging when considering the uncertainties from both transportation network and smart grid, e.g., vehicle behaviors, charging habits, time-varying electricity prices, and real-time renewable generations. Currently, there are several preliminary studies on online and stochastic EV charging scheduling to tackle different types of uncertainties under different types of knowledge of future data [45][46]. However, most of them suffer from high computational complexity. A promising solution is to exploit data-driven learning approaches to adapt the decisions to the dynamic environment. For instance, deep reinforcement learning methods can be designed to learn optimal policy for EV charging scheduling and routing.

5.1.3 For charging station planning. Another important issue is to optimize the city-wide charging station/battery swapping placement to maximize the overall IoEV efficiency. Most existing work on charging station planning focus on proposing mathematical formulation such as mixed-integer linear programming or discrete optimization under a variety of assumptions and validated the formulation using simulation. Notice that historical EV trajectory data and user behavior data are likely to reveal key information of the fine-grained charging demand at different time/location throughout a day. In addition, high-resolution population data such as LandScan and Worldpop can further facilitate the estimation of the long-term charging demand. Hence, a data-driven approach is preferred for the charging network design that can better accommodate with the spatially and temporally varying charging demand.

5.2 Exploit an economic perspective

5.2.1 Profit model for providing ancillary service. The studies so far mainly focused on designing operation and control strategies for EVs to provide ancillary service like frequency regulation and voltage control. However an important aspect is to incentivize EV users to participate in the process in practice. Unlike BESSs, the mobility of EVs brings the opportunity to respond quickly to the unexpected event of the power grid. In the meantime, a great challenge here is to characterize and model the demand and flexibility in a spatial-temporal context, which should be carefully considered while developing a profit model that can motivate EVs to provide ancillary service.

5.2.2 Cooperation and competition among different charging station operators. Most existing work investigated the operation strategy of a single charging station or a charging station operator that coordinates a number of charging stations. In reality, multiple charging station operators owned by different companies often coexist, and new investors may also enter into the market. One interesting direction is to model the cooperation and competition among different charging station operators and design pricing strategies under different perspectives and considerations. In [57], the optimal pricing contract and quantity contract for service providers is studied in the duopoly market with Bertrand competition and Cournot competition. Pricing strategies for other scenarios are worth further investigation.

5.2.3 Data privacy concern. Efficient real-time charging scheduling algorithms often require both historic and real-time data from the charging facilities/utilities and EV users. The data often contains private information, such as location information, travel destination, models of EVs, and individual consumer profiles. Hence privacy-preserving data collection and processing is a practical and interesting research problem. For example, efficient incentive schemes can be designed to ensure the truthfulness of the collected data. To encourage EV users to share the accurate data with the system operator, incentive mechanisms based on, for example, pricing, auction, and contract theory, plays an important role. In some cases, EV users are not willing to reveal their demand response functions due to the privacy concerns, causing difficulty on pricing scheme design. In this case, a viable approach is to predict the demand functions by learning from the historical data collected by the charging stations.

5.3 Integration of advanced charging facilities

Most existing algorithms of joint optimization of EV routing and charging scheduling only considers G2V. In fact, it has been shown that deployment of V2G and V2V can largely improve the flexibility of energy storage systems and benefit both EV users and smart grid. For instance, with the implementation of V2G technology, EVs have the incentive to buy cheap electricity from the renewable charging stations and then sell the electricity to the charging stations with heavy load demands. Accordingly, the routing decisions of the EVs are significantly different when the V2G technology is available. In addition, wireless charging facilities can be deployed on the roadway to charge EVs on the move. On the other hand, despite its convenience, wireless charging on the road may encourage slow driving speed, as the amount of energy charged to an EV per unit distance is inversely proportional to its driving speed. Thus, the optimization problem needs also take into account the road congestion levels. This leads to a whole host of new problems that require close coordination of IoEV, smart grid, and transportation systems.

6 CONCLUSION

In this article, we introduced the EV charging control problems in two directions: G2V and V2G. For G2V, we discussed the problem setting and charging control techniques for a single charging station and for charging stations in coupled transportation and power networks. For V2G, we illustrated how EVs can perform energy trading in the electricity market and provide ancillary services to the power grid. Besides, we highlighted some open problems and future research directions of charging scheduling problem for IoEVs. It is foreseeable that advanced charging technologies for IoEV will spur new research interests, which finally leads to a highly efficient, reliable, and sustainable smart power grid, intelligent transportation network, and smart city.
REFERENCES


Public review for

Empiricism and Collaboration on Grid Data Analytics: The Need for a New Information Ecosystem

Alexandra von Meier, Laurel N. Dunn

This ‘vision’ paper discusses the operation of the legacy grid, and why new developments, such as the loss of inertia from inverter-based generation, the integration of storage, and the availability of controllable loads require more frequent and more precise control. These needs can only be met by the collection and use of ‘big data’ and, more generally, information technology. Broad in outlook and visionary in exposition, this is an excellent paper that I am sure will be read with great interest by many!

Public review written by

Srinivasan Keshav

University of Cambridge, UK
Empiricism and Collaboration on Grid Data Analytics: The Need for a New Information Ecosystem

ALEXANDRA VON MEIER†, University of California, Berkeley, USA
LAUREL N. DUNN, PingThings, Inc., USA

This paper discusses the need for data-driven tools to manage modern electric grids, where planning and operational decisions increasingly require empirical data on various time scales. The advancement of such tools will hinge on deploying instrumentation to collect faster and more localized measurements, capitalizing on state-of-the-art software solutions to facilitate big-data workflows, and enabling open exchange of data and information with research collaborators.

CCS Concepts: • General and reference → Empirical studies; • Applied computing → Engineering; • Information systems → Data exchange.

Additional Key Words and Phrases: Electric power systems, Grid modernization, Time series data, Collaboration

1 INTRODUCTION
It is a truism that electric grids are changing. Much has been written about the need for smarter grids. Here, we step back to contemplate more broadly the environment that can enable the growth and maturation of new analytics to inform the energy transition, given the evolving physical characteristics of grids with more renewable and distributed generation. Given the complexity of the challenges that lie ahead and the societal importance of “getting it right,” this essay advocates for an empirical and collaborative approach.

Electric power systems are governed by physical laws that have been well understood since the 19th century. We can write exact, deterministic equations to describe the behavior of just about every individual component. And yet, owing to the number and diversity of interactions, the interconnected a.c. grid appears to the operator more like an animal than a machine. Formal properties of components are insufficient to predict their collective behavior with confidence, especially under unusual conditions. For this reason, operator experience—the accumulation of observational evidence, studied with awareness—has always been vitally important in this industry [13].

Traditionally, power systems planning and operation have been informed by very limited observations of electrical phenomena, and aided by automated processes informed by strictly local data. The problem of keeping the lights on—with remarkable success!—was made tractable in large part by heuristic methods and rules of thumb. In this sense, the legacy electric grid has been aptly described as “a system that works in practice, not in theory” [9].

But formal rules and human intuition alike can only go so far. Key barriers to moving beyond this paradigm are the limited availability of empirical data, and the need for analytical tools that can quickly digest this data into actionable information. In modern grids, measurement data are becoming more abundant and precise. On the one hand, this provides new and useful insights. But untamed, the quantity and interrelations in the data quickly exceed human grasp.

As we will show, the information and decision processes are becoming more interdependent and less compartmentalized, making it increasingly necessary to synthesize measurements both temporally and spatially. Distributed and fast-acting controllers are causing highly localized interventions to become more closely coupled with geographically broader concerns. This coupling causes growing overlap among automated control interventions, near-term optimization and long-term planning decisions. Consequently, the ongoing processes of managing the many dimensions of system state are becoming richer in information that cannot be reduced to the rules or approximations decision-makers have leaned on in the past.

This has several important implications: First, we need more granular measurement data from many locations. Second, we need tools to usefully digest data, referencing across time, place, and scale. And third, we need to bring together different skill sets and expertise—grid operators, control theoreticians, data scientists—to interpret these data, and to collaboratively build tools for engineering and operating the grid as it evolves.

2 THE OLD WORLD
Decisions about electric grid planning and various aspects of operations have always been segregated by time scale. For example, capacity upgrades are planned over years, informed by forecasts of daily peak load. Generation unit commitment and dispatch are compartmentalized in hourly and smaller sub-hourly increments. Customer energy billing evolved from monthly to 15-minute intervals using smart meters, enabling time-of-use pricing but sending little information (e.g., power outage status but not voltage) to distribution operations. Generally, the resolution of measurements and analysis tools match the time scale of the decisions to be made, and decision processes are kept separate.

Accounting for energy and capacity happens in the realm of steady-state analysis, where many physical details about power flow are neglected. This is quite separate from concerns about instantaneous dynamic interactions among a.c. power components in synchronous grids, whose time steps are measured in seconds, cycles, or degrees of angle.

Here it is worth remembering that electric grids as we know them were conceived, built, and successfully operated prior to computers and even telephone service. This was possible because power balance can be achieved by droop control, based on frequency as a locally measured variable, plus rotational inertia to absorb the discrepancies and transients. In the context of balancing supply and demand, the grid is described as moving from one steady-state to another—making it tractable for analysis by pencil and slide rule.

Authors' addresses: Alexandra von Meier, vonmeier@berkeley.edu, University of California, Berkeley, California, USA, 94720; Laurel N. Dunn, PingThings, Inc., Oakland, California, USA, laurel@pingthings.io.
Dynamics, like those described by the generator swing equation or angle stability across long transmission lines, are difficult to quantify and historically had to be allowed for by way of design margins. In lieu of direct monitoring to inform rapid control actions, the grid was built with extra capacity and “shock absorption” to accommodate the uncertainty due to transient phenomena that could not be captured by steady-state models.

Voltage and frequency have generally been considered as separate, decoupled control dimensions, owing to the physics of reactance-dominated transmission networks. Control interventions to regulate voltage magnitude (such as generator excitation, transformer taps, and capacitor bank switching) rarely involved much operational urgency. Like frequency regulation, bus voltage regulation by synchronous generators to absorb rapid changes can be accomplished with completely local feedback control, while optional, broader adjustments for economic purposes can happen on a slower (e.g., hourly) scale.

The big exceptions to the steady-state view of the grid are fault analysis and protection, which reach down into the sub-cycle time scale. Here we abandon not only the phasor domain for explicit time-domain analysis, but also the balanced three-phase approximation. Notably, protection has always been a distinct sub-specialty among power engineers. Its scope is primarily local: as a matter of safety, the actuation of protective devices must always be based on foolproof, instantaneous, local measurements (even if today they may also receive supporting information from afar).

Another specialty concerned with sub-cycle phenomena is power quality, which includes voltage variations, transient disturbances, and harmonic content. Even more so than protection, power quality has been treated as a local issue. The impacts of current harmonics, for example, are attenuated over distance, and are rarely urgent. Thus, while there are ample devices for making excellent power quality measurements, very few of them communicate or time-synchronize their data.

In sum, the problem of managing electric grids has been made tractable by parsing different aspects of monitoring and control into domains which, to a good approximation, don’t interact. This simplified informational environment is illustrated for the time dimension in the upper Figure 1.

3 EVOLVING NEEDS

A fundamental shift in modern grids is an increasing temporal and spatial dependency among components. In particular, the broad adoption of renewable and distributed energy generation has several distinct implications.

First, the volatility of solar and wind resources introduces significant minute-to-minute or even second-to-second variations in power flow and voltage. In areas with high penetrations of distributed generation, this local variability impacts the design considerations for both capacity and voltage regulation. Though loads have always been volatile, the maximum expected rate of change of net load has increased. From the system perspective, the problems of assuring resource adequacy and scheduling generators become more stochastic. The logical trend is to augment solar and wind resources with either storage or headroom capacity and recruit them as active providers of primary and secondary frequency regulation. But with a larger number of interacting devices, optimization has many more variables, and the information and communication requirements increase. An apt metaphor is a complex musical score for a much larger orchestra, and the challenge of conducting the performance in real-time.

Inverter-based generation also differs from conventional rotating machines in its lack of inertia. This can be both a liability and an asset. While synchronous generators rely on passive physical effects (i.e., angular momentum and feedback from magnetic forces) to maintain their rotational frequency and relative angular position, switch-controlled power electronics are free of such constraints. The loss of inertia as synchronous machines are supplanted by solar and wind farms has widely been seen as an erosion of grid-stabilizing resource. However, it may also reduce vulnerability to oscillations associated with generator swings, and offer new opportunities for much faster, active control of frequency and relative voltage phase angle [4]. Power electronics at HVDC converter stations can similarly contribute oscillation damping, if properly informed [11].

The implication is that grid dynamics and inertia are becoming intertwined with the task of actively managing generation assets, blurring the separation between the time scale domains as illustrated in the lower Figure 1.

Controllable loads and storage resources are also joining the orchestra of active and agile distributed energy resources (DER). Modern appliances, electric vehicle chargers, smart service panels, and widespread battery deployments all afford new opportunities for highly granular control in both time and space. These devices can be called upon to help balance supply and demand at the system scale, and also to address more local issues such as infrastructure capacity and voltage regulation. In doing so, they might alleviate the need for capital investments that would otherwise be needed to accommodate new loads and distributed generation. The siting and operational control of these resources thus relates to information on time scales ranging from cycles to years.

Spatially, high penetration levels of DER transform our radial distribution infrastructure into bi-directional systems. This means accounting for a wider range of operational states, including reverse power flow and inverted feeder voltage profiles, and consequently a need for planners, protection engineers, operators, and prosumers to strategize and coordinate more closely. Even the traditional division between distribution and transmission operations is blurred, as system operators want better situational awareness about DER connected behind the substation and behind the meter. On the macro-scale, remote solar and wind farms and associated long-distance transmission lines imply a continually growing need for wide-area monitoring of system dynamics. Meanwhile, by limiting redundancy or oversized investments, economic pressures tend to drive system operations closer to the margin, necessitating ever more careful vigilance.

Another dimension to consider is the likelihood of exceptional versus routine conditions. A complex system too multi-faceted and interactive to fully describe in theory can be managed effectively in practice by heuristics, but only if it adheres to a limited range of familiar operating states. With the advent of noticeable climate change effects, more frequent and extreme weather events have
begun to upend all rules of thumb for what is prudent. Increasingly, we look to infrastructure resilience rather than simple reliability. Again, this calls for careful real-time assessment of system state, along with mitigation and recovery options. Detailed and comprehensive information about operating states before, during, and after an extreme event is necessary to help decision makers to define new heuristics—or even altogether new operating strategies—that will make the system more robust to evolving climate conditions.

4 THE PATH FORWARD: THE ROLE OF DATA

As interactions of grid components and thus the interrelations of planning and operating decisions across time and space increase, so do the information requirements, with no rules of thumb to consult. For example, how much inertia does a balancing authority need? What inverter settings will prevent a common-mode trip response to a disturbance event? What is the feeder hosting capacity for behind-the-meter solar or electric vehicles, if some fraction of devices are under rigorous control? Questions like these become answerable only with a wealth of empirical measurement data to validate models and monitor real-world conditions.

Broad and effective use of such data will require major advances in instrumentation, software tools, and analytical methods that inform engineering practices. There is no shortage of potentially transformative ideas for using data to improve the grid. But maturing these ideas into tools that can be operationalized at scale and deployed with confidence hinges on improving our capacity to readily explore, test, and validate new ideas using empirical measurement data [2].

What is needed today is a focus on analytics that distill volumes of data into information streams that can provide timely and actionable insights to decision makers. Time-series measurements will play a particularly important role, as these data capture the factual physical reality of how the state of the grid evolves over time, independent of models or simplifying assumptions. Comparing this factual record against expectations—based on past experience, or based on modeled predictions—will be instrumental to inform both human and automated interventions.

A number of key enabling technologies have matured in recent years and are ready to be put together for this purpose. Grid sensors are available to monitor both electrical and mechanical properties of the grid at various time resolutions. The cost of communications bandwidth needed to support real-time data streaming has declined dramatically, as has the cost of data storage. Advancements in data management solutions make it much easier (and much faster) to explore, develop, and deploy new analytics [7]. Commercially available data storage [1] and visualization tools [8] make it easier to interface with large volumes of data, and distributed computing makes it easier for analytics to scale up as their utility is demonstrated or as data volumes continue to grow. Finally, secure cloud
storage removes barriers that have made it logistically challenging for stakeholders to exchange data with collaborators at other institutions.

Capitalizing on these advancements will allow the electric power industry to lean more heavily on data that were historically underutilized, complementing the deployment of new sensors where appropriate. Data will be instrumental in alerting key stakeholders when heuristics and simplifying assumptions used in the past are no longer valid. The alternative is to wait until something goes wrong, to retroactively determine the cause. At present, there is ample research to alert us to possible issues that grid decarbonization and climate change itself could bring. But early detection and timely diagnosis could help address emerging issues before a catastrophic failure occurs.

Collecting data at the appropriate time scale is crucial. As indicated in Figure 1, SCADA and AMI systems typically report measurements only once every few seconds or every few minutes. More advanced point-on-wave sensors, on the other hand, may report tens, hundreds, or even thousands of precisely time-stamped measurements per second, and can stream them continuously. While such high-resolution data provide more detail than is needed for legacy use cases, they are indispensable for diagnosing behaviors relevant to power electronic switching and inverter dynamics.

High-frequency data streams, however, pose challenges to conventional data analysis and visualization techniques. One solution to this problem is to present data in terms of statistical aggregates. Figure 2 shows an example of putting this principle into practice to locate waveform data recorded during a small fault.

The ability to zoom in and out to visualize data at different timescales has important implications beyond convenience. It means that a data stream from the same sensor can support diverse applications, including continuous monitoring for anomalies indicative of both known and unknown phenomena. This in turn prevents data from becoming siloed in limited use cases that are already well-understood.

For example, consider the problem of common-mode inverter tripping in response to transmission-level disturbances, such as a frequency discontinuity or voltage waveform step change (e.g., the Blue Cut Fire incident). Diagnosing such a surprise event is greatly facilitated by synthesizing time synchronized point-on-wave measurements from different locations. In the above case, available data could only illustrate the sudden loss of large solar farms; the response from rooftop PV was impossible to quantify due to limited access to measurements. Interpreting relevant data also requires diverse expertise: understanding the causes and possible magnitude of such events draws on broader system analysis; estimating disturbance propagation into the distribution system takes local knowledge; and correcting the vulnerabilities with improved ride-through standards gets into detailed inverter controls.

For another example, consider load management and microgrids for resilience. Extreme weather events such as the recent wildfires in California or winter storm Uri in Texas can necessitate drastic load shedding, sometimes with deadly consequences. Those impacts could be mitigated with more granular control strategies, which in turn require electrical visibility and coordination across areas. Moreover, the combination of distributed PV, storage and load control makes it increasingly feasible to contemplate local power islands during emergencies, for various aggregations of customers. Novel strategies for islanding distribution systems, however, need to be thoroughly vetted and carefully implemented with empirical data—from planning and balancing to re-synchronization.

Thus, as data-driven tools become increasingly necessary, solution development and adoption will hinge on efficient processes for data analysis and collaboration. Contributions from various domains will be needed to translate interesting algorithms into operational tools that offer practical, actionable intelligence. But to build such
tools, there is a need for more fluid exchange of information—both data and expertise—between industry stakeholders and data analysts [10].

A new model for collaboration is called for to achieve these results. Researchers, for example, have to focus efforts on developing practical solutions to real-world problems. From stakeholders, greater transparency is needed to ensure that researchers (and funding agencies) can more effectively identify and address research gaps, rather than letting known challenges on the grid continue to persist. Finally, both researchers and stakeholders must engage in dialogue and experimentation—sharing results and providing feedback early and often—to ensure that collaborations create effective and useful tools.

5 CONCLUSION

Data-driven tools will play an increasingly prominent role in grid operations and planning, as physical properties and dynamics of the grid evolve in the face of new technology adoption. The advancement of such tools will hinge on deploying instrumentation to collect faster and more localized measurements, capitalizing on state-of-the-art software solutions to facilitate big-data workflows, and enabling open exchange of data and information with research collaborators. A new information ecosystem for rapid iteration and learning will help keep the lights on as we decarbonize the energy sector on an aggressive schedule, and deal with the climate change effects to which we are already irrevocably committed.

ACKNOWLEDGMENTS

The work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001104, A National Infrastructure for Artificial Intelligence on the Grid (ni4ai.org).

REFERENCES

Public review for

Characteristic Profile: Improved Solar Power Forecasting Using Seasonality Models

Julian de Hoog, Maneesha Perera, Peter Ilfrich, Saman Halgamuge

This paper studies the problem of solar generation forecasting. The paper first characterizes the profile of daily solar power generation patterns and then adopts a seasonality model, such as the SARIMAX, to help to remove the seasonality and forecast the residual. In particular, the paper evaluates two recent seasonality models (including an improved model based on one of them), and introduces a more effective data-driven method called “characteristic profile”. Simulation results based on the proposed model show significant improvement over those based on the other two existing seasonality models, especially on some unique characteristics including sudden ramp down, curtailment, and shading.

All reviewers find the paper interesting and well-written; they also give constructive comments and suggestions. In particular, a suggestion for further investigation is that, since a key innovation of the work is the characteristic profile identification, it may help the community to fully understand its potential if some benchmarking evaluation can be carried out with respect to the state-of-the-art solar generation forecasting schemes.

Public review written by

Minghua Chen

City University, Hong Kong
Characteristic Profile: Improved Solar Power Forecasting Using Seasonality Models

JULIAN DE HOOG, The University of Melbourne, Australia
MANEESHA PERERA, The University of Melbourne, Australia
PETER ILFRICH, Melbourne, Australia
SAMAN HALGAMUGE, The University of Melbourne, Australia

The rapid uptake of rooftop solar photovoltaic systems is introducing many challenges in the management of distribution networks, energy markets, and energy storage systems. Many of these problems can be alleviated with accurate short term solar power forecasts. However, forecasting the power output of distributed rooftop solar PV systems can be challenging, since many complex local factors can affect solar output. A common approach when forecasting such systems is to extract the daily seasonality from the time series using some form of seasonality model, and then forecast only the residuals that remain after seasonality extraction. In this work, we explore in detail the effectiveness of three commonly used seasonality models, and we propose a new one, called the "characteristic profile". We find that when seasonality models are integrated into the forecasting process, significant gains in forecast accuracy may be obtained – particularly for machine learning based approaches, which have a reduction in forecast error of 5-25%. Among the seasonality models, the characteristic profile demonstrates the highest forecast accuracy, resulting in reductions in forecast error of 8% or more compared to forecasting models that do not take seasonality into account. The benefits of this approach are particularly pronounced when forecasting solar PV systems that are curtailed, suffer from local shading, or consist of multiple sets of panels having different orientations and tilts. Our results are demonstrated on a high resolution dataset obtained from 258 sites in Western Australia over the course of a full year.

1 INTRODUCTION

Installation of distributed solar photovoltaic (PV) systems on the rooftops of homes, commercial buildings and industry has been growing close to exponentially [28]. In Australia, more than one in five homes now have solar PV systems on their rooftops, and many other markets are expected to follow a similar trajectory [9]. In some neighbourhoods, more than 50% of homes now have solar PV, and on some days, entire distribution zones are beginning to generate more than they consume [48].

Such high levels of rooftop solar PV introduce many challenges. For system operators, high levels of solar PV can introduce short, steep ramps, oversupply risk, decreased frequency response, voltage rise and reverse power flow [14, 38, 48]. Market operators, who must accurately forecast net demand at least one day ahead, are finding it increasingly important to integrate the impacts of distributed solar PV into their forecasts [39]. For operators of energy storage systems, short-term solar power forecasts are essential for optimal system operation [1, 3].

As a result there is a strong need to better understand, model, and forecast the generation output of distributed solar PV systems – both for individual systems, and for the aggregated generation of distributed solar PV across entire regions.

To model and forecast distributed solar PV, it is necessary to understand all the many factors that affect a system’s output. These can be broadly categorised as shown in Table 1.

There has been significant effort over many decades to forecast the solar generation of large-scale systems, such as solar farms. These are typically carefully planned and designed, with panels installed at an optimal tilt and orientation, and sites being carefully maintained to avoid any shading or other negative impacts on generation. The process to model and forecast such systems therefore usually focuses primarily on the temporal, geographic, and environmental factors that affect system output, with most of the emphasis on modelling and forecasting solar irradiation.

However, for small to medium scale systems (such as residential, commercial or industrial solar PV), many additional system-specific and location-specific factors must be considered. Such systems are...
usually installed on existing rooftops, which may not have an optimal tilt and orientation, or may have multiple sets of panels with different tilts and orientations. They may be subject to local shading effects, due to nearby buildings, trees, or rooftop features. They may be exposed to local distribution network limitations, such as inverter self-curtailment. They may suffer from losses due to inverter or wiring. Some of these effects can be seasonal or can change over time (such as trees growing, or new structures being built nearby).

Attempting to accurately model such systems from first principles requires detailed knowledge of many different factors. The process for just a single system can be time-consuming and prone to errors. Developing detailed models of hundreds or thousands of individual systems across entire regions is simply infeasible.

However, for an increasingly large proportion of systems, time series data of power output is now being recorded and made available, either through improved advanced metering infrastructure, or through logging and monitoring functionality provided by installers, hardware suppliers, or third parties. The system-specific and location-specific factors affecting each system are inherently represented in such time series data, and can subsequently be integrated into the forecasting process without requiring detailed prior knowledge of all the system’s parameters and specifications.

When forecasting solar power output using such time series data, it can be helpful to “de-season” the time series – in other words, to remove the daily seasonality from the time series using some form of seasonality model, and to focus instead on the residual time series. For distributed, small-scale solar PV systems, the system-specific and location-specific factors form part of the daily seasonality of that system’s time series. If that daily seasonality can be accurately detected and removed from the time series, then only the environmental factors need to be forecasted, and this can be done without noise being introduced into the time series by the system-specific and location-specific factors.

In this paper we explore in detail four seasonality models for solar PV time series data. We implement three well-known and commonly used approaches and discuss their respective advantages and disadvantages. The first is a commonly used “clear sky model” that does not require any time series data but rather estimates solar output for a given time and location by taking into account solar elevation angle, site altitude, and atmospheric conditions [26, 31]. The second, described by Boland [12] and using insights from earlier work by Dong et al. [21], uses Fourier decomposition to express the solar seasonality as a sum of periodic functions. The third, inspired by a set of papers by Chen, Irwin and colleagues [6, 16–18], uses a “physical black box” model that fits three parameters (tilt, orientation, and system conversion efficiency) to historical time series in order to learn the most likely values for these system parameters. For this model we further develop a gradient descent-based approach that makes it possible to detect these parameters faster than the binary search originally proposed by the authors. The fourth, which we call “characteristic profile”, is our own seasonality model and one of the main contributions of this paper. It uses a simple filtering approach on recent generation data to learn a unique profile of the system under consideration.

We test and demonstrate all four models on a large and high-resolution dataset collected in Western Australia (comprising more than 12 months of 15-minute data for 258 homes). We show that all four seasonality models can lead to significant forecasting improvements for multiple commonly-used forecasting models. The characteristic profile-based approach consistently performs best of all, and can lead to forecasting improvements of 8% or more when compared to forecasting models that do not take seasonality into account.

The key contributions of this paper can be summarised as follows:

- We illustrate in detail multiple features of distributed solar PV time series data that are difficult for existing forecasting approaches to take into account.
- We implement and compare three existing seasonality models and extend one of these (the “physical black box” model) with a gradient descent-based approach that enables faster identification of system parameters.
- We introduce a new seasonality model, the “characteristic profile”.
- We evaluate and compare all four seasonality models across a large dataset, and show the forecast accuracy improvement that can be obtained with each of them.

2 RELATED WORK

Several extensive reviews of existing work in solar power modelling and forecasting have been published [20, 27]; here we provide only a brief overview of key approaches that are most relevant to this work. To guide our literature review we use a categorised list of the main factors affecting solar PV generation, as shown in Table 1.

**Temporal and geographic factors** are deterministic and well understood. They are often addressed using a “clear-sky model”, which estimates the amount of solar irradiation a specific location is exposed to at a specific time under cloudless conditions [4, 8, 22, 26]. Often, clear sky models form a starting point for more sophisticated approaches that seek to further integrate environmental, system-specific, and location-specific factors. When used for solar irradiation forecasting, they can be thought of as seasonality models themselves. When used for solar power forecasting, they may be integrated into seasonality models that additionally take into account the conversion of irradiation into power.

The way in which environmental factors are taken into account depends on the time scales involved and on the availability of sensors and data. Over longer time scales, numerical weather prediction models are often used [34]. Like standard weather forecasting methods, such approaches attempt to model the physical processes that determine the evolution of atmospheric variables. These approaches are valuable for forecasting a system’s generation over several days, months or years, but are typically not accurate enough for short-term applications. At shorter time scales, environmental factors are often taken into account by using satellite imagery [5, 32, 40], or on-site sky imaging systems [33]. The use of satellite imagery can involve generating cloud motion vectors, or more recently, the use of deep learning for forecasting solar irradiation in future intervals [5]. On-site sky imaging systems involve the use of local sensors such as sky-facing cameras to predict cloud movement in the very short term. For time scales of up to 30 minutes,
these can be very effective, but they must be used in conjunction with other forecasting methods for longer horizons.

**System-specific and location-specific factors** are treated differently depending on the application. For large scale solar PV generation systems, they may be addressed via the use of physical modelling. For medium- to small-scale systems, however, it may be infeasible or uneconomic to model the system to this level of detail. Furthermore, physical models may become less accurate over long time horizons, and recalibrating them may be impractical. In such cases, statistical (i.e. data-driven) approaches to forecasting are more common.

The data used in statistical approaches typically involves at minimum a historical log of solar generation, but may also include historical logs of other relevant values such as temperature, wind speed, pressure, humidity, etc. Approaches that attempt to express solar power as a function of such exogenous variables include neural networks [29, 45], support vector regression [50], or genetic algorithms [36]. Recently, recurrent neural networks have been shown to be effective at integrating the correlation between solar generation in consecutive intervals [37, 41].

Often, solar generation time series are adjusted such that the seasonality is removed ('de-seasoning'), as shown in the example in Fig. 1. One way to do this is to normalise the time series with respect to a clear-sky model or a clearness index ('multiplicative de-seasoning') [4, 22, 44]; another is to apply Fourier decomposition or use wavelets ('additive de-seasoning') [10–12, 21]. This makes it possible to represent the remaining stochasticity of the solar generation time series as a stochastic process without seasonality, and well-known methods exist for handling these.

When the stochastic process is stationary, auto-regressive (AR) or moving average (MA) models may be applied [10, 13]. These can also be combined into an ARIMA process [42]. When exogenous inputs are included, these become ARIMAX processes; and when seasonality is further taken into account, SARIMA(X) models may be used [30]. In some studies, ARIMA-based models have been shown to be more accurate than competing approaches [42, 43].

Hybrid approaches may use seasonality models to extract the residual time series, and then another approach to forecast the residuals. For example, wavelets have been used in combination with neural networks [15, 35]. An auto-regressive approach combined with a dynamical model of the system shows advantages over other systems [23]. Other interesting approaches involve learning and using the cross-correlation of (sometimes lagged) generation across multiple sites to increase the forecasting accuracy [11, 22].

Extensive comparison of solar generation forecasting approaches can be difficult due to differences in time scales, solar variability of a given site, and error metrics used [51]. In one study on short term solar irradiance forecasting models, it was shown that persistence and regression-based models perform better at 15-minute horizons, ARIMA models perform better at 1-hour horizons, and frequency-domain models perform better at 2-hour horizons [43].

The precise impact of seasonality models is also not that clear. While many of the discussed approaches integrate seasonality models into the forecasting process in one way or another, to the best of our knowledge there has not been a detailed study of the value that such seasonality models actually add. In this paper we intend to shed some light on the value of such models, particularly for small-scale distributed solar PV systems.

### 3 DISTRIBUTED SOLAR PV TIME SERIES DATA

In this section, we describe the dataset used for this study and illustrate in detail the challenges that arise when forecasting distributed solar PV time series data.

The data we use throughout this study consists of 258 individual time series collected from standalone solar PV systems on the rooftops of homes and businesses in Western Australia. The data was collected from 12-Feb-2020 to 10-Mar-2021 and has a resolution of 15 minutes. For each solar PV system we know its rated capacity and the postcode that it is located in, which allows us to approximate its exact location to within a couple of kilometres.

In exploring this dataset, it became clear that a significant number of these time series exhibit characteristics that can be challenging for many seasonality models and forecasting approaches.

**Curtailment**: In much of Western Australia, regulations stipulate that system output may not exceed 5 kW (similar regulations exist in many other jurisdictions around the world). Despite this, many home owners install systems that have a capacity exceeding this value, since the benefits of additional generation during the shoulder periods outweigh the loss of some generation being curtailed during peak hours. It is instructive that the most commonly installed system capacity in our dataset is in fact 6.6kW, despite the 5kW export limit; presumably this represents an optimal cost-benefit trade-off. The result of this, however, is that a large number of systems experience some level of curtailment, particularly during summer months. An example of this is Site 3 in Fig. 2 (and to a lesser extent, Site 2). During the summer, this system can generate more than 5kW from 10am to 3pm, but due to regulations, generation is capped, leading to a flat peak.

We conducted a manual review of all 258 systems and found that 130 displayed some form of curtailment during at least part of the year – in other words, approximately half of the systems are curtailed in this way. This can be difficult for seasonality models to detect and accurately model, as we discuss in the next section. However, when such a large proportion of systems exhibit this behaviour, this can impact not just the forecasts of individual systems, but also regional forecasts that attempt to take into account the full cohort of systems when producing a regional forecast.

**Shading in morning or evening**: Many systems have a large, sometimes instant, ramp up or down in the morning or evening.
This can be due to shading from adjacent parts of the roof or nearby buildings causing a sudden rise or fall in generation. Site 2 in Fig. 2 is an example of this, with a steep ramp down in the afternoon, particularly in summer. Again, such impacts can be difficult for many seasonality models to detect and handle.

Multiple sets of panels: Since solar PV panels are usually installed on existing rooftops, and many rooftops have multiple angles and faces, distributed solar PV systems often consist of multiple sets of panels having different tilts and orientations. Each set will have a unique contribution to total system output that can change throughout the year and have its own particular seasonal characteristics. Seasonality models that attempt to estimate a system’s seasonality as a whole typically struggle to handle such impacts.

Other unique impacts: There can be many other unique impacts that don’t fall into one of the categories above. Consider for example Site 4 in Fig. 2. The significant reduction in generation between 11:30 and 15:15 may be due to shading from a nearby structure whose shadow passes over the solar PV system in the middle of the day (perhaps from some overhanging part of the roof). This impact is only felt in summer, emphasising the need for seasonality models that can adapt to different impacts at different times of the year.

4 SEASONALITY MODELS

In this section we describe four solar seasonality models, and compare their performance on the dataset described in Section 3. These models were deliberately chosen: the first (“Clear Sky Model”) is the only model that is not data-driven (it does not require pre-existing time series data from the site of interest), and we include it for benchmarking purposes. The second (“Fourier Series”) does not attempt to directly model any specific physical parameters within the time series, but rather uses awareness of frequencies commonly present in Fourier analysis of solar irradiation data to approximate the time series seasonality. It represents an attempt to apply a model considered among the best for modelling irradiation seasonality to the domain of solar power seasonality. The third (“Physical Black Box Model”) estimates physical characteristics of the system and subsequently determine seasonality using well known equations that relate clear sky irradiance, solar position, and system parameters to solar power generation. We consider it the most promising among existing data-driven methods to handle the unique features of distributed solar PV systems described in Section 3. The fourth (“Characteristic Profile”) is our own contribution, and aims to address in particular the many unique impacts that can be present in small-scale distributed solar PV data (as discussed in Section 3).

4.1 Clear Sky Model

This is an example of this, with a steep ramp down in the afternoon, particularly in summer. Again, such impacts can be difficult for many seasonality models to detect and handle.

Multiple sets of panels: Since solar PV panels are usually installed on existing rooftops, and many rooftops have multiple angles and faces, distributed solar PV systems often consist of multiple sets of panels having different tilts and orientations. Each set will have a unique contribution to total system output that can change throughout the year and have its own particular seasonal characteristics. Seasonality models that attempt to estimate a system’s seasonality as a whole typically struggle to handle such impacts.

Other unique impacts: There can be many other unique impacts that don’t fall into one of the categories above. Consider for example Site 4 in Fig. 2. The significant reduction in generation between 11:30 and 15:15 may be due to shading from a nearby structure whose shadow passes over the solar PV system in the middle of the day (perhaps from some overhanging part of the roof). This impact is only felt in summer, emphasising the need for seasonality models that can adapt to different impacts at different times of the year.

4.1 Clear Sky Model

Clear sky models are widely used in the solar forecasting literature and we include a clear sky model here for benchmarking purposes. These models do not require pre-existing time series data, but rather estimate solar irradiation for a given location and time by taking into account solar elevation angle, site altitude, aerosol concentration, water vapour, and various atmospheric conditions [44]. A large number of well-known clear sky models exist [4, 8, 22, 26], and these are well understood, easy to calculate, and widely used [27].

We include here a model originally developed by Ineichen and Perez [26], which is made available via the open source pvlib python library [31]. To convert solar irradiation estimates for each location into solar power estimates for each solar PV system, we multiply solar irradiation by the system’s rated capacity and an efficiency factor. This is a crude way to convert irradiation into power, but for benchmarking purposes it is helpful to include such an approach which (other than system capacity) does not assume availability of any other data.

4.2 Fourier Series

Solar PV power generation clearly has at least two frequencies: the daily cycle, and the annual cycle. This has inspired several researchers to explore the application of Fourier analysis to solar irradiation time series data. However, analysis of extended solar PV datasets using Fourier series has shown that it is also necessary to include the twice-daily cycle, as well as so-called “beat frequencies” adjacent to the daily and twice-daily cycles [12, 21]. The reason for including the twice-daily cycle is that “as well as night being different from day, morning is different from afternoon” [12]. The reason for including the beat frequencies is that these modulate the daily and twice-daily cycle to help obtain a better approximation of solar seasonality (although this is less important for locations close
to the equator, see [12]). Analysis of the dataset used in this study confirms the findings of [12, 21], as shown in the power spectrum in Fig. 3. There are clear spikes (and beat frequencies) at both 365 and 730. The model proposed in [12] thus consists of the following terms:

\[ P_I = a_0 + a_1 \cos \frac{2\pi t}{T} + b_1 \sin \frac{2\pi t}{T} + \sum_{n=1}^{N} \sum_{m=1}^{M} \left( a_{56n+m} \cos \frac{2\pi (365n + m)t}{T} + b_{56n+m} \sin \frac{2\pi (365n + m)t}{T} \right) \]  

(1)

where the \( a \) and \( b \) coefficients are determined for their respective frequencies using the discrete Fourier transform of the original time series. \( T \) is the number of intervals in a year, in this case 35,040 (for 15-minute resolution data).

This may be applied to an existing solar PV time series in a fairly straightforward manner, by calculating the Fourier transform on any available historical data and filtering out all frequencies except those in Equation 1 above. Some examples for solar seasonality profiles calculated in this way are provided in Fig. 4. Note that in this Figure, all values for the Fourier series profile that are before sunrise or after sunset have been set to zero. (The Fourier series seasonality model often has small positive or negative values during nighttime hours, but we consider it reasonable to post-process the data in this way and set these intervals to zero, since identifying them is trivial).

As can be seen, the Fourier seasonality model can approximate solar PV generation of a given system reasonably well when the solar PV system has a fairly “standard profile” (such as Site 1). However, even then it does not get the shape quite right, with slight overestimation in the morning and evening, and underestimation at peak. The Fourier series model also struggles to account for non-smooth features, such as sudden ramp down (Site 2), curtailment (Site 3), or shading (Site 4).

4.3 Physical Black Box Model

The “physical black box” solar performance model – inspired by extensive work by Chen and colleagues in [16–18], and made available as an open-source toolkit [6] – uses physical model parameters to build a seasonality model (called “maximum generation profile” in [6]). However, these physical model parameters do not need to be manually specified, but rather are learned from a small amount of historical data by finding parameter values that best fit the data. Assuming that a solar PV system’s latitude and longitude are known, its power at any given time \( t \) of the year can be determined by using the standard equation

\[ P_I = I_r \times k \times [\cos \Psi_I \sin \beta \cos(\phi - \alpha_I) + \sin \Psi_I \cos \beta] \]  

(2)

where \( I_r \) is the solar irradiance incident on the module, \( k \) is a system-specific parameter that estimates the conversion of solar irradiation into power (which is a combination of module size and conversion efficiency), \( \Psi \) is the sun’s elevation, \( \beta \) is the system’s tilt, \( \phi \) is its orientation, and \( \alpha \) is the sun’s azimuth. \( I_r \) can be determined using a clear sky model that provides the global horizontal irradiance (many open source libraries exist for this, and are considered fairly accurate). \( \Psi \) and \( \alpha \) are deterministic and can be calculated in a straightforward manner (again, open source libraries provide these).

The parameters specific to the system, therefore, are the parameters \( \beta \) (tilt), \( \phi \) (orientation), and \( k \) (conversion efficiency factor). Chen et al. proposed to find these by conducting an iterative binary search that first chooses \( k \), and subsequently determines \( \beta \) and \( \phi \) that best fit the model to historical data [18]. “Best fit” is determined by the set of parameters that minimise the root mean square error (RMSE) between the model and any of the days in the historical data. In other words, the most sunny day in the historical trace is typically the one that defines which \( \beta, \phi \), and \( k \) should be used to determine the seasonality of the time series.

While \( \beta \) and \( \phi \) should remain constant throughout the year, \( k \) incorporates additional factors that affect the conversion of irradiation into power. As the authors themselves note, \( k \) may be significantly different from one part of the year to another. As a result, to ensure that an accurate seasonality model is obtained, \( k \) should be re-determined periodically throughout the year using the method described above and in [16].

There are two minor drawbacks to the physical black box model. The first (as also acknowledged by the authors in [18]) is that there is a small chance that the parameter fitting approach may choose a local minimum. For example, if there is a day in the historical trace that is very cloudy all day long, the algorithm may find that a very small \( k \) actually determines the season profile having lowest RMSE, leading to an incorrect (near-zero) seasonality model. However, this drawback can be easily circumvented by ensuring that only those days with a maximum generation value that is within some threshold of the full historical trace’s maximum may be considered.

Fig. 4. Comparison of seasonality models for 21-Dec-2021 (a sunny summer day) – full profile (bottom) and zoomed in view of profile peak (top).
We now describe a simple method for determining solar seasonality. A plot of the time series of a distributed solar PV system usually shows a clearly discernible daily generation profile. On sunny days, the shape of this profile is immediately evident, but in periods having no fully sunny days, it may be less obvious. However, when multiple consecutive days are plotted together (as shown in left column of Fig. 5), typically the unique shape becomes clearly evident regardless of whether there have been fully sunny days in the recent past or not. This is due to the fact that the probability of having had a sunny period in a given interval on at least one of the prior days is high. For example, one day may have been sunny in the morning but cloudy in the afternoon, and another may have been cloudy in the morning but sunny in the afternoon. Taken together, they can be used to identify a system-specific “full sun” profile for a full day.

An outline of our method for generating a characteristic profile is shown in Alg. 1. Our input is a dataset consisting of historical PV generation measurements over $D$ recent days, having $T$ intervals in each day. Some examples of measured daily solar PV generation profiles are shown for the four sites in the first column of Fig. 5.

The first step involves determining for each interval of the day the maximum value measured in that interval across all recent days. The outcome of this first stage is $M$, a profile of interval maximum values, as shown in the middle column of Fig. 5.

For high resolution data, this ‘interval maximum’ profile often still has spikes, outliers, and errors that distort the shape of the generation profile. This can be due to many factors, for example: errors in measurement or faulty data; low probability events such as a particular interval having been cloudy on all days in the historical data record; or one-off spikes due to natural phenomena such as refraction of sunlight at cloud edges [47]. As a result, the second step involves removing outliers and smoothing or filtering the profile $M$. Our aim is to determine a characteristic profile that represents the general shape of a system’s generation well, as shown in the right column of Fig. 5. Any standard data filtering approach may be used in principle, and we explored the use of median, moving average, Butterworth, Kolgomorov-Zurbenko and Savitzky-Golay filters. In our experience the performances of each of these filters were very similar, and for the remainder of this study we applied a simple median filter due to its ease of computation and interpretability.

4.4 Characteristic Profile

We now describe a simple method for determining solar seasonality that is one of the main contributions of this paper. An outline of this method was originally proposed in [19]; in this paper we provide a more thorough description, an extensive analysis, and a detailed comparison to existing related work.

A plot of the time series of a distributed solar PV system usually shows a clearly discernible daily generation profile. On sunny days, the probability of having had a sunny period in a given interval on at least one of the prior days is high. For example, one day may have been sunny in the morning but cloudy in the afternoon, and another may have been cloudy in the morning but sunny in the afternoon. Taken together, they can be used to identify a system-specific “full sun” profile for a full day.

An outline of our method for generating a characteristic profile is shown in Alg. 1. Our input is a dataset consisting of historical PV generation measurements over $D$ recent days, having $T$ intervals in each day. Some examples of measured daily solar PV generation profiles are shown for the four sites in the first column of Fig. 5.

The first step involves determining for each interval of the day the maximum value measured in that interval across all recent days. The outcome of this first stage is $M$, a profile of interval maximum values, as shown in the middle column of Fig. 5.

For high resolution data, this ‘interval maximum’ profile often still has spikes, outliers, and errors that distort the shape of the generation profile. This can be due to many factors, for example: errors in measurement or faulty data; low probability events such as a particular interval having been cloudy on all days in the historical data record; or one-off spikes due to natural phenomena such as refraction of sunlight at cloud edges [47]. As a result, the second step involves removing outliers and smoothing or filtering the profile $M$. Our aim is to determine a characteristic profile that represents the general shape of a system’s generation well, as shown in the right column of Fig. 5. Any standard data filtering approach may be used in principle, and we explored the use of median, moving average, Butterworth, Kolgomorov-Zurbenko and Savitzky-Golay filters. In our experience the performances of each of these filters were very similar, and for the remainder of this study we applied a simple median filter due to its ease of computation and interpretability.
To conduct a preliminary evaluation of the accuracy of all three accuracy of each seasonality model, it does not provide a de
box model. Systems, achieving a 54% reduction in RMSE compared to the Clear
outperforms all three other methods, even more so for curtailed
results are shown in Fig. 6. The characteristic pro
le consistently
malised to the maximum actual power output of that day. The full
output measured at the site, averaged over the full day, and nor-
square error (nRMSE) between each model and the actual power

determined across the entire dataset). The date for each site was
the approximate 50-50 split of curtailed / non-curtailed that we
know experienced some level of curtailment, and 75 sites
le). We randomly selected 75 sites
le. Each of these represents one site on one fully sunny day (no impact
on any given day into a single pro
"copies and pastes", so to say, the most sunny values of each interval
any need for a fully sunny day in the historical time series; rather, it
other hand, can piece together a solar seasonality model without
parameters etc.) that may have an impact. In addition, this kind of metric does
not take into account the other benefits that certain seasonality
models may offer. For example, the Physical black box model has
a valuable ability to detect actual physical parameters, which may
be valuable for other purposes, and the Characteristic profile is not
able to provide this. A different measure of the value of a seasonality
model is to determine how much it improves solar PV forecasts, and
we discuss this in the next section.

5 VALUE OF SEASONALITY MODELS IN FORECASTING
To understand the value of seasonality models in forecasting, we
implemented four forecasting models that are commonly used in
forecasting studies. We analysed the forecasts obtained from these
models when trained with the original time series (without sea-
onality extraction) and when trained with residuals that remain
after de-seasoning by each of the seasonality models (Clear sky,
Fourier series, Physical black box and Characteristic profile). For
the seasonality models, the residual forecasts are added to the sea-
onality profile to obtain the final power generation forecasts. For
this analysis, we conducted a rolling-window [7] evaluation from
2020-03-31 to 2021-03-01, where for each evaluation window a day-
ahead forecast was made at a 15-minute resolution by retraining
the forecasting model(s) using the most recent one month of data.

Fig. 6. Comparison of RMSE for four seasonality models across 150 sites
(75 curtailed, 75 not curtailed), on fully sunny days (no cloud cover)
day may have other environmental factors (temperature, humidity,
etc.) that may have an impact. In addition, this kind of metric does
not take into account the other benefits that certain seasonality
models may offer. For example, the Physical black box model has
a valuable ability to detect actual physical parameters, which may
be valuable for other purposes, and the Characteristic profile is not
able to provide this. A different measure of the value of a seasonality
model is to determine how much it improves solar PV forecasts, and
we discuss this in the next section.

5 VALUE OF SEASONALITY MODELS IN FORECASTING
To understand the value of seasonality models in forecasting, we
implemented four forecasting models that are commonly used in
forecasting studies. We analysed the forecasts obtained from these
models when trained with the original time series (without sea-
onality extraction) and when trained with residuals that remain
after de-seasoning by each of the seasonality models (Clear sky,
Fourier series, Physical black box and Characteristic profile). For
the seasonality models, the residual forecasts are added to the sea-
onality profile to obtain the final power generation forecasts. For
this analysis, we conducted a rolling-window [7] evaluation from
2020-03-31 to 2021-03-01, where for each evaluation window a day-
ahead forecast was made at a 15-minute resolution by retraining
the forecasting model(s) using the most recent one month of data.
In addition to the historical power generation data, weather data
temperature, humidity, cloud cover, UV index, pressure, wind speed
dew point) collected from Dark Sky [46] were used as exoge-
nous features when forecasting. The weather data was collected for
30 locations across Western Australia that were specifically chosen
to be as close to the solar PV sites as possible. For each individual
site we use the weather data from the closest data collection point,
which in most cases is within a few kilometres.

5.1 Forecast models
Periodic Persistence: Periodic persistence models are naïve forecast-
ing models where the forecast of a future period has the same values
as the previous period (e.g. the power generation of the next day is
the same as the previous day). While this is clearly a very simple
approach for forecasting, it is often used as a baseline for comparing
more advanced forecasting methods [2].

(S)ARIMAX: As discussed in Section 2, autoregressive models can
take many variations based on the time series of interest. These mod-
els represent the forecast variable as a linear combination of the past
observations and forecast errors. We implemented a seasonal model
(SARIMAX) for the original time series as the seasonality needs to


case (1)
case (2)
case (3)
case (4)

ACM SIGENERGY Energy Informatics Review

102

Volume 1 Issue 1, November 2021
be accounted for, and non-seasonal models (ARIMAX) for the residual time series (where seasonality has already been removed). The weather time series are given as exogenous data for these models. The model parameters for the seasonal and non-seasonal ARIMAX models were tuned using the auto-arima approach [24].

**Multiple Linear Regression (MLR):** MLR is a simple machine learning algorithm where a linear relationship is modelled between the forecast variable of interest and exogenous variables. The MLR model was therefore trained with the weather time series of the forecasting day and the historical power generation data.

**Support Vector Regression (SVR):** SVR is a machine learning algorithm that is used for regression tasks where the model tries to find the optimal line (with a given set of exogenous features) that helps in predicting the variable of interest. The SVR model is trained similar to the MLR model to produce a day-ahead forecast when weather time series of the forecasting day is given as an input.

### 5.2 Metrics

Choosing the right metric to evaluate and compare solar forecasts is not straightforward. Standard metrics such as mean absolute percentage error (MAPE) and root mean square error (RMSE) can suffer from some drawbacks; good discussions of error metrics can be found in [27, 49, 51]. We use the mean absolute scaled error (MASE), first proposed in [25]. This metric is being increasingly used as a standard metric in the forecasting community, and does not suffer from the problems with zero values or scale dependency that other metrics do, while being robust to differences between different geographies.

Mean absolute scaled error (MASE) is defined as follows:

\[
\text{MASE} = \frac{\frac{1}{h} \sum_{t=n+1}^{n+h} |Y_t - F_t|}{\frac{1}{(n-m)} \sum_{t=m+1}^{n} |Y_t - Y_{t-m}|}
\]

The denominator represents the mean absolute error of the one-period naïve forecast model (also called periodic persistence, as above), evaluated on the training set (consisting of intervals 1 to \(n - 1\)), where \(m\) is the number of intervals in one period (the seasonality). The numerator represents the mean absolute error of all intervals in the forecast horizon \(h\) (consisting of intervals \(n\) to \(n + h\)), where \(Y_t\) is the actual value at time \(t\), and \(F_t\) is the forecast value at time \(t\).

### 5.3 Results

The summarised results, across the full dataset (258 time series), across the full year (forecasted one day at a time) are presented in the box plot in Fig. 7. The percentage of error reduction resulting from the use of each seasonality model is presented in Table 2.

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Clear Sky</th>
<th>Fourier Series</th>
<th>Physical Black Box</th>
<th>Characteristic Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periodic persistence</td>
<td>0.0</td>
<td>-0.9</td>
<td>-1.7</td>
<td>-0.2</td>
</tr>
<tr>
<td>(S)ARIMAX</td>
<td>-5.2</td>
<td>-0.8</td>
<td>-6.2</td>
<td>-0.2</td>
</tr>
<tr>
<td>MLR</td>
<td>11.5</td>
<td>14.6</td>
<td>15.1</td>
<td>21.5</td>
</tr>
<tr>
<td>SVR</td>
<td>6.5</td>
<td>9.2</td>
<td>5.6</td>
<td>12.7</td>
</tr>
</tbody>
</table>

In essence, this metric compares the generated forecast with the naïve one period-ahead forecast, making it scale-invariant, and avoiding the problems with zero values that e.g. MAPE can have.

**Best overall forecasting model:**

The best overall forecasting model uses the Characteristic profile to de-season the data, and then applies multiple linear regression to the residual time series, taking weather data into account. This forecasting model represents an 8% improvement in forecasting accuracy over the best forecasting model that does not use a specific solar seasonality model in the forecasting process (SARIMAX).

**Impact of different seasonality models:**

The best model using the Characteristic profile for seasonality extraction (MLR) improves forecast accuracy more than any forecasting approach using of the other seasonality models: it improves forecast accuracy by 6.9% compared with the best model using the Clear sky model (SVR); by 4.1% compared with the best model using the Fourier series (SVR); and by 7.5% compared with the best model using the Physical black box model (MLR).

**Impact of incorporating seasonality models into standard forecasting approaches:**

For the periodic persistence-based forecast, de-seasoning the time series makes almost no difference (as expected). Since the forecast for the next day is simply the same values as the previous day, these
results only reflect that some seasonality models change more from one day to the next than others.

For the statistical (S)ARIMAX-based forecasts, de-seasoning also has minimal impact. Seasonal ARIMA-based models already inherently address seasonality that exists in the time series. Adding a seasonality model to extract this seasonality is to some extent redundant and does not lead to forecast accuracy improvements.

However, for the two machine learning models, MLR and SVR, seasonality extraction makes a big difference. Unlike some statistical approaches (such as SARIMAX), these machine learning-based approaches do not inherently take seasonality into account. By extracting and removing the seasonality from the time series, these machine learning-based approaches can focus entirely on modelling the relationship between the exogenous variables (cloud cover, temperature, and so on) and solar power output. Machine learning-based approaches can learn this relationship better than statistical models such as SARIMAX, leading to much better overall forecast accuracy. We anticipate that this would be similarly true for even more advanced deep learning-based approaches, which are starting to receive more attention in the solar power forecasting community.

6 CONCLUSION

The increasing uptake of distributed solar PV is starting to introduce many challenges for system and market operators. Accurate short-term solar power forecasting can help to alleviate many of these impacts. While there exist many approaches for solar power forecasting, most of these do not take into account the unique characteristics that small-scale, distributed solar PV systems may have. Modelling such systems from first principles is generally not feasible, and as a result there is increasing interest in data-driven and so-called “black box” methods to model and forecast these systems. A common approach to data-driven solar power forecasting is to de-season the time series data, and subsequently focus instead on forecasting the residual time series after seasonality has been removed.

In this work, we implemented and compared three existing seasonality models – the first based on a Clear sky model, the second based on Fourier series, and the third based on a Physical black box model – and further introduced our own approach, the “Characteristic profile”. We evaluated these models on a large dataset of 15-minute resolution data collected from 258 sites in Western Australia over the course of a full year. It was found that all seasonality models are effective at improving forecast accuracy when compared with forecasting models that do not explicitly take daily seasonality into account – particularly for forecasting approaches that use some form of machine learning. The characteristic profile-based approach was shown to outperform the other seasonality models by 6.9%, 4.1% and 7.5%, respectively, and to lead to an overall forecasting accuracy improvement of at least 8% compared to the best forecasting model that did not use a solar seasonality model.

The characteristic profile is easy to implement using only a small amount (as little as one to two weeks) of historical data, and can therefore be a valuable tool for improved forecasting accuracy. More accurate short term solar forecasts will enable better management of distribution networks, energy markets, and energy storage systems, and will help to enable the ongoing transition to renewable energy sources.

ACKNOWLEDGMENTS

The authors are grateful to Solar Analytics for providing anonymised solar power generation data.

REFERENCES


A GRADIENT DESCENT TO FIND PHYSICAL BLACK BOX MODEL PARAMETERS

The physical black box model in Section 4.3, developed by Chen et al. in \[6, 16–18\], requires values for parameters \( V \), \( \phi \) and \( k \) when calculating power output using Equation 2. Here we describe a gradient descent-based approach to find these parameters.

For convenience we repeat here Equation 2:

\[
P_t = I_t \times k \times [\cos \Psi_t \sin \beta \cos(\phi - \alpha_t) + \sin \Psi_t \cos \beta]
\] (2)

We determine the partial functions:

\[
\frac{\partial P_t}{\partial \beta} = I_t \times k \times [\cos \Psi_t \cos \beta \cos(\phi - \alpha_t) - \sin \Psi_t \sin \beta]
\]

\[
\frac{\partial P_t}{\partial \phi} = I_t \times k \times [-\cos \Psi_t \sin \beta \sin(\phi - \alpha_t)]
\]

\[
\frac{\partial P_t}{\partial k} = I_t \times [\cos \Psi_t \sin \beta \cos(\phi - \alpha_t) + \sin \Psi_t \cos \beta]
\]

Let \( D = [d_t] \) represent the day in the historical time series data for which we have found the best fit (lowest RMSE) with current values of \( \beta \), \( \phi \) and \( k \), and let \( P = [p_t] \) represent the estimated power for this day calculated using Equation 2. We can think of \( d_t \) as the actuals and \( p_t \) as the predictions. We can represent the mean squared error as:

\[
E = \frac{1}{n} \sum_{i=0}^{n} (d_i - p_i)^2
\]

and therefore

\[
\frac{\partial E}{\partial \beta} = -2 \sum_{i=0}^{n} (d_i - p_i) \frac{\partial P_i}{\partial \beta}
\]

\[
\frac{\partial E}{\partial \phi} = -2 \sum_{i=0}^{n} (d_i - p_i) \frac{\partial P_i}{\partial \phi}
\]

\[
\frac{\partial E}{\partial k} = -2 \sum_{i=0}^{n} (d_i - p_i) \frac{\partial P_i}{\partial k}
\]

In each iteration the parameters can follow the standard gradient descent-based update using learning rate \( \ell \) (we use \( 1e - 6 \)):

\[
\beta = \beta - \ell \frac{\partial E}{\partial \beta}
\]

\[
\phi = \phi - \ell \frac{\partial E}{\partial \phi}
\]

\[
k = k - \ell \frac{\partial E}{\partial k}
\]

Like the authors in \[18\], we choose sensible starting values: \( \beta \) equal to the system’s latitude, \( \phi \) equal to 0° (or 180° in the northern hemisphere), and \( k \) equal to 0.4. Typically gradient descent converges within only a small number of intervals (10-20). When this is implemented in a rolling horizon manner (for example when parameters for each day in a time series are calculated based on a number of previous days), the previous day’s values for \( \beta \), \( \phi \), and \( k \) clearly form a good starting point.
The paper provides an elaborate review of concepts to generate models for demand side flexibility automatically. It focuses on machine learning techniques for model generation. The authors conclude that there are multiple options for the automated generation of models for demand side flexibility, so they highlight aspects to consider when developing a surrogate-based approach for the exploitation of flexibility. The statistical information that the surrogate models provide can be used to handle the model inaccuracy. Even though inaccuracies remain, automated surrogate model generation for demand side flexibility can simplify the integration of DERs in the opinion of the authors, as it makes the most labor-intensive integration step easier and thereby increases the amount of exploitable demand side flexibility. Such solutions are highly welcome for making demand side management easy to implement, which is one reason for accepting the paper for this journal, along with its high linguistic quality.

Public review written by
Anke Weidlich
University of Freiburg, Germany
Automated Generation of Models for Demand Side Flexibility Using Machine Learning: An Overview

KEVIN FÖRDERER, Karlsruhe Institute of Technology, Germany and FZI Research Center for Information Technology, Germany
VEIT HAGENMEYER, Karlsruhe Institute of Technology, Germany
HARTMUT SCHMECK, Karlsruhe Institute of Technology, Germany and FZI Research Center for Information Technology, Germany

Flexibility in consumption and production provided by distributed energy resources (DERs) is a key to the integration of renewable energy sources into the energy system. However, even for identical DERs, the flexibility can vary widely, based on local constraints and circumstances. Therefore, handcrafting models can be labor-intensive and automating the generation of models could help increasing the volume of controllable flexibility in smart grids. Depending on the underlying mechanism for controlling demand side flexibility, there are various ways how an automation can be achieved. In this paper, we discuss fundamental concepts relevant to the automated generation of models for demand side flexibility, give an overview of different approaches, and point out fundamental differences. The main focus lies on model generation by means of machine learning techniques.

CCS Concepts: • General and reference → Surveys and overviews; • Hardware → Smart grid; • Computing methodologies → Modeling methodologies.

Additional Key Words and Phrases: Demand Side Management; Machine Learning; Automated Model Generation

ACM Reference Format:

1 INTRODUCTION

Renewable energy sources (RES) play a key role in the decarbonization of the energy system. However, they also pose new challenges. One major challenge is balancing the supply and demand of electricity, caused by the fluctuating nature of solar and wind power [44]. Another challenge is the integration of this decentral generation into distribution grids, as the additional power feed-in can cause grid congestions and voltage violations [30, 32]. In order to tackle these issues economically, without simply installing more and more grid capacity wherever RES are installed, the former demand side must partake in the effort of balancing electricity supply and demand, as well as alleviating grid congestions and voltage violations.

Measures for influencing energy demand are generally covered under the term demand side management (DSM) and include procedures for direct and indirect control of distributed energy resources (DERs) [36], i.e., storages, generators, and controllable loads. Control is exercised by some demand side manager (DSMgr). Depending on the regulatory framework and potential contractual agreements, examples for DSMgrs include distribution system operators (e.g., [11]), virtual power plant operators, and regional or district automated energy management systems (EMSSs). In order to plan interventions and perform this control, a DSMgr needs to know their available courses of action, that is, models from which options to influence DERs can be derived. Furthermore, these models of demand side flexibility need to be increasingly detailed, the more accurate a DSMgr wants to plan and control. The demand side, on the other hand, is very heterogeneous, as individual providers of demand side flexibility operate different DERs and operate in different ways under varying operational constraints. Consider different kinds of production sites, office buildings, or residential homes for example. Crafting new models by hand or configuring existing models for each site is indeed possible, but labor intensive and needs at least some expert knowledge. If, instead, the required models could be generated or parameterized automatically, the acquisition of customized models for the available flexibility would be simplified, potentially leading to more overall usable demand side flexibility.

Before we further outline the prospects of automatically generating such models, let us first briefly discuss the term flexibility. Demand side flexibility, or simply “flexibility” in the context of smart grids generally either refers to the capability of influencing the operation of DERs (in the sense of “being flexible”) or a description of how DERs can be controlled (characterizing “how flexible” it is) [31, 37]. One such description is a set of feasible load schedules. A feasible load schedule is a schedule a DER or an ensemble of DERs can reproduce, while satisfying all operational constraints. Consider exemplary instances for DSMgrs include distribution system operators (e.g., [11]).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM SIGENERGY Energy Informatics Review

Volume 1 Issue 1, November 2021

© 2021 Association for Computing Machinery. XXXX-XXXX/21/11-ART $15.00
https://doi.org/10.1145/nnnnnn.nnnnnnn
by increasing demand and storing excess energy. Shortages, on the other hand, can be dampened by decreasing demand and releasing energy from storages. This coordinated control can be achieved or incentivized with different approaches, e.g., the aforementioned direct and indirect DSM measures. The automated generation of models could help in increasing the volume of controllable flexibility in a smart grid, especially when detailed descriptions are desired, as it would automate one of the most labor-intensive steps in integrating DERs into a coordination mechanism. Each model encodes the flexibility of a set of DERs and provides the necessary information for deriving operational choices. It may be generated by the associated flexibility provider themselves, i.e., the owner of the DERs, or some other, external party. The approach of using generated models is especially promising in combination with flexibility providers who employ similar methods. If, for instance, the local, automated EMS of a flexibility provider uses model-based reinforcement learning, it already has learned a model of the DERs’ dynamics, which could be passed on.

There are different options for the automated generation of models, for instance preparing a comprehensive model by hand and automating the parameterization, that is, using a grey-box approach. Grey-box models are combinations of white-box and black-box models [17]. While white-box models transparently describe the dynamics of a system, black-box models approximate the system output from a given input [17]. In other words, with a white-box model it is possible to analyze and explain in detail how the output is generated from the input. A black-box model, on the other hand, has an unknown inner behavior so that only inputs and outputs are known for certain. Such a black-box is depicted in Figure 1. Black-box models are statistical models and learned from data [17].

In this paper we view flexibility providers and their DERs from a black-box perspective and focus on machine learning, as this allows for great versatility. Using supervised learning, the mapping \( f \), i.e., the black-box model, is learned from given input and output data \((x, y)\). The goal is to find an \( f \) such that the model output \( f(x) \) approximates the true output \( y \) with minimal error, e.g., \( \|f(x) - y\| \).

Since the black-box models act as surrogates for the actual DERs or hand-crafted models, they are also referred to as surrogate models, in this paper.

With this paper, we aim to motivate and aid future research on the automated generation of models for the flexibility of DERs by means of machine learning techniques. There are various approaches for encoding flexibility in a learned model, i.e., many ways to generate such a model [16]. Based on our previous work and conducted review of general mechanisms for the exploitation of flexibility [16], we present and distinguish different modeling options, putting our previous results in a broader context and creating an overview of the different concepts and their associated use cases.

Furthermore, in this paper, we outline and illustrate fundamental differences between the approaches and point out important aspects to be considered when conceiving new methods for the automated generation of models for demand side flexibility.

Figure 2 outlines the fundamental questions discussed in this paper and shows how they are connected from a general perspective. Each question is associated to one step in the development of an approach for the automated generation of models for demand side flexibility and each answer influences the others. Starting point is the selection of an exploitation approach, which is closely tied to the specific use case and the specification of a control signal. The questions are separated into two groups: a group of more general considerations on the left and a group of considerations specific to an exploitation approach on the right. Interconnections between the individual questions of both groups do certainly exist and may be relevant in specific cases, but are not depicted. The paper’s structure is based on this distinction. In Section 2, related work is presented, outlining the current state of research in the field. Afterwards, in Section 3, the general questions are discussed: we show how a DSMgr can exploit demand side flexibility, illustrate the consequences of who generates the model and discuss the challenges of acquiring data and detecting flexibility in observed data. Section 4 is dedicated to use case specific consideration. It provides an overview of modeling approaches using machine learning models and discusses important aspects. Finally, the paper is concluded by Section 5.

2 RELATED WORK

The wish to automate model generation processes is not exclusive to energy related topics. "Automated model generation (AMG)” is investigated in, but not limited to, the context of integrated circuit verification and testing [26, 55]. The generated model, e.g., in form of differential (algebraic) equations, replicates input-output characteristics of complex circuits [55].

The generation of simulation models with "flexible structures", in the sense of constructing the model structure dynamically during the generation process, is investigated in [26]. The generation process...
involves transforming the given data into a graph, specifying how individual components from an existing “domain model component library” are coupled, model instantiation, and model calibration [26]. Similar approaches are conceivable for the modeling of demand side flexibility. However, in this paper we focus on the training of machine learning models, instead of the utilization of predefined components.

Automated model generation is also applied in energy related fields. One example is given by [35], where a generic space heating model representing an entire building as a single heating zone is presented. The model has a fixed structure. In order to generate an actual model for a building, the missing parameters are estimated from the given input data. In [9] models for evaluating the performance of heating, ventilation, and air-conditioning in buildings are generated from data. The input data is given according to a custom data model. From this data, a component-based system description in the form of a network of components is derived. Model generation for district heating systems is explored in [19] and [18]. The proposed generation process starts with the import of a given graph, e.g., OpenStreetMap data. Subsequently, a heating network is created based on the graph and parameterized. The generation process can be influenced with a set of configurations, like the number of connected buildings and probabilities for houses to be connected to the network [19]. Another example for an automatically generated thermal grid model is presented in [49].

Regarding DSM and flexibility, automated model generation is closely related to the detection and quantification of (potential) flexibility, as both derive information about flexibility from given data. Flexibility may be subdivided into different categories (see also [16]), such as energy flexibility and time flexibility [33, 34]. While energy flexibility means adaptability of power or running time, time flexibility means the possibility to change the starting time [33, 34]. In [33, 34], the (potential) energy and time flexibility of a building of a chemical factory are derived from load consumption time series by means of motif discovery. Motifs in this context means reoccurring patterns in time [33, 38]. The results are statistics for the length, power intensity, as well as the potential starting times. Nevertheless, detecting and learning about flexibility in recorded data is a challenging task, as discussed in Section 3.5.

Encoding the feasible set of DERs into surrogate models for the purpose of communicating flexibility was first proposed in [6]. The authors trained support vector data descriptions (SVDDs) to distinguish between feasible and infeasible load schedules. The basic idea of SVDD is to enclose a given dataset with the smallest possible sphere within a so-called feature space [53]. Given an SVDD model of the feasible set and an arbitrary load schedule, the schedule can be classified by checking whether it lies within the sphere or not. If it lies outside, the schedule is infeasible. Projecting infeasible schedules onto or into the sphere and calculating the pre-image, that is, a potential feasible schedule, within the data space is proposed in [7, 8] and thoroughly investigated in [3]. This SVDD-based “repair” mechanism has since been extended (e.g., [2, 43]). A similar approach by the same authors makes use of Chi-shapes instead, that is, polygons enclosing a dataset (similar to a convex hull) [4].

A classification-based approach to encode the feasible set is proposed in [41] and further investigated in [40, 42]. It makes use of a cascade of classifiers to determine whether a schedule is feasible or not. Each classifier evaluates a specific part of the load schedule and the schedule is deemed feasible if all classifiers label their part of the schedule as feasible [41]. Different types of classifiers may be used, including support vector machines, one-class support vector machines, and artificial neural networks (ANNs) [40, 41, 41].

Inspired by [6], we first proposed to model the flexibility of DERs by encoding it into ANNs in [13]. For this purpose, we conceived and outlined different “usage patterns”, which could be used by a DSMgr in order to derive control signals from a given ANN. The patterns are evaluated in [12] and based on these results another, similar pattern is proposed and investigated in [15]. Finally, a more systematic approach to the topic is presented in [16], where opportunities for the utilization of ANN-based surrogate models are identified. Additionally, a more thorough evaluation of the idea of ANN-based surrogate modeling is presented.

Data-driven models can also be found in the field of model predictive control. In [1] a review and case study of the ANN-based optimization of heating, ventilation, and air conditioning system is presented. The utilization of automatically generated hybrid automata is proposed in [50] and evaluated for a heat pump and a boiler. In contrast to these works, which generate models for the local, automated building energy management, in this paper, we focus on models used by some external entity in order exploit demand side flexibility.

3 EXPLOITATION OF DEMAND SIDE FLEXIBILITY WITH AUTOMATICALLY GENERATED MODELS

The first step towards the utilization of automatically generated models for the exploitation of flexibility, i.e., the direct or indirect control of DERs, is the selection of an existing or conception of a new exploitation approach. A categorization of different approaches found in the literature is presented in the following.

3.1 Approaches for the Exploitation of Demand Side Flexibility

A variety of different approaches for the exploitation of flexibility can be identified, using different characteristics [16, 28, 31, 37]. In this paper, we make use of the classification proposed in [16], which is based on [37]. In summary, five different categories of approaches for the exploitation of flexibility, also named “patterns”, are distinguished [16]:

Direct exploitation: DERs are directly controlled by some external entity. In order to be able to do so, the external entity uses a predefined interface and may even be able to request and collect information relevant to the DERs operation. As an example, consider an aggregator who controls individual DERs using custom interfaces for different DERs and flexibility providers.

Exploitation of abstracted flexibility: DERs are controlled with the help of highly abstracted models, meaning the models are not specific to a fixed DER type, i.e., a generator or a storage, but instead suitable to describe multiple types of DERs or an entire ensemble of DERs. One example from this category would be the curtailment of generation and
shedding of load with a reduction or on/off signal. Another example is the activation of automatic Frequency Restoration Reserves, which is implemented via a setpoints.

**Market-based exploitation**: How DERs are operated is decided by some market mechanism. Flexibility providers submit bids or requests to a market and receive the result after market clearing. Examples are energy markets and balancing energy markets.

**Indirect Exploitation**: Flexibility providers receive a signal incentivizing them to change their behavior and reschedule their DERs. Incentives are usually provided in the form of price signals. Dynamic tariffs are an example for an indirect exploitation approach.

**State information-based exploitation**: DERs are operated based on some state information observed or received by the flexibility provider. This exploitation pattern is usually employed in decentralized or distributed approaches. One exemplary application is the activation of the Frequency Containment Reserve, which is coupled to the observed power grid frequency.

The examples of Frequency Containment Reserve and automatic Frequency Restoration Reserve show that multiple patterns can be combined, as the monetary compensation for providing balancing power is determined with the help of an auction (market-based exploitation). Based on this classification, we conducted an extensive literature review, which is presented in [16], in order to identify common modeling approaches for the exploitation of demand side flexibility. It is important to note that the approaches are not limited to the automated generation of models. The result, which was compiled from a total of 173 analyzed publications, is depicted in Figure 3.

It shows the different classes of exploitation approaches (blue boxes), the associated modeling approaches within this class (white boxes), as well as different communication schemes, and relates them in terms of their abstractness. Abstractness (x-axis) was judged, based on the following two statements:

- “Abstraction is a process of generalization, removing restrictions, eliminating detail, removing inessential information (such as the algorithmic details).” [54]
- “Abstract specifications have 'more potential implementations', moving to a lower level means restricting the number of potential implementations.” [54]

On the y-axis, the approaches are grouped according to their genericity: from specific to a single DER or type of DER to generic approaches applicable to any DER.

Please note that Figure 3 does by no means capture every possible exploitation mechanism. Instead, it depicts and classifies the approaches commonly found in the literature. Moreover, it is possible to devise exploitation mechanisms that do not fit into this framework, simply by creating novel combinations of the depicted elements. For a detailed discussion of each element, we refer to [16]. A short summary of the approaches is provided in the following [16]:

**Classification of characteristics**: DERs are assigned characteristics, such as "shiftable", "curtailable", or "interruptible", and are modelled accordingly. Therefore, different DERs or types of DERs may share identical models.

**Parameterization of a common model**: A communication scheme in which flexibility providers describe their flexibility to an external entity by determining and sending the parameters of a predefined and commonly known model.

**Intervals**: Models are built around or from intervals, e.g., intervals for power, energy, and ramping capacity.

**Set of schedules**: DERs and their flexibility are described by a list of feasible load schedules. External entities simply choose the most beneficial entry.

**On/off**: DERs can be stopped and started.

**Imposed constraints**: External entities restrict the operation of DERs by setting operational constraints, e.g., a feed-in limit. Usually only one limitation is active at a time for either flow direction, that is, feed-in or consumption.

**Requests and offers**: A communication scheme in which flexibility providers explicitly offer flexibility, e.g., on a market, and/or external entities request the utilization of flexibility, for instance based on a contract.

**Data driven**: Flexibility is exploited in a data-driven way, for instance by learning a model and then using it to determine control signals. This is one of the key approaches for the automated generation of models.

**Elasticities**: Price-elasticities can be used to predict the reaction of a flexibility provider to price changes. With the help...
of these predictions, different price signals can be evaluated before selecting one.

**Tariffs:** Dynamic tariffs allow the adaptation of prices in order to influence the demand side and exploit its flexibility indirectly [37]. In the literature, the existence of dynamic tariffs is frequently assumed, but rather rarely the investigated subject.

**Stigmergy:** Stigmergy is a coordination approach making use of anonymous and only indirect communication via the manipulation of a shared environment [47]. In case of the Frequency Containment Reserve, the electrical grid is the shared environment and the observed information is the grid frequency. Each reserve provider manipulates the frequency by a tiny fraction via their load, but it is not possible for a reserve provider to tell what others have contributed.

**Exchange of individual state information:** A communication scheme in which flexibility providers share their state information, such as their planned load schedule. Other flexibility providers can use this information and adapt their own schedules in order to pursue a collective goal. In a distributed approach, flexibility providers are external entities to each other.

These categories and general approaches are vital for the definition of procedures for the automated generation of models for demand side flexibility. A selection of general concepts is presented in Section 4. For now, let us assume, we know the pattern the external entity uses to exploit flexibility, what kind of model should be generated, and which kind of data is required. The next step is then model generation, which poses the questions of who should generate the model, that is, the flexibility provider or the external entity, and from where the necessary data should be sourced.

### 3.2 External Entities Generate Models from Observed Data

In case models should be generated by a DSMgr or any other external entity, they must acquire the necessary data, before they are able to do so. Data can be acquired in different ways: firstly, it may be observed with the help of a (advanced) metering infrastructure. The infrastructure measures energy usage and can provide this information to the external entity, who stores the received individual measurements or load profiles. This data may, however, include inflexible power flows, which are not related to DERs. Secondly, the flexibility provider could periodically send the latest dynamic DER data to the external entity. Examples for dynamic data from DERs are state information, such as the state of charge of a storage or the operation status of a heat pump. The downside of this approach is that the flexibility provider is at all times completely transparent to the external entity. Lastly, sets of data may explicitly be requested from the flexibility provider. If the exact date of measuring the data does not matter in the intended exploitation approach, the data could be partitioned, the partitions shuffled, and date information removed, in order to obfuscate at least some behavioral patterns of the flexibility providers DERs.

The process of model generation by an external entity and subsequent flexibility exploitation is depicted in Figure 4. After acquiring a sufficient amount of data, using one or multiple of the options outlined above, the external entity generates the model. Which amount of data is sufficient, strongly depends on the employed machine learning model (see Section 4). The generated model is then stored for later use. Whenever DERs need to be controlled, that is, periodically or in case a specified event is detected, the model is loaded and used to derive control signals. Any type of signal, such as tariffs, load schedules, or operational constraints, could be derived given a suitable model (see Section 4). The signal, i.e., the "flexibility choice", is then sent to the EMS of the flexibility provider, which replans and controls the DERs accordingly. A feedback mechanism may optionally be implemented in order to detect discrepancies between model output and actual flexibility. Discrepancies arise since the models only approximate the real flexibility. Depending on the quality of the generated model, this approximation may be very rough. As an example, consider an EMS receiving a load schedule from an external entity. The EMS checks whether it can reproduce the schedule by means of rescheduling the local DERs and sends back the closest schedule it can achieve. The feedback can be collected to improve the respective model and may trigger a replanning process, in case the deviation is too large for the intended use.

### 3.3 Flexibility Providers Generate Models

Models may also be generated by the flexibility provider and sent to the external entity. A major advantage is the reduced amount of communicated data, as only the generated model needs to be transferred, instead of the training data. Hence, this decentralized model generation enables the utilization of additionally generated, artificial data in order to improve model accuracy. There are also privacy related implications: on the one hand, privacy is improved in comparison to the centralized model generation, as the training data is not directly available to the external entity. On the other hand, the communicated model may be exploited to draw conclusions about the dynamics and constraints of the DERs, and thus the behavior and habits of the flexibility provider.
The automated generation of models for flexibility is an especially promising concept if it is employed on multiple hierarchical levels and makes use of similar models on each level. As an example, consider a flexibility provider who optimizes the operation of DERs with the help of model-based reinforcement learning. This means that the flexibility provider has already trained a model of the DER dynamics (see [51]). Such a model could be passed to an external entity, that is, upward in the control hierarchy, and additional information or models may be provided to further facilitate signal generation.

A decentralized model generation can be implemented in two different ways: firstly, we can utilize reusable models, which only need to be generated whenever there are changes to the DERs and their operational constraints. For a model to be reusable, we must be able to pass state variables, that is, dynamic data determining the actual flexibility within the frame of theoretical flexibility given by the model and its (static) parameters. Take a battery energy storage system, for example. The amount of energy that can be charged or discharged is determined by technical properties and its current state, e.g., its nominal capacity and the state of charge. The nominal capacity is a parameter and can be communicated upfront, as it remains fixed, but the state of charge needs to be transmitted timely, whenever flexibility needs to be assessed.

Figure 5 depicts the utilization of reusable models. Models are generated by the EMS of the flexibility provider in such a way that they remain valid as long as the underlying parameters remain unchanged. As stated before, this can be achieved by generating models in which state information can be incorporated dynamically. Consider an ANN-based model which takes the state variables as an input, for instance. The resulting models are sent to the external entity and stored for later use. In case the DER dynamics change, e.g., when another DER is installed, the process is repeated.

Whenever flexibility is to be exploited, that is, when a given event is detected or periodically, the stored models are loaded and the flexibility providers are requested to send their current state information. Using this data, control signals are generated and sent to the flexibility providers’ EMSs, which in turn may provide feedback, analogous to Figure 4 and the external model generation.

If the generated model encodes the current state of the DERs, it is only valid in this exact state. In this sense, the model can only be used once, as the next time flexibility needs to be assessed, the DERs will likely be in another state. Therefore, new models need to be generated and communicated for each single exploitation attempt. An example for such a model is the set of feasible schedules encoded in an SVDD (see [3]). The complete exploitation process is depicted in Figure 6. Since all dynamic information is already encoded in the model, it does not need to be transmitted separately. Furthermore, there is no need to store the model, as every time a new one is generated.

In comparison, single-use and reusable models have different advantages and disadvantages: while a reusable model only needs to be generated once, it must be able to capture more complex dynamics, since it needs to be valid for a wide range of different states. If the model is learned from data, this means that training data from many different states is needed. As a consequence, the training process takes longer and needs more computational resources, since the relationship of state and flexibility must be captured. This challenge may also be reflected in model quality. In general, the reusable model is likely to perform worse in a given state than a single-use model trained to describe exactly this state (compare [16] and the results of [16] and [3]). However, since the reusable model is not immediately needed by the external entity, there is time for a thorough training. Vice versa, the generation of the single-use model must be fast (seconds to minutes), to be applicable on short notice. But, since only a description of the current flexibility is needed, simpler models and therefore fewer computational resources can suffice for generating each individual single-use model. As an example, compare again SVDDs generated from a few thousand samples [3].
and ANNs generated from hundred thousands of samples over the course of many training epochs [16].

3.4 Communication Channel
Receiving dynamic data from the EMS, sending control signals, and receiving feedback requires communication between EMS and the external entity. The external entity provides the necessary interfaces for the exchange of information. Technical limitations regarding the communication channel may be posed by the amount of data to be communicated, data transfer rates, reliability, and the mere existence of the communication infrastructure in remote areas. While the data may be communicated via the Internet, communication channels isolated from the Internet should be considered. The German Smart Meter Gateway infrastructure, for example, is able to provide a secure channel for such communication (e.g. [14, 29]).

3.5 Detecting Demand Side Flexibility in Data
If one were to handcraft custom models for individual flexibility providers, one would usually examine the associated DERs, inspect their environment, and talk to operators and users in order to determine constraints and record their preferences. Automating these steps requires collecting the same information from given data.

In the most basic case, only historic load profiles are given, from which we need to estimate parameters (grey-box approach) or learn a mode from scratch. Imagine we want to quantify how flexible an office building can be operated, solely by looking at observed load profiles. When there is feed-in and it follows a curve with a peak around noon, we can plausibly assume that there is a photovoltaic system present, but does the absence of such a feed-in curve mean that there is no photovoltaic system? The answer is: not necessarily. The installed generation capacity could simply be too small to outweigh the demands during office hours. There could also be a battery energy storage system, preventing feed-in and storing energy for later use. Only if there are days, in which we see the characteristic curve from solar generation in the load profile, we can conclude that there probably is a photovoltaic system. Then, using this information we can try to find evidence for batteries. This example illustrates the two core challenges of learning flexibility from load profiles alone:

1. In order to detect flexibility, it must show in the data. If DERs are always operated in the same way in a given situation, we cannot know for sure whether there are alternative operational choices or not.
2. Load profiles alone can only provide clues of the actual flexibility. Further assumptions, e.g., made by scanning for characteristic sequences, or information, for instance given by some master data register, are needed.

The first challenge is related to the exploration-exploitation dilemma found in reinforcement learning (compare [31]): an agent has to explore alternatives to the already known, well performing operational choices in order to discover new, potentially better options. Translated to the operation of DERs, this means that the normal, daily operation needs to be disturbed in some way from time to time, in order to force the selection of load schedules unseen before. The resulting load profiles then provide information about the available flexibility, when compared to periods without disturbances, especially if details about the disturbances are known. If, for instance, charging or discharging the previously considered battery energy storage system is restricted on some days, there will be days in which electricity generation from solar power is not concealed. Now, if we additionally know that charging was restricted, the difference in consumption gives us a clue about the flexibility. Please note that it is not necessary to know about the storage itself, it suffices to know that there was a restriction.

Knowingly causing disturbances in DER operation effectively generates additional information, which brings us to the second challenge: we need further assumptions or information to confidently draw conclusions about the available flexibility. For instance, one could try to disaggregate the collected load profiles or try to identify characteristic patterns (compare [33]) and conduct further analyses based on the results. Looking at the (imaginary) load profiles from our introductory example and reasoning that there likely is a photovoltaic system and a battery energy storage system equates to making assumptions. Based on these assumptions we can try to quantify the flexibility offered by the battery, but we cannot know for sure that the battery actually exists. In other words, estimates are generated on the basis of estimates. In order to make more confident statements, more definitive information is needed. Basic DER information could alternatively be given by hand or automatically collected from some master data register like the Marktstammdatenregister in Germany, an official register for power plants and battery energy storages.

Overall, the more information is available, the more certain and accurate conclusions can be drawn. Information cannot only be generated during model generation, but also when the model is used to control the DERs of a flexibility provider. Whenever a flexibility provider is not able to implement the schedule changes derived from the learned model, we gain new data, which we could use to improve our model. Additionally, since it is clear that the models will only provide estimates, the utilization of stochastic models should be considered. If, for instance, probability distributions or likelihoods are given, a DSMgr could compute confidence intervals when planning how DERs should be controlled.

3.6 Sources for Data
To generate a model from data, data is needed in the first place. Which data is required strongly depends on the intended use case (see Section 4). In case of the external model generation, data may be collected locally by the EMS and sent to the external entity periodically (“dynamic data” in Figure 4). From a general perspective, there are multiple possible ways to collect data:

**Observation during operation:** Automated EMSs manage energy flows in some optimized way. To do this, the EMS must observe sensor inputs and collect state information from DERs. In combination with the control signals issued by the EMS itself or the observed reaction of the DERs, a model of the DERs’ dynamics can be trained. However, the issue of determining flexibility from observed data, as discussed in Section 3.5, persists. Recording not only the DERs reaction to the optimized controls, but also feasible, alternative
options, is one possible solution to this problem. Such alternative options, for instance in the form of a set of feasible actions, could be provided by some future, standardized DER interface. Depending on the exact use case and modeling approach, different data is required.

**Querying from DER:** If such a standardized interface allows passing a presumed state as an argument, the necessary information may be collected directly from the DER independent of the true state and actual operation.

**Fake data points:** One option to introduce flexibility into the dataset is adding artificial data points. The creation of fake data for the sake of training better models is called dataset augmentation [22]. Depending on the modeling approach, there can be many different options for the generation of fake data. The major challenge is creating valid fake data, or else the learned model is at least partially invalid, that is, it does not represent the flexibility correctly.

**Extraction from existing models:** In order to control and optimize local energy flows, the automated EMS makes use of optimization algorithms and models. It would be possible to extract information on operational constraints, e.g., maximum and minimum power in a given state, from these existing models. If an approach for the automated generation of surrogate models was established, there could be a standardized interface for extracting all required information from optimization models.

Simulation models can be another source of data (e.g. [3, 12, 16]). We trained surrogate models from newly developed simulation models in our previous work [12, 16] to demonstrate the viability of the concept of ANN-based flexibility models. However, manually creating a simulation model solely for generating a surrogate model defies the purpose of surrogate modeling in this context.

4 DIFFERENT USE CASES AND MODELING OPTIONS

With the general considerations from the previous section in mind, let us now look at different options for generating descriptions of demand side flexibility. From Figure 3 in Section 3 we can derive three basic options:

**Generating a mathematical model:** An algorithm or artificial intelligence able to generate mathematical models (e.g., genetic and evolutionary programming [52], “evolvable mathematical models” [24], automatic generation of hybrid automata [50], or some regression algorithm) could be used to implement modeling approaches from the direct exploitation category in Figure 3. If the model is generated by the flexibility provider, the resulting mathematical model needs a common interface, that is, specific variables that are known, manipulated, or observed by the external entity. Without these, the external entity would not be able to make use of the model.

**Parameterizing a common model:** If the model structure is predefined and fixed, specific models are generated by estimating the missing parameters, which is a process that may be automated. This approach is mostly suitable for the abstracted flexibility category in Figure 3, where models with a higher degree of abstraction are used. There also may be multiple model structures to choose from, e.g., a battery, a bakery, and a bucket [45]. The selection of the best “template” could be achieved with the help of clustering methods.

**Generating a (data-driven) surrogate model:** For this, the goal is to generate a surrogate model, i.e., a black-box model, which in its inputs and outputs provides all the information needed to derive and select control signals. Control signals could be any signal, including load schedules, load deltas, and dynamic tariffs. Which signal is needed, how often a signal is sent, which time horizon it covers, and many other parameters depend on the exact exploitation approach. We refer again to Figure 3 for an overview of commonly found approaches and their associated signals. An overview of surrogate modeling approaches is provided in the next section.

Depending on the employed algorithms and whenever the model is generated step by step, e.g., evolution or gradient descent in the case of ANNs, the duration of model generation may be shortened for any of these options by starting from a pregenerated solution instead from scratch. Improving upon a prior solution can also help to improve model quality (see [22]), but may as well lead into a local optimum, potentially far away from the global optimum which would only be reached from a different starting point. In the context of ANNs, starting with pregenerated models is known as transfer learning and pretraining (see [22] for an introduction).

In the following we will focus on the third option for the generation of models, that is, the generation of surrogate models, which is usually achieved with the help of machine learning. With this option, demand side flexibility is encoded into a machine learning model, such as an ANN or support vector machine.

4.1 Surrogate Models for Demand Side Flexibility

Generally speaking, one could try to implement any approach depicted in Figure 3 with the help of surrogate models. Remember that from the black-box perspective the only thing we know about a given system is a set of system inputs and their associated system outputs. Hence, the main precondition is the possibility to derive adequate control signals from the considered model inputs and outputs, which in turn is only possible if the generated models are sufficiently accurate. Training accurate models requires an adequate amount of data, meaningful data (see also Section 3.5 on the issue of detecting flexibility), and a relationship between model inputs and outputs. Furthermore, the external entity using the model must be able to make sense of at least some of the data and know how the model needs to be used.

In [16], we derived and compiled a list of prospective surrogate modeling approaches for ANN-based surrogates by looking at the various types of models found in the literature and contemplating different input and output combinations. A generalization of this list, including many non-ANN examples, is given by Table 1. Table 1 is not exhaustive and many variations of the listed methods can exist. Furthermore, despite our best efforts, for some approaches no suitable examples could be identified in the literature, as the algorithmic generation and utilization of surrogate models as models...
Table 1. Outline of possible surrogate modeling approaches. The generated model provides predictions of the necessary information. Based on: [16].

<table>
<thead>
<tr>
<th>Signal</th>
<th>Method</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Repairing infeasible schedule</td>
<td>[3, 4, 8, 12, 43]</td>
</tr>
<tr>
<td>Schedule</td>
<td>Generation from abstract representation</td>
<td>[12]</td>
</tr>
<tr>
<td>Schedule</td>
<td>Prediction of state and action trajectory</td>
<td>[10, 15, 16]</td>
</tr>
<tr>
<td>Schedule</td>
<td>Classifying feasibility of schedule fragments</td>
<td>[40–42]</td>
</tr>
<tr>
<td>Schedule</td>
<td>Classifying feasibility of entire schedule</td>
<td>[6, 12]</td>
</tr>
<tr>
<td>Schedule</td>
<td>Prediction of costs</td>
<td>-</td>
</tr>
<tr>
<td>Tariff</td>
<td>Predicting resulting load (schedule)</td>
<td>[12], [27]*</td>
</tr>
<tr>
<td>Load delta</td>
<td>Predicting how long load is changed</td>
<td>[21]*</td>
</tr>
<tr>
<td>Constraint</td>
<td>Predicting how long constraint is satisfied</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>Predicting resulting load (schedule)</td>
<td>-</td>
</tr>
</tbody>
</table>

*related

for demand side flexibility is still a niche approach (compare [16]) and existing, similar models are often hand-crafted or used in other contexts (see for example [25], where load forecasting in general is investigated).

The "Signal" column lists the type of signal the external entity sends to the providers of flexibility. In order to control flexibility providers and their DERs via load schedules, one must be able to identify feasible load schedules within the generally much larger space of all load schedules. Since we define flexibility as the set of all feasible load schedules, any surrogate model that allows us to identify feasible schedules can be seen as a model for the flexibility of DERs [12, 13, 16]. Table 1 lists six potential methods for identifying feasible schedules. For the sake of brevity, we only outline the different methods for signal generation. More elaborate explanations and illustrations of the generation processes are provided in [16].

**Repairing infeasible schedule:** Given a mapping from infeasible load schedules to (close-by) feasible load schedules, one cannot only identify feasible load schedules by passing any random input, but also search for desirable schedules by systematically checking different inputs. This concept is used by the SVDD approach [3], for instance. It can easily be integrated into optimization heuristics, such as particle swarm optimization, by searching within the (unconstrained) space of all schedules and performing a schedule repair just before evaluating the target function value [3].

**Generation from abstract representation:** A surrogate model may provide a mapping from some abstract (potentially random) representation to a load schedule, similar to the mapping from genotype to phenotype in evolutionary algorithms. As an example for a generative model, consider a generative adversarial network (GAN) [23]. A GAN maps random input noise to some artificial data point, such as an image [23]. Since any data space is valid, one may train a GAN to generate feasible load schedules. Furthermore, GANs can be trained to consider additional input variables [39], e.g., dynamic state variables, which makes the model reusable. With such a GAN at hand, the external entity could pass (random) inputs until a desired load schedule has been returned or until a given time or computational budget is exceeded and choose the best generated schedule.

**Prediction of state and action trajectory:** With a model of the state space it is possible to predict how the states of a flexibility provider’s systems evolve. Starting from a given initial state, the external entity repeatedly determines a desirable power level (action) and the resulting subsequent states. Doing so, a schedule is generated power level by power level. Whether a power level (action) is feasible in a given state or not, can be detected with the help of an additional classifier [15, 16].

**Classifying feasibility of schedule fragments:** Classifiers for fragments of a load schedule are another option to identify feasible load schedules. A schedule is deemed feasible if every fragment is judged as feasible. One option to implement this method would be to train a list of classifiers, the first one judging only the first value, the second one judging the first two values, the third one judging the first three values, and so on. An external entity could then select value after value, always using the next classifier in line (see also [16]). Another option would be the utilization of the “cascade classifier” [40].

**Classifying feasibility of entire schedule:** A classifier able to distinguish feasible and infeasible load schedules allows us to identify feasible load schedules. However, randomly generating inputs most likely generates infeasible solutions, as the vast majority of possible schedules is infeasible. Additional information and systematic search methods are required to make use of this method. For ANN-based classifiers, for instance, one could try to use backpropagation to find input vectors, i.e., schedules, with a high likelihood of being feasible, that is, inputs yielding a high output value [16].

**Prediction of costs:** This method is closely related to the classification of an entire schedule. While the classifier rates whether a given schedule is likely feasible or not, with ratings from 0% to 100%, the cost prediction yields some value from the set of real numbers. In order to distinguish feasible and infeasible schedules, the model must be trained to penalize infeasibility with very high costs. By searching schedules with a cost below a given schedule, feasible schedules can be identified. As an added benefit, the estimated cost of implementing the schedule is known.

The length of the load schedules can be selected arbitrarily before model generation. Depending on the schedule generation method and the employed models, it may even be possible to change the length dynamically, as needed. One example for a method allowing dynamic schedule lengths is the prediction of state and action trajectory, in case the trajectory is put together one time step at a time. Then, the schedule generation can be stopped once the desired length has been reached. The length of the individual time steps, on the other hand, is generally fixed.

The remaining methods listed in Table 1 yield different types of signals and are not directly comparable. Furthermore, they are generally not suitable for the detailed control of DERs, as only...
incentives and operational boundaries are passed to the flexibility providers.

**Prediction of load (schedule) resulting from tariff:** A model predicting the resulting load schedule for a tariff or the resulting power level for a short term monetary incentive can be used to assess price signals. By comparing the expected reactions to price changes, the external entity can evaluate different tariffs and make more informed choices. However, depending on the modeled DERs, the changes between different tariffs may be rather small, which is one reason for the good results we achieved in [12]. In [27] the reward received by the external entity due to the changed schedule is predicted, instead of the load schedule itself.

**Predicting how long load is changed:** Short-term power changes can be evaluated by predicting if and how long the different changes can be implemented. For each flexibility provider, the desired change in Watts is passed to the surrogate model, which then provides the necessary predictions. Additional models or model outputs may be used to predict catch-up effects caused by each power change. In [21] a variation of this method is proposed, where the external entity computes how long a load can be interrupted.

**Predicting how long constraint is satisfied:** Limiting the (total) power drawn from or fed into the grid, for instance by issuing quotas of the maximum feed-in and consumption [11], is another possible approach to DSM. Here, the surrogate model predicts how long a given constraint can be satisfied by the flexibility provider. Again, additional models or model outputs could predict catch-up effects.

**Prediction of load (schedule) resulting from signal:** If flexibility is guided by the state of its environment, that is, flexibility providers observe the environment, interpret the state and react accordingly, one could try to manipulate this state in order to deceive flexibility providers. In this case, the surrogate model would predict the reaction to some defined, arbitrary signal. Please note that this case essentially is an abstraction of the tariff-based prediction. The different presented methods for signal generation may be implemented with various types of (machine learning) models and in different ways.

### 4.2 Considerations for the Utilization of Surrogate Models

Different machine learning models come with different advantages and disadvantages. For instance, we can easily create conditional models with ANNs, where the output depends on some additional input variables (e.g., [12, 39]). On the downside, ANNs are comparatively expensive to train, as a large amount of data is needed [22]. In this section, we outline fundamental considerations important to choosing a model type and implementing a signal generation method.

**Computational effort:** The computational effort required to generate/train and use a model can vary widely between different types of models. In the first step, a model must be generated either by the flexibility provider or the external entity. While some models can be generated rather fast, such as a simple regression model, others need extensive training, e.g., some deep ANN. Higher computational effort may be mitigated by the generation of reusable models. As explained before (see Section 3.3), if a surrogate model can process and make use of dynamic state information, the model only needs to be generated once, every time the DERs or other underlying dynamics or constraints change. Single-use models, on the contrary, must be generated every single time flexibility is needed and the model generation process must be sufficiently fast for the given use case. Hence, even though initially more computation resources are required, a reusable model may need fewer resources in the long run.

### Model size

Model size is not only relevant for the external entity storing the model, but also for the transmission step and generation process. Different types of models need different amounts of data to describe them. For example, a support vector machine is described by a set of support vectors and some parameters. Even more compact is a simple linear regression model, as it only needs a few parameters. In contrast, the description of an ANN comprises its topology, neuron weights, activation functions, and further parameters. Especially in case of deep ANNs, the number of weights may grow very large, yielding large models requiring many megabytes of data.

**Quality of approximation:** A surrogate model in general only provides approximations of the true flexibility. How well this true flexibility is approximated depends on many different parameters, including the general capabilities of the selected type of model, successful training, and the data used during the training process (see Section 3.5). While some types of models may already yield good results from a relatively small amount of data, others may need much more. Furthermore, approximation is a more difficult task for reusable models than for single-use models, since additional inputs must be interpreted (see also Section 3.3).

**Optimization:** In general, an external entity is not interested in identifying just any feasible control signal, but instead wants to find signals beneficial to their overall goals. While it is possible to generate a set of random feasible load schedules or other possible signals for all flexibility providers and then select the best combination, a more systematic and targeted search approach is desirable. In case of the schedule repair method, one may try to identify the best possible schedule, ignoring all constraints, and then use the surrogate model to find a (hopefully) close-by feasible schedule. Such an approach is evaluated in [3]. When generating load schedules from abstract representations and similar representations (genotypes) lead to similar load schedules (phenotypes), the external entity may be able to optimize the schedules with...
As stated before, surrogate models for demand side flexibility only may actually be infeasible. The only way for an external entity to know for sure whether a specific signal is feasible or not, is to ask the associated flexibility provider. This brings us to the question of how an external entity should deal with this issue. In the following we present building blocks for a possible solution:

Making use of a feedback mechanism: Figures 4, 5 and 6 all indicate the possibility of the flexibility provider sending feedback to the external entity. Such a feedback could, for instance, be the load schedule planned by the EMS as a reaction to the received signal. The external entity can then use this feedback to evaluate whether the predictions were accurate and whether corrective actions are needed. Different kinds of corrective actions are possible, depending on the type of model. In case of reusable models, the feedback data may be used to improve the models by conducting further training steps. In doing so, it is also possible to deal with gradual or incremental concept drifts, as the model can gradually be updated. Concept drift refers to a changing relation between input data and target variable over time [20]. If an abrupt concept drift is detected, for instance caused by the installation of new DERs, the external entity may alternatively dismiss the current model and restart the model generation from scratch. For single-use models, concept drift is irrelevant, as the model is only expected to be valid for a very limited time.

Making use of statistical information: Some types of machine learning models are able to provide us with statistical information, such as the distribution of a predicted value or estimated probabilities. This information may for instance be used to predict confidence intervals or avoid choices for which the surrogate model predicts high uncertainty. None of the examples listed in Table 1 makes truly use of such a statistical approach. The only related application would be the limitation of the set of feasible schedules by adapting classification thresholds as discussed in the following.

Limiting the flexibility: In cases where the feasibility of a signal is more important than making available every last bit of flexibility, the artificial limitation of the encoded flexibility is a viable and effective option (see [15, 16]). This limitation can be achieved in multiple ways: firstly, by encoding only a subset of the true flexibility and thereby constricting the surrogate model itself [16]. Such a limitation can be incorporated by applying artificial constraints to the data used during model generation. As an example, think of preventing power levels near the nominal power of a DER. If later, in the application of the model, too large power levels are predicted, this artificial buffer can help to keep the prediction inside the true boundaries. A second possible way is to manipulate model inputs [15] in order to shift them closer to their boundaries, which again creates an artificial buffer. However, this option must be used with caution, as the opposite boundary is shifted away. Finally, in cases where classifiers are used, one may be able to tweak false positive and false negative rates. If the classifier returns a rating between 0 and 1 indicating the confidence of a match, e.g., a load schedule being feasible, one may increase the threshold

<table>
<thead>
<tr>
<th>Reusable</th>
<th>Signal generation</th>
<th>Model type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Various methods</td>
<td>ANN</td>
<td>[12, 15, 16]</td>
</tr>
<tr>
<td>No</td>
<td>Classification</td>
<td>SVDD</td>
<td>[7]</td>
</tr>
<tr>
<td>No</td>
<td>Repair</td>
<td>SVDD</td>
<td>[3, 8, 43]</td>
</tr>
<tr>
<td>No</td>
<td>Classif. of fragments</td>
<td>Different classifiers</td>
<td>[40–42]</td>
</tr>
<tr>
<td>No</td>
<td>Repair</td>
<td>Chi-Shapes</td>
<td>[4]</td>
</tr>
</tbody>
</table>

**Scalability:** A DSMgr generally manages a larger number of flexibility providers at the same time, whether their goal is to resolve congestions in the electricity grid or to plan and control the operation of a virtual power plant. For the DSMgr, that is, the external entity, this means that they must be able to process multiple surrogate models in parallel. Hence, efficient and scalable algorithms for signal generation are crucial, especially when signals must be determined fast. The aggregation of DERs or flexibility providers is one possible way to improve the scalability of an exploitation mechanism, as fewer signals need to be generated. Disaggregation of the signal may be performed on a subordinate level of the control hierarchy. How aggregation can be achieved (e.g., [5, 10]) and how existing surrogate models can be aggregated into a new combined surrogate model (e.g., [21]) depends on the model type.

As already pointed out in Section 3, some surrogate models are reusable while others are not, which more or less has an impact on every aspect discussed in this section. In Table 2, we list different types of machine learning models, associated examples, and point out whether the models are generated in a reusable way or not. Reusability is by no means limited to ANNs. For example, any regression model taking the DERs’ states as an input could be reusable. However, the only surrogate modeling approaches with reusable models we could identify in the literature are ANN-based.

4.3 Dealing with Inaccuracy
As stated before, surrogate models for demand side flexibility only provide approximations of the underlying, true flexibility. Depending on the quality of a model, the majority of the derived signals may actually be infeasible. The only way for an external entity to deal with this issue is to ask the associated flexibility provider how an external entity should deal with this issue. In the following we present building blocks for a possible solution:
in order to decrease the false positive rate. As a result, the false negative rate will generally increase, since now a more limited flexibility is described. Low false positive rates at the cost of higher false negative rates may also be achieved by knowingly overfitting the model to the training data [16]. Whether any artificial limitations are needed or not depends on the exact use case. However, we suggest to look into the alternatives first before applying such limitations, since prediction errors may cancel each other out when multiple flexibility providers are jointly controlled.

5 CONCLUSIONS

In this paper, we provide an overview how models for the flexibility of DERs can be generated in an automated process and subsequently be utilized by external entities, such as DSMgrs. Either flexibility providers or external entities may generate the necessary models and each of both options has its own benefits. When flexibility providers generate models, in general, less data needs to be transmitted and more accurate models may be generated, but more computational resources are needed by the flexibility provider. The models are generated from data, but detecting flexibility in historical data is a challenging task, which requires assumptions or additional information. Aside from observations, possible sources for information include fake data, already existing models, and standardized DER interfaces.

There are multiple options for the automated generation of models for demand side flexibility and even more different types of surrogate models, which can be used to generate control signals. Which approach is the most fitting one, depends on the specific use case, therefore only an overview of different approaches is provided here. Regardless of the approach, we point out aspects to consider when developing a surrogate-based approach for the exploitation of flexibility. Most importantly, surrogate modeling only provides approximations of the true, underlying flexibility. The magnitude and possible consequences of erroneous control signals should always be kept in mind. Utilizing statistical information, provided by the surrogate models, and integrating a feedback mechanism are two possible ways to deal with inaccurate models.

Overall, the automated generation of surrogate models for demand side flexibility could simplify the integration of DERs, as one of the most labor-intensive integration steps is automated. Moreover, the amount of exploitable demand side flexibility could thereby be increased, which in turn would facilitate the integration of RES into the energy system. There is a wide range of aspects to consider when developing such an approach.

ACKNOWLEDGMENTS

We gratefully acknowledge the financial support from the Federal Ministry for Economic Affairs and Energy (BMWi) for the project C/sells (funding no. 03SIN121). Furthermore, the present work was partially supported by the Helmholtz Association under the program “Energy System Design” in the Research Field Energy.

REFERENCES


