Re-enacting rare multi-modal real-world grid events to generate ML training data sets

Daniel Hauer[ab], Matthias Bittner[a], Stephan Cejka[a], Ralf Mosshammer[a],
Florian Kintzler[a], Thomas Leopold[b], Stefan Wilker[b]
[a]Siemens AG, Vienna, Austria, {firstname.lastname}@siemens.com
[b]TU Wien, Vienna, Austria, {firstname.lastname}@tuwien.ac.at

***Abstract*—Today's energy grids are facing huge challenges caused by the growing diversity of energy consumers and producers as well as an ongoing increase of renewable energy sources and e-mobility. Hence, it is essential that the grids continuously evolve by introducing new monitoring, protection and optimization concepts including machine learning (ML) approaches. To overcome the lack of existing monitoring data for rare real-world grid events, this paper presents a concept for generating training data sets for ML approaches based on a multi-modal grid simulation tool. The simulation tool as well as the proposed semi-automated data generation approach are introduced and the concept is verified based on a real-world battery storage maintenance event.**

***Index Terms*—Smart grids, Simulation, Deep learning**

I. INTRODUCTION

While traditional power grids are characterized by a uni-directional energy flow from large power plants to passive consumers, Smart Grids have to cope with bidirectional load flows to and from intelligent 'prosumers'. Renewable energy sources, such as photo-voltaic systems or wind farms, generate energy in the low and medium voltage grids while on the other hand a growing number of e-cars have a high but hardly predictable power demand. These new players are no longer restricted to classic electrical load flow calculations and simple standardized load profiles, but introduce a variety of heterogeneous influence factors ranging from weather data, dynamic energy policies, to social aspects such as local price optimization within energy communities. In addition, the future grid has to operate closer to its limits to keep the level of necessary investments within an acceptable range.

If these challenges are not addressed properly either the transition towards modern Smart Grids is slowed down or the security of supply is endangered. The necessity for a quick transformation is underlined by an expected 93 % increase in global electricity generation during the 2010-2040 period and renewable sources to account for 24 % of total energy generation in 2040 [1]. Moreover, to mitigate the climate crisis, the European Union requires its member states to reach a share of renewable energy production of 32 % already by 2030 [2], though a further elevation of those target can be expected in near future in connection with the Green Deal program [3].

Several traditional distribution and protection concepts become inapplicable and new approaches have to be introduced. As a result, the research regarding machine learning (ML) and deep learning (DL) applications in the Smart Grid domain increased drastically, e.g., a quantitative descriptive analysis shows that 72 % of Smart Grid related ML research during the period from 2010 to 2019 was published in the last 3 years [4]. Especially in supervised learning, large training data sets are required, while often a lack of high-quality training data sets is experienced when training DL models to detect events and anomalies in the evolving heterogeneous Smart Grids.

Grids for which measurement data is available are either newly build or updated. Their properties are known and they tend to be rather stable, which makes it hard to extract sufficient training data sets for rare grid events. Thus, a technique is needed to generate the data for the ML algorithms even before state monitoring components are available in legacy grids, to which the new functionalities shall be added. Especially events that are unlikely but may impose a high severity are hard to be tackled. They cannot be sufficiently tested or observed in the real environment but may be addressable if a suitable method models the cause of these events in simulations.

We therefore propose an approach to re-enact rare multi-modal grid events to generate ML training data sets using Bifrost, a heterogeneous Smart Grid simulation tool [5]. The key contributions of this paper are:

A. To introduce Bifrost and its enhancements for re-enacting real-world multi-modal grid events,
B. To present a method for semi-automated data generation of realistic real-world grid events using Bifrost,
C. To verify the approach by generating a training data set for a maintenance event at a battery storage system within a low voltage grid section, and
D. To train a Long Short-Term Memory (LSTM) deep learning network architecture on this simulated test set, for detecting maintenance events in real-world data.

The rest of this paper is organized as follows: After a review of related work in section II, the simulation tool Bifrost and the proposed data generation approach is introduced in section III. The use case and its experimental setup is described in section IV and the results are evaluated in section V. Finally, conclusions and an outlook are given in section VI.

II. RELATED WORK

A. Modeling and Simulation of Smart Grids

A central question concerns the areas to be covered by a Smart Grid model/simulation and the functionalities provided

978-1-7281-9023-5/21/$31.00 ©2021 IEEE

by current available simulation tools [6]. In fact, there are many simulation environments [7] that are also effective in simulating the individual aspects of a Smart Grid (power flow simulation, demand/response, dynamic pricing, communication). However, if one wants to simulate the Smart Grid in its multi-modal nature, current solutions do not provide satisfactory results, as they were not developed for this modular purpose. To integrate these modular approaches there exists research about co-simulation [8], to create interactions between existing simulation frameworks.

B. Machine Learning/Deep Learning in Smart Grids

Regarding the application of DL in Smart Grids, most of the trending topics are related to load/demand forecasting, security and reliability of the grid, cyber security, power system analysis and control, renewable energy generation prediction and defect/fault detection of electrical equipment [9]. While all these topics aim for different goals (e.g., cost optimization, outage prevention, reduction of maintenance work), the used ML approaches are mostly based on deep Artificial Neural Networks (e.g., Recurrent Neural Networks, LSTM Networks, Convolutional Neural Networks, Feedforward Deep Networks) and Support Vector Machines, which together make up 74 % of ML methods in the Smart Grid domain [4]. These approaches share a common need for large training data sets.

III. Multi-modal Smart Grid Simulation

A. Bifrost Core

The co-simulation framework Bifrost[1] consists of a core simulation engine to drive dynamic data generation and a 3D web UI for the construction of settlements. The Bifrost core itself does not make assumptions as to the provenience of domain data, nor produce any. It does, however, provide a data model, which is built from a plain-text *directory*. This directory, which is freely editable even during runtime, lists syntactic (the shape of data, e.g., that a voltage consists of 3 floating-point values) and semantic (e.g., that voltage has a unit of Volt) characteristics. Within the Bifrost data model *dynamics* represent those aspects that can change (e.g., due to user interactions as in the case of a power switch, due to underlying models as in the case of houses' power consumption, or by an external simulation controller, cf. subsection III-C).

External modules, connected via a Representational State Transfer - Application Programming Interface (REST API), can subscribe to the Bifrost data model. At every simulation loop, all registered modules are called in-order, with a payload corresponding to their subscribed data. The modules in turn can respond with modified, updated, or new dynamic values that are stored in a time-series database, and can be visualized in graphs directly on the Bifrost UI. Figure 1 shows the Bifrost web UI design and highlights the different heterogeneous building types. While the Bifrost core and its modules can be fully controlled via the REST API, a graphical interaction (play/pause, module configuration and result visualisation) helps the user to construct and test the individual settlement.

[1]https://bifrost.siemens.com

B. Multi-modal Bifrost Modules

The main strength of Bifrost comes with the open interface to nearly any kind of behaviour model (Bifrost *module*) and its flexibility of allowing new and diverse characteristics (Bifrost *dynamic*). While classic load flow related modules range from standard load profiles (e.g., for residential buildings) to a load flow solver, heterogeneous aspects can be introduced by modules such as a weather generator or controller modules (e.g., battery storage controller or energy community controller). Following an overview about those modules with respect to the multi-modal nature of Smart Grids, which were used for the use case presented in section IV, will be given. This list is not complete but limited to the presented work.

1) Load Flow Solver: This module covers the basic load flow within an electrical grid. All power values generated by other modules as well as the grid topology are analyzed and the load flow is calculated and written back to the current simulation step.

2) Weather Generator: This module introduces various weather parameters into the simulation environment. It generates dynamics such as temperature, cloud coverage and precipitation. All values can either be taken from real historic weather data sets or can be generated based on any model.

3) Building Model: The main task of the building module is to calculate the power consumption of residential and commercial buildings. It is therefore split into sub-classes, each dealing with a specific domain. The load class provides the base load consumption of the building. Those values can be extracted from standardized load profiles or specific use case related profiles. Additional effects such as randomization or noise overlay help to generate different and more realistic profiles. A photo-voltaic (PV) class handles optional solar panels on each individual building. Beside the electrical characteristics and geographical orientation, this class uses the weather data to calculate the PV output. In addition, an e-mobility (EV) class simulates the charging of an e-car based on information about the charging pole, e-car and some parameterizable characteristics. Additional features like local battery storage, or heating pumps can be added.

4) Battery Model: This module simulates large battery storage systems, which are not part of a building but directly connected to the grid and controlled by global or communal instances. Possible applications could be the provision of (primary) control energy, local overload prevention or optimization of energy communities. The module therefore simulates a realistic behaviour of configurable batteries (e.g., by adding features such as aging and self discharge) and provides an interface for other modules, which can control the battery by sending charging and discharging commands.

5) Battery Controller: This module is responsible for adding any kind of battery controlling strategy. Implemented algorithms include, for example, a peak-shaving method, which uses the battery to prevent local transformer overloads during the daily peak times. The main functionality used for the presented work is to re-enact a maintenance event at the battery. This event is described in detail in section IV.

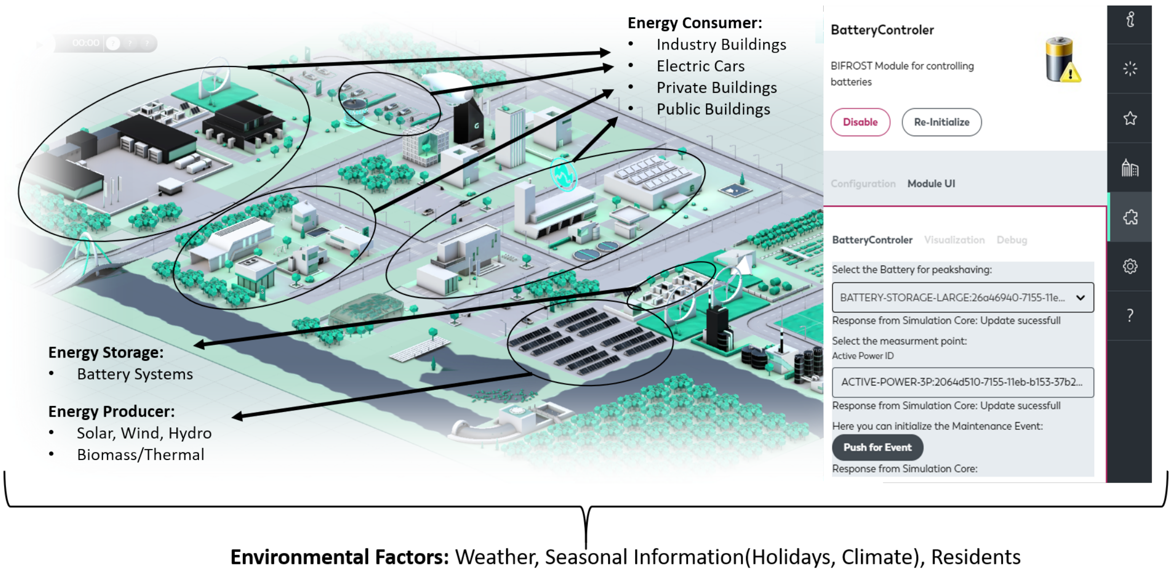

Fig. 1. Bifrost web UI with its heterogeneous building types and an exemplary module interface

6) Time Controller: Although the Bifrost core handles the simulation time and step size, the additional Time Controller module is needed to manipulate the simulation time in Bifrost to target specific timestamps. This is necessary in case certain parameter settings are influenced by the current simulation date and time (e.g., weather).

For future extension of the use case described in section IV the following modules are currently being implemented:

- Energy community controller: To model local energy communities.
- Power plant modules: To model wind farms or emergency backups like gas turbines.
- Inclusion of the energy price market: To model the dependency between consumption and the energy market.
- Public charging slots: To model the charging behaviours of e-car users at public charging slots.
- Augmented Bifrost Reality: To connect real-world sensors to Bifrost (e.g., for real-time co-simulation)

C. Semi-Automated data generation

We propose to use a multi-modal simulation tool such as Bifrost to create training data sets for rare multi-modal real-world grid events as a means to generate large and high-quality data sets, which are crucial for the success of many ML approaches. The approach proposed can be split into four steps, which will be introduced and described in the following:

1) Event Identification: Prior to every simulation run, the event under investigation has to be identified and analysed. This step typically involves domain experts such as grid operators and stakeholders. Once an event is identified (e.g., grid endangering weather behaviour), it has to be analysed and translated into the simulation world. Bifrost modules to re-enact a real-world event can be created in two ways:

- *Recreating the event as a time series*: If the event can be characterized by any kind of time series (e.g., specific load profile or weather period), a Bifrost module can be used to replay this time series. Additional randomization or noise overlay can help to improve the quality of the resulting training sets.
- *Recreating the cause of the event*: If possible, instead of the event itself, the underlying cause of the event should be modeled and implemented as a Bifrost module. Thus e.g., for a battery maintenance event the controller of the battery can be modeled instead of modeling the resulting time series. Using this approach, modeling can also be done even before any real-world event was recorded. Thereby specific behavior models (e.g., physical model) or abstract model approaches can be used.

2) Bifrost Settlement Setup: In a second step a simulation environment is specified by building a settlement that contains the grid topology with different types of buildings and integrates the needed Bifrost modules (see subsection III-B). This settlement should mirror the real-world situation, in which the event under investigation can occur. It is worth noting, that the setup can be iteratively optimized based on the extracted training sets and their verification.

3) Semi-automated Simulation Runs: After a settlement was specified, a single simulation run can be started and stopped via the Bifrost frontend. However, using the semi-automated simulation control tool, all simulation parameters and the event scheduler can be adapted between multiple Bifrost simulations, which are started and stopped automatically. This process is visualized in Figure 2.

First, the entry point of the simulation has to be set according to the first two steps. This includes the settlement design and constant module parameters such as the start timestamp, simulation step resolution, chosen power profiles, as well as specific module parameters (e.g., capacity and charging power of battery storage). Next the parameter sets and an event scheduler for the event under investigation have to be defined:

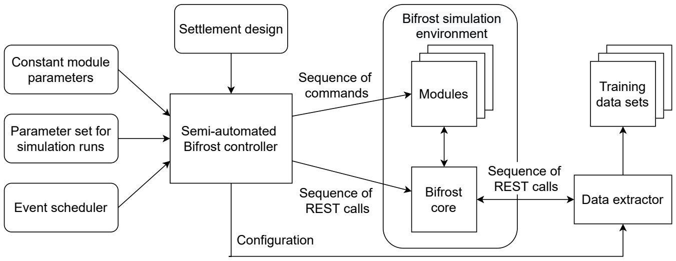

Fig. 2. Semi-automated data generation concept

- *Parameter sets*: This includes all parameters and corresponding ranges, which should be varied between the automated simulation runs. As Bifrost itself makes no assumptions about the module's purpose and function, a huge variety of modules and therefore possible parameter sets allow for diverse and realistic simulation results. For example, this could include a list of building power profiles, which should be replaced in every single simulation run. This would result in multiple runs with a different base load.
- *Event scheduler*: The event under investigation has to be triggered multiple times during the simulation run. Depending on the event and its nature, this could either be a simple schedule, which calls the event to given and maybe randomized timestamps, or a more complex model- or data-based sequence. As for all Bifrost modules, the event module designed in the prior steps can be triggered via a REST call.

After all parameters and schedules are set, the semi-automated Bifrost controller takes this information as input and generates a timed sequence of commands and REST calls, which are then automatically sent to the Bifrost core and any included module. Without further human input, multiple simulation runs are started, stopped and the resulting data sets are stored.

4) Training Data Extraction: After the automated simulation runs, the training data sets have to be extracted from the simulation results. Bifrost's data crawler module stores all simulation data (Bifrost *dynamics*) in an time-series database. The data extraction is then responsible for automatically collecting the data sets of the different simulation runs and storing them for later training of ML applications. In addition the following data processing steps are applied:

- *Data post-processing*: The simulation data is optimized for the target ML architecture. This includes, for example, manipulating the data and time resolution as well as data filtering (e.g., normalization).
- *Labeling*: Using the information from the Bifrost modules and the proposed controller enables automated data labeling. The event under investigation as well as other information (e.g., weather related events) are labeled and stored together with the data sets.

The resulting data sets can now be used according to the defined use case and ML approach. If possible, the quality of the simulation data should be verified by applying the targeted approach on historic real-world data. Experiences from such real-world tests can help to improve the training data set quality be redefining the simulation parameters and rerunning the semi-automated data generation. To illustrate the proposed data generation as well as to verify the overall concept, section IV now shows the results for a battery storage maintenance use case.

IV. Battery Storage Use Cases

In modern power grids an increasing number of parties are producer and consumer at the same time. These so-called prosumers are not restricted to a classic standard profile anymore and therefore represent a potential risk for the grid operation, especially when it comes to grid stability. Typical examples of these prosumers include PV systems, e-mobility charging stations or wind turbines.

As can be assumed from the above examples, in some cases it is a matter of coincidence if, when, and how the energy is distributed through the different grid segments. One measure to reduce potential local load peaks or over-production of energy is the use of battery storage systems. These can actively contribute to dampen the instabilities which are generated by the prosumers. Such battery storage systems are already part of modern real-world Smart Grid concepts, but there are still essential research gaps with regard to the analysis and detection of fault cases.

A. Exploratory Analysis

The available historical time series of the grid values were recorded over a period of several years using the grid monitoring devices installed at the Aspern[2] testbed in Vienna, Austria. For the presented use case, time series from a single distribution substation, supported by a battery system, from 2018 with a given sample rate of 2.5 min are considered. The underlying battery system is configured to limit grid peaks per phase to 60 kW. From the initial analysis of the power consumption data, it is observable that there are states in the system that do not represent the desired/designed behaviour. Figure 3 shows the following observed day-profiles:

- Battery maintenance event,
- Peak-shaving inactive over the whole day,
- Peak-shaving partly active over the day,
- Peak-shaving active for the whole day,

whereas the first three are considered as anomalous profiles.

B. Day-profile Clustering

As illustrated in Figure 3, one can classify the historical time series into a certain number of day-profiles. To acquire the number of different classes and their appearance frequency, an unsupervised day-profile clustering concept was implemented, which is divided into three major steps:

- Feature extraction (spatial, temporal and statistical domain) of the time series of one day.

[2]https://www.ascr.at

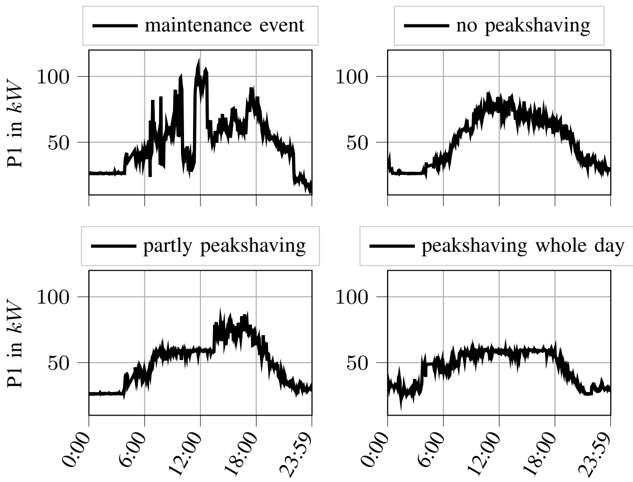

Fig. 3. Consumption day profiles of the investigated transformer

- Unsupervised (k-means) clustering in the corresponding feature space of the different days.
- Extracting the information about the frequency of the different day profiles.

After applying this approach on the historic time series, four different classes are identified. Table I shows the profiles and their occurrence in percent.

day-profile	maintenance event	no peakshav.	partly peakshav.	whole day peakshav.
appearance	1%	17%	68%	14%

TABLE I
RESULTS DAY PROFILE CLUSTERING

Depending on the use case and ML algorithm, the classification of the time series is helpful for ML training. However, the clustering also provides the information that the class "maintenance event" only occurs with 1% in this data set. To train neural networks to recognize this event in the real-world data online, significantly more training samples are required.

C. Battery Maintenance Event

This event is caused by a local maintenance work at the battery storage, which is not reported to central control. The battery is completely discharged and charged. This operation is often accompanied by an update of the battery controller, which changes the system properties. Thus, the detection of these events is of importance to the grid operator.

In order to reproduce the event in our multi-modal simulation, the information about the root cause and the connected parameters must be known first. In this case, the event is manually triggered on site. So time and date are random in this situation. The duration of the maintenance event depends on the storage size of the battery and the maximum charging and discharging power. These parameters are now specifically manipulated in the simulation, see V-A.

V. Results

A. Generated Training Data

The approach presented in subsection III-C is now used to simulate training data for the battery storage use case (section IV), especially to generate time series for identifying the battery maintenance event in the historic data. For this the used Bifrost settlement is designed to behave in a similar way like the testbed Aspern[2], and in order to create a diverse training set the parameters of the Building Model, Battery Module and Battery Controller (e.g. load profiles, event start time, charging/discharging power of the battery, see subsection III-B) are modified between simulation runs.

Figure 4 illustrates the created/simulated normalized training data $p1$, where the battery maintenance event is at least triggered once per day. The lower part of this figure contains two example day profiles. The corresponding labels are automatically extracted from the simulation and mark the discharging and charging period of the battery.

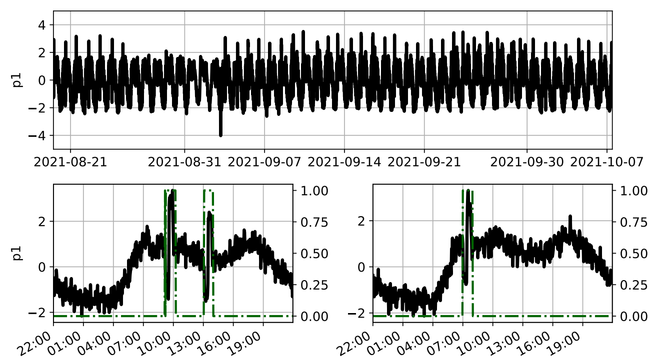

Fig. 4. Simulated data set

B. Neural Network Architecture

When it comes to analyzing/classifying time series, LSTM networks are frequently used [10]. Their ability to learn patterns in a sequence seems almost perfect for this battery maintenance use case. In this application the Keras framework[3] and its LSTM implementation is used. Table II lists the implemented layers as well as those parameters, that differ from the standard implementation. Additionally a dropout layer (dropout rate of 0.1) is inserted between the individual layers in order to avoid over fitting.

layer	L1: LSTM	L2: LSTM	L3: Dense
parameter	units = 264 rec.act. = "tanh"	units = 64 rec.act. = "tanh"	act. = "tanh"

TABLE II
NEURAL NETWORK ARCHITECTURE

The network is trained with a sequence length of 25 (which corresponds to approx. 1 hour with a sampling rate of 2.5 minutes). The input to the network is defined as
$\mathbf{X}_{train}$ = $[\mathbf{p1}, \dot{\mathbf{p1}}, sin(\omega\mathbf{t})] \in \mathbb{R}^{25\times3}$, where $\mathbf{p1}$ is the normalized (zero mean, unit standard deviation) time series of the power consumption and $\dot{\mathbf{p1}}$ its derivative. To let the network also recognize/learn relative temporal relationships, the third feature vector consists of the relative time $sin(\omega\mathbf{t})$, where $\omega = \frac{2\pi}{24\cdot60\cdot60}\frac{rad}{s}$.

[3]https://github.com/keras-team/keras

C. Event Classification

After training the previously presented LSTM network with 30 $epochs$ and a $batch\ size$ of 200 samples, we achieve a training accuracy of 98%. The network is therefore capable of detecting the event in the simulated data. However, more interesting are the results on the real grid data. Figure 5 shows the prediction of the network on the real grid time series of 2018 (with a recording gap due to malfunctioning sensors during autumn). The goal of this real time event detection is to recognize behaviours which are similar to the battery maintenance event sequence. Considering the result of the day profile clustering (battery maintenance event appears 4 times) as the ground truth we result in a validation accuracy of 80%.

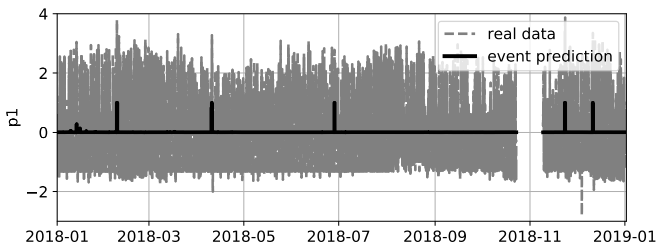

Fig. 5. Overall battery maintenance event classification

Figure 6 illustrates four detailed day profiles and their battery maintenance event prediction. Apparently in all four cases anomalous behavior is prevalent and the battery can be the reason for this behavior. However, the main attention should be drawn to the two day profiles on the right hand side of the figure. In 2018-04-10 the maintenance event is detected in addition to two other anomalies. In 2018-11-23 the event is carried out twice in a row and is detected on both occasions.

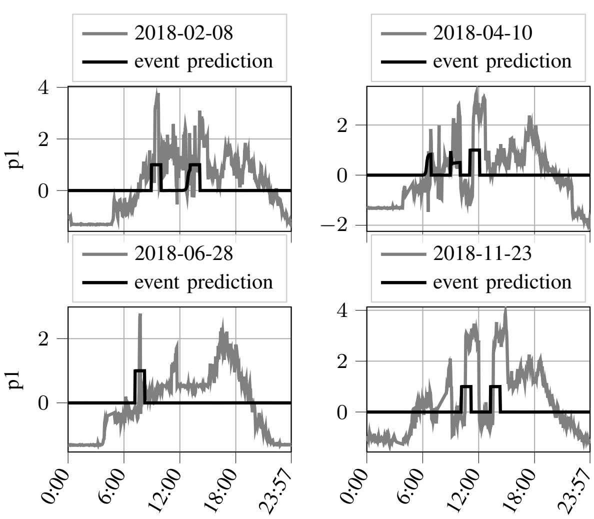

Fig. 6. Daily based battery maintenance event classification

VI. Conclusion and Outlook

We demonstrated a novel concept for re-enacting rare multi-modal real world grid events to generate training data sets for ML algorithms. The approach is based on the heterogeneous simulation tool Bifrost as well as a semi-automated simulation controller. The approach was verified by re-enacting a battery maintenance event within a low voltage grid section. The results show that it is possible to generate suitable training data sets for this event and to use them to train a LSTM network architecture, which is able to detect the maintenance event within real world data from a Smart Grid testbed.

The main challenges lie in the semi-automated simulation configuration as well as the parametrization with respect to the ML algorithm since the simulation configuration is of great importance for the quality of the results. Although the used modules already provide a heterogeneous and realistic environment for many use cases, domain experts and knowledge have to be included into the configuration process for the event under investigation.

To overcome these limitations, we are currently working on new approaches on event and outlier detection without a-priori knowledge about events. For example, the concept can be adapted to create training data sets which are then used to compare real world data streams with a predicted one based on the training data. Furthermore, the addition of real controller hardware could also be used to improve the parameter tuning for real world components (e.g., control strategy).

Acknowledgments

This work was partially supported by EU H2020 IoTwins Innovation Action project (g.a. 857191). The presented approach will be implemented and tested in the Smart Grid testbed of Aspern Smart City Research (ASCR), which also provided the data used for first investigations. In addition, this work has been partially funded by partners of the ERA-Net SES 2018 joint call RegSys (www.eranet-smartenergysystems.eu) - a network of 30 national and regional RTD funding agencies of 23 European countries. As such, project SONDER has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 775970.

References

[1] A. Colmenar-Santos, C. Reino-Rio, D. Borge-Diez, and E. Collado-Fernández, "Distributed generation: A review of factors that can contribute most to achieve a scenario of DG units embedded in the new distribution networks," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 1130–1148, 2016.

[2] European Commission, "2030 climate & energy framework," 2019, https://ec.europa.eu/clima/policies/strategies/2030_en (accessed 22.02.2021).

[3] A. Sikora, "European Green Deal - legal and financial challenges of the climate change," *ERA Forum*, vol. 21, no. 4, pp. 681–697, 2021.

[4] T. S. Bomfim, "Evolution of machine learning in smart grids," in *2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)*, 2020, pp. 82–87.

[5] R. Mosshammer, K. Diwold, A. Einfalt, J. Schwarz, and B. Zehrfeldt, "Bifrost: A smart city planning and simulation tool," in *Intelligent Human Systems Integration 2019*, W. Karwowski and T. Ahram, Eds. Cham: Springer International Publishing, 2019, pp. 217–222.

[6] M. Pöchacker, A. Sobe, and W. Elmenreich, "Simulating the smart grid," in *2013 IEEE Grenoble Conference*, 2013, pp. 1–6.

[7] M. Jdeed, E. Sharma, C. Klemenjak, and W. Elmenreich, "Smart grid modeling and simulation — comparing gridlab-d and rapsim via two case studies," in *2018 IEEE International Energy Conference (ENERGYCON)*, 2018, pp. 1–6.

[8] T. Duy Le, A. Anwar, R. Beuran, and S. W. Loke, "Smart grid co-simulation tools: Review and cybersecurity case study," in *2019 7th International Conference on Smart Grid*, 2019, pp. 39–45.

[9] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 3, pp. 362–370, 2018.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.