Reporting Interval Impact on Deep Residential Energy Measurement Prediction

Grigore Stamatescu
Department of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

Irina Ciornei
MicroDER Lab, University Politehnica of Bucharest
KIOS Centre of Excellence, University of Cyprus
ciornei.irina@ucy.ac.cy and irina.ciornei@upb.ro

Radu Plamanescu
Faculty of Electrical Engineering
University Politehnica of Bucharest
Bucharest, Romania
radu.plamanescu@upb.ro

Ana-Maria Dumitrescu
Faculty of Electrical Engineering
University Politehnica of Bucharest
Bucharest, Romania
anamaria.dumitrescu@upb.ro

Mihaela Albu
Faculty of Electrical Engineering
University Politehnica of Bucharest
Bucharest, Romania
mihaela.albu@upb.ro

***Abstract*—Forecasting and anomaly detection for energy time series is emerging as an important application area for computational intelligence and learning algorithms. The training of robust data-driven models relies on large measurement datasets sampled at ever increasing rates. Thus, they demand large computational and storage resources for off-line power quality analysis and for on-line control in energy management schemes. We analyze the impact of the reporting interval of energy measurements on deep learning based forecasting models in a residential scenario. The work is also motivated by the development of embedded energy gateways for online inference and anomaly detection that avoid the dependence on costly, high-latency, cloud systems for data storage and algorithm evaluation. This, in turn, requires increased local computation and memory requirements to generate predictions within the control sampling period. We report quantitative forecasting metrics to establish an empirical trade-off between reporting interval and model accuracy. Additional results consider the time scale variable feature extraction using a time series data mining algorithm for multi-scale analytics.**

***Index Terms*—energy forecasting, embedded inference, model robustness, cyber-physical systems, smart buildings**

I. Introduction

Using dense deployments of Internet of Things (IoT) devices, multi-variate electrical measurements are being reliably collected at ever increasing reporting intervals. This happens concomitantly with grid monitoring systems, at large commercial consumers to the lowest level, at the residential consumer. Making use of this dense data requires complex intelligent algorithms for prediction and anomaly detection. Many methods are described in the literature using mostly deep neural networks as: Fully Connected Neural Networks (FCNN) and sequence models such as Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), and convolutional hybrid models (CNN-LSTM) [1] alongside conventional machine learning methods as regression trees, support vector machines, etc. that allow fine grained control over the input data through expert guided feature engineering. For training of such systems, large amounts of quality input data are required to capture fine grained nonlinearities over various operating conditions. In some situations, like deploying models on resource constrained embedded hardware, there is a need to downsample the measurement or sensor signal for reduced training or inference time with bounded decrease in forecasting accuracy.

We investigate the dependence of the energy prediction model accuracy on the sampling interval of the input signal. This allows to adapt and fine tune the algorithm for detecting and anticipating both fast transient phenomena, like switching behaviour or faults, and more persistent changes in the signal behaviour, like appliance usage in daily activities. This approach applies to both uni-variate measurement time series (e.g., the power drawn at the electrical meter of domestic houses), and to multi-variate measurements (e.g, measurements of multiple electrical parameters, fine grained submetering traces for individual appliances or for significant consumers). These types of insights become also helpful when setting up an IT system for data acquisition, storage and processing of the electrical measurements, in conjunction with the latency and off-line/on-line analysis of the results. As the performance and complexity of modern measurement devices has increased, the resulting collected information can be classified using typical characteristics of big data in terms of volume, velocity, variety, veracity, and value. This requires specialized computing infrastructure and efficient primitives for information extraction [2] leading to improved labelling of relevant phenomena.

Motivation of the work is supported by the need of establishing a trade-off between cost and computational resources for training data-intensive models and the desired application accuracy for forecasting and classification. Novelty of the contributions lays in the analysis of the effect that the reporting

This work was supported by a grant of the Romanian Ministry of Education and Research, UEFISCDI no. PN-III-P4-ID-PCE-2020-2876 ”Advanced Measurement Framework for Emerging Electric Power Systems” (EMERGE).

interval has on the application of new deep learning methods with smart meter data. This comes as a direct consequence of the increase of smart meter applications not only at residential level but also at the grid level. The issue that the study deals with is at the convergence of measurement science, as applied to power systems dynamics and computer science, specifically machine learning algorithms as applied to energy related measurements. Main tasks that are carried out to derive the relevant results are as follows:

- A comparison on training and accuracy of four popular deep learning (DL) models applied for energy forecasting, using high reporting interval smart metering data;
- Illustrate an empirical dependence between the reporting interval of input data and the accuracy of the model (MSE, MAE, MAPE) along with a strategy for the generalisation of such approach to different scenarios.

The rest of the paper is structured as follows. Section II discusses related work mainly aimed at identifying suitable time periods for training data-driven time series forecasting models. We present the methodology, the algorithms and the benchmarking of the dataset in Section III. Section IV discusses the results achieved for the target application, including replicable implementation details. Furthermore, we provide an exemplification of energy time series feature extraction using the Matrix Profile algorithm at various time scales. Section V discusses the foreseen context of leveraging the results to automatically adjust the reporting interval based on required performance in multi-scale analytics for energy systems.

II. Related Work

Modelling and forecasting time series sampled at different frequencies in a general econometric context is discussed in [3]. The authors present their findings in the key that by lowering the sampling rate of the respective time series the core dynamic components remain observable while fine grained and seasonal elements become unobservable through aggregation. Disaggregation and establishing a correspondence between the lower and higher sampled data can be realised but requires a highly nonlinear model leading to inexact matching and reconstruction. This insight can be used in our technical context as well when discarding dense measurements due to lack of storage or computational limitations. Relevant, macroscopic scale, analysis on an energy system is presented in [4], where the authors report the decrease in computational requirements with various downsampling rates for renewable energy generation. A nonlinear decrease in normalized CPU time is reported when switching from 1h time-steps to 3, 6, 12 and finally 24h time-steps on 25 years of simulated wind and PV generation data from the UK. The suitable approach should be flexible to accommodate different input data and constraints regarding the modelling technique. Our goal through this contribution is to perform this approach at the microscopic level for low voltage residential consumers with different factors affecting the load shape, with second-level reported measurements.

A statistical framework to select appropriate sampling rates for time series analysis is introduced by [5]. The study combines historical data sampled at a slow rate with cost information for higher data rate collection, and a small subset of more frequently sampled data. The relation between the two can be framed as a missing data problem for the less often sampled dataset. Specific methods such as spectrum estimation and others can be applied to achieve a correspondence between the two. For the particular context of power system analysis (e.g, load flow calculations), the authors of [6] leverage feature extraction to reconstruct synthetically representative time series. The aim was to reduce computational demands of the algorithms with bounded degradation of the quality of models. Computational intelligence methods such as generative adversarial networks can be used to learn and extrapolate measurement time series patterns (e.g., TimeGANs) for generating quality datasets [7]. In our case we use realistically collected data in a residential context that could be augmented with synthetic measurements using such methods.

In [8] 1s load power profiles for residential consumers are analysed with the goal of detecting power steps in a sampled load power profile. A noninvasive error monitoring technique is devised through comparison of the tested and reference meters by means of synchronized statistical methods on the two measurement series. Smart energy information systems design with IoT features and reporting interval discussion are performed in [9]. Main contribution lays in establishing the requirements of an Energy Information Management System (EIMS) for large scale energy consumption in buildings: hardware and software for data collection, transmission and analysis. Embedded monitoring and control for energy storage systems is presented in [10] using distributed sensor and data acquisition nodes and hardware-in-the-loop type evaluation of the performance.

III. Methodology

We briefly introduce the methods, the reference dataset and associated metrics that we use for this work. Recently, many data-driven methods for energy time series forecasting rely on sequence learning models. These algorithms operate on subsequences of the input time series and can be used for both single and multi-step forecasting or for classification tasks. Standard implementation is in the form of recurrent neural networks (RNN). RNNs are neural network architectures with built-in loops, that allow the learning process to consider the time dependencies between individual components. By contrast, conventional neural networks use independent training per component. In order to mitigate the negative effects that might appear during training over long sequences (e.g., exploding or vanishing gradients) more complex architectures have been devised, such as gated recurrent units (GRU) and long short-term memory (LSTM) networks. A common characteristic of these structures is the use of dedicated ”gates” that control the information flow through the networks, and include additional trainable parameters for the gate weights. This allows the network to propagate relevant information through multiple time steps while selectively discarding irrelevant or redundant extracted features. The basic LSTM cell [1] includes an input gate (i), a layer input gate to update the cell state (g), a forget gate (f) for discarding information and an output gate (o). The state of the LSTM cell memory at time step t is updated through the Hadamard product, as element-wise multiplication of the matrix operands, as follows:

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes g_t, \tag{1}$$

The output state (h) at time step t is given by the output gate (o) which implements a read function combined with the cell state (c) as in:

$$h_t = o_t \otimes tanh(c_t), \tag{2}$$

where the output is expressed as:

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o). \tag{3}$$

based on the cell input (x) and with σ representing the activation function of the LSTM cell, W the weights of the cell, R the recurrent weights of the cell and b the bias terms.

Based on single layer LSTM networks, several variants are available and implemented through specific software packages. Further layers can be stacked for increased complexity and the ability to extract more fine grained features. Bidirectional networks are able to parse through the input sequences in both directions. An adaptation of the convolutional layers, typical for bidimensional inputs as encountered in image processing, can be applied for time series models by assembling the univariate input sequence vectors into bidimensional matrix formats and applying the convolution operator for feature extraction.

The dataset used in this study stems from a long term data collection of energy measurements from a typical residential appartment from Bucharest, Romania. The dataset is available for testing purposes from the authors. For illustration purposes, a daily plot of active power in Watts from the month of September 2020 is shown in Figure 1.

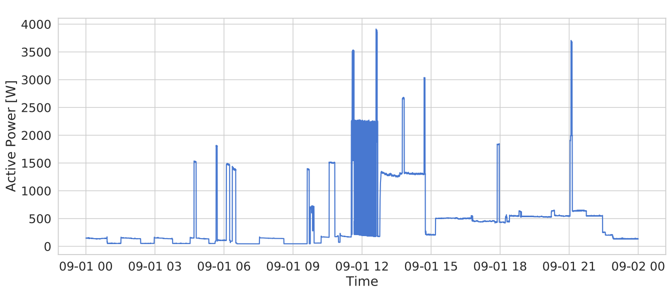


Fig. 1: Sample input data

For evaluation of the energy prediction performance at various reporting intervals we use the following metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). MSE and MAE balance small and large prediction errors, while MAPE provides a relative metric of accuracy that can be used across different scales of magnitude for the input. These are computed as follows:

$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}; MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n};$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| 100; \tag{4}$$

where y_i is the actual value of sample i, $\hat{y}_i$ is the predicted value of the sample i, and n the number of samples. In particular for MAPE we use the Python *sci-kit learn* package implementation[1] which as a small error term in the denominator to avoid division by zero and numerical inconsistencies.

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{max(\epsilon, |y_i|)} \tag{5}$$

where ϵ is an arbitrary small constant, thereby shifting the interval of the relative metric from $[0, 100]$ to $[0, 1/\epsilon]$.

A basic prediction example for the test set associated with one day of residential power measurements sampled at 1s and using a single layer LSTM network is illustrated in Figure 2.

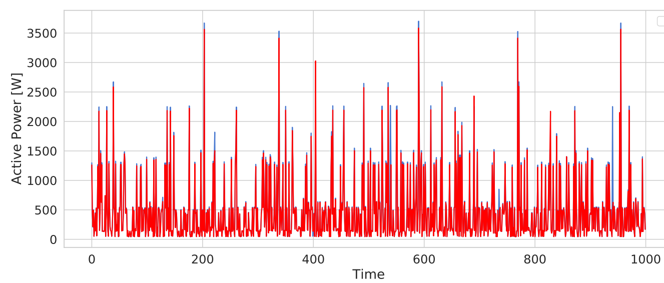


Fig. 2: Prediction results

IV. RESULTS

For the purpose of our study, we have trained and evaluated the following deep learning models: single-layer LSTM network (LSTM-1), two-layer stacked LSTM network (LSTM-2), bidirectional LSTM network (BiLSTM), and a hybrid convolutional and LSTM network (ConvLSTM). The number of units per layer is fixed at 50. For each of the models we report the MSE, MAE and MAPE testing metrics at the baseline (1s) reporting interval as well as 2x/5x/10x decimated reporting interval. The first goal is to derive an empirical relative dependency between the reporting interval and the prediction model accuracy for these variations of state-of-the-art deep learning algorithms. From the available data we establish a 70/30% split between the training and testing sets. The $random\_seed = 42$ parameter is set for the implementation to control for the randomness of the training and test datasets. The training data is reshaped in a suitable manner as input to the algorithms with the parameter $n\_steps = 4$ denoting that each training example uses a sequence of four previous

[1]https://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-percentage-error

values as input features. Each model is trained for 50 epochs. Implementation is based on [11] using *sci-kit learn v0.24*, *numpy*, *pandas* and *keras* packages on a server-class system with Intel Xeon processor and 16GB RAM under the Linux operating system.

Prediction test set accuracy results are summarised in Table I-III for each of the available metrics. The hardware and software dependant training time in Table IV is relevant for relative comparisons between the trained model types at different reporting intervals. In general the convolutional variant of the LSTM prediction model yields the best results albeit at very large training times. The bidirectional LSTM model provides the best trade-off between test set forecasting accuracy and training time in our study.

TABLE I: Test prediction results - MSE

	MSE $[kW^2]$			
	LSTM-1	***LSTM-2***	***BiLSTM***	***ConvLSTM***
Baseline	6.53	8.72	6.6	7.25
2x	16.36	15.59	16.46	15.92
5x	37.8	40.62	37	36.2
10x	62.46	61.52	65.48	60.12

TABLE II: Test set prediction results - MAE

	MAE $[W]$			
	LSTM-1	***LSTM-2***	***BiLSTM***	***ConvLSTM***
Baseline	9.9	14	9.5	19
2x	25	24	26	19
5x	39	47	34	45
10x	55	58	92	59

TABLE III: Test set prediction results - MAPE

	MAPE [%]			
	LSTM-1	***LSTM-2***	***BiLSTM***	***ConvLSTM***
Baseline	0.032	0.032	0.027	0.069
2x	0.066	0.05	0.056	0.078
5x	0.13	0.12	0.095	0.1
10x	0.12	0.18	0.17	0.18

TABLE IV: Training Time

	Time $[s]$			
	LSTM-1	***LSTM-2***	***BiLSTM***	***ConvLSTM***
Baseline	2282	3990	3276	4025
2x	1304	2268	1492	2517
5x	581	1038	878	868
10x	187	408	289	357

Figure 3 shows the comparison between test MSE and training time for the four models at various reporting interval reduction factors. A light nonlinear relation between both the error and decimation interval as well as training time and decimation interval of the time series can be observed. We can therefore reduce the input data reporting interval in accordance

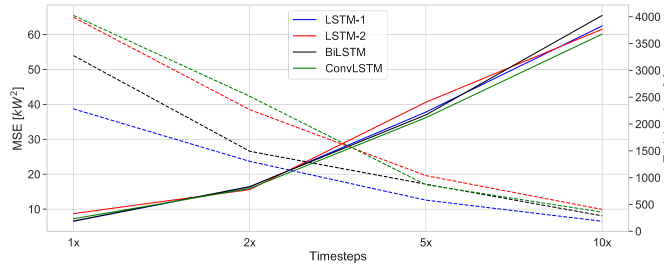


Fig. 3: MSE versus training time results

to the dynamics of the observed measurement phenomena with bounded decrease in MSE.

We attempt to further validate and generalized the study results by running one of the models (BiLSTM) on a full month of data as global model. The model is chosen based on the previous results that show the best performance in the trade-off between the decrease in MSE versus the increase in training time for a more complex model, over the various investigated reporting intervals of the data. The same parameters are kept, in particular the sequence length for the $n\_steps = 4$ parameter. The global approach is tested on the baseline reporting interval (1s) and the 10x decimated reporting interval (10s). The model structure with 20901 trainable parameters is presented in Figure 4. The model includes an input layer at the top of the diagram which receives training examples of four timesteps each. These are passed to the core hidden bidirectional LSTM layer which processes the data and feeds the output to a dense, fully-connected, layer. The output of this layer represents the final scaled energy prediction for the respective example. Model training attempts to identify the optimal model parameters (weights and bias terms) that minimize the average error over all the training examples. In the case of regression problems, the usual metric for optimization is the mean squared error.

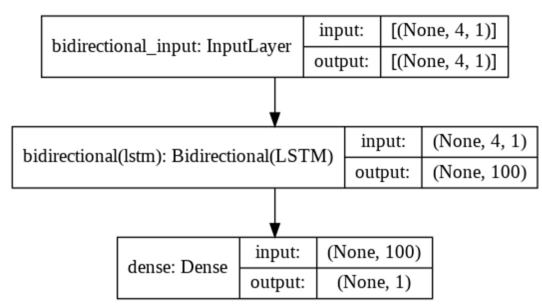


Fig. 4: BiLSTM global model structure

Figure 5 illustrates the training loss (i.e. MSE in Watts for our application) for the baseline model over the training epochs. The global model - 10s achieves $MSE = 10kW^2$ and $MAE = 24W$, while the global model at the baseline reporting interval of 1s achieves $MSE = 0.910kW^2$ and $MAE = 2.49W$. Training time for the decimated model is $t = 8400s$, while for the baseline model we use an early stopping criterion to stop the training once there is no significant decrease in the loss over multiple training epochs using the $patience = 3$ callback parameter.

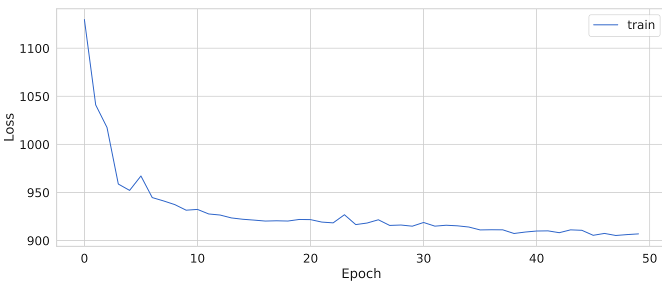
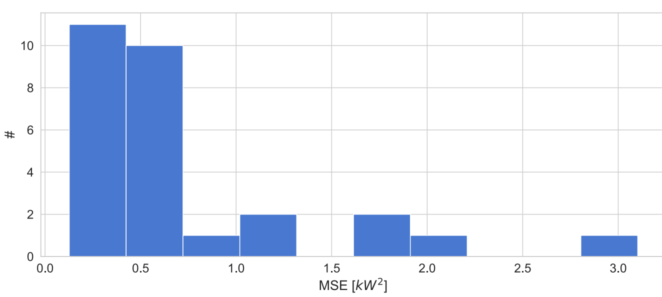
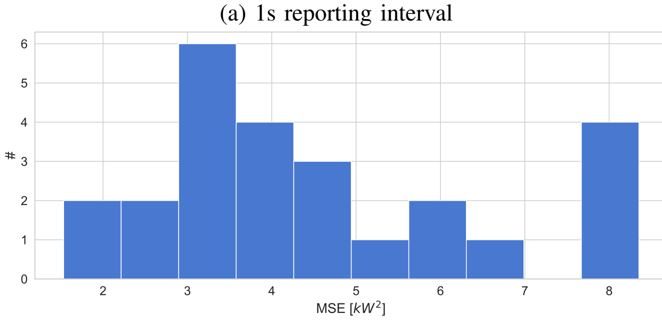


Fig. 5: Training loss for monthly global model (1s)

Individual models for a full month are also trained and tested, composed of 30 daily subsequences, corresponding to the month of September 2020, at the baseline and 10x decimated reporting intervals. The aggregated results are presented in the form of testing MSE metric histograms over the 30 individual models in Figure 6. Top 15% of the outliers have been eliminated from the error array. Further, segmentation of time of day and day of week models is possible for more specific forecasting performance. Reducing the variability of the MSE can be achieved for the residential energy use case by including contextual variables and time series in the model such as outdoor temperatures, albeit with increased complexity that can be quantified with regard to the provided benefits.



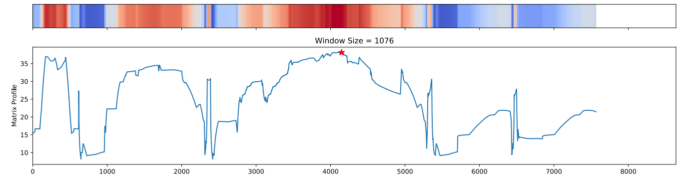
(a) 1s reporting interval



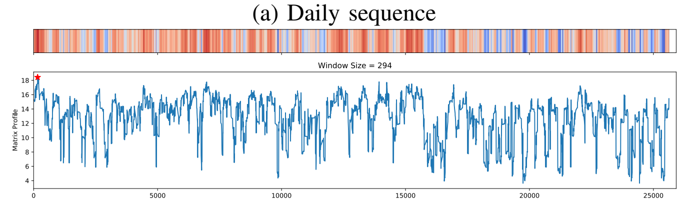
(b) 10s reporting interval

Fig. 6: Testing MSE distribution for independent daily models over one calendar month

We also present a Matrix Profile (MP) exemplification at the $1s$ reporting interval for multi-scale feature extraction applied to energy measurements. This is an efficient time series data mining method which allows feature extraction and anomaly detection over large series. The algorithm outputs the minimum sequence by sequence Euclidean distance based on a single parameter, the subsequence size, which is used for finding motifs, recurring patterns in the series, and discords, the most dissimilar patterns. Figure 7 (a) illustrates the computed profile for the baseline interval while identifying the most dissimilar sequence in the original daily data - corresponding to the readings at noon from the daily series. In Figure 7 (b) a similar analysis is performed for the monthly series at a similar reporting interval where the daily variations are reflected in low values of the profile. The main result here is the identification of the first day in the month (September 1st) as the most dissimilar sequence in the full measurement series.



(a) Daily sequence



(b) Monthly sequence

Fig. 7: Matrix profile for anomaly detection

V. Conclusion

We have investigated the performance of various types of deep learning models on residential energy measurement data at various reporting intervals. The goal was to establish an empirical relation useful for choosing the appropriate amount of data required to train a good quality model while considering the limitation of available computing resources. Future work will consider extending the study to publicly available benchmarking datasets such as Pecan Street Dataport [12] and use the derived results to guide a energy time series classification framework for steady-state evaluation on multivariate data.

References

[1] C. Nichiforov, G. Stamatescu, I. Stamatescu, V. Calofir, I. Fagarasan, and S. S. Iliescu, “Deep learning techniques for load forecasting in large commercial buildings,” in *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, 2018, pp. 492–497.

[2] C. Nichiforov, I. Stancu, I. Stamatescu, and G. Stamatescu, “Information extraction approach for energy time series modelling,” in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, 2020, pp. 886–891.

[3] J. Casals, M. Jerez, and S. Sotoca, “Modelling and forecasting time series sampled at different frequencies,” *Journal of Forecasting*, vol. 28, no. 4, pp. 316–342, 2009.

[4] S. Pfenninger, "Dealing with multiple decades of hourly wind and pv time series in energy models: A comparison of methods to reduce time resolution and the planning implications of inter-annual variability," *Applied Energy*, vol. 197, pp. 1–13, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261917302775

[5] G. P. Nason, B. Powell, D. Elliott, and P. A. Smith, "Should we sample a time series more frequently?: decision support via multirate spectrum estimation," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 180, no. 2, pp. 353–407, 2017.

[6] J. Henze, T. Kneiske, M. Braun, and B. Sick, "Identifying representative load time series for load flow calculations," in *Data Analytics for Renewable Energy Integration: Informing the Generation and Distribution of Renewable Energy*, W. L. Woon, Z. Aung, O. Kramer, and S. Madnick, Eds. Cham: Springer International Publishing, 2017, pp. 83–93.

[7] G. Baasch, G. Rousseau, and R. Evins, "A conditional generative adversarial for energy use in multiple buildings using scarce data," *Energy and AI*, p. 100087, 2021.

[8] Z. Nakutis, S. Rinaldi, P. Kuzas, and R. Lukocius, "An analysis of customer power profile events for non-invasive energy meter error monitoring," in *2019 IEEE 10th International Workshop on Applied Measurements for Power Systems (AMPS)*, 2019, pp. 1–6.

[9] J. Walia, A. Walia, C. Lund, and A. Arefi, "The characteristics of smart energy information management systems for built environments," in *2019 IEEE 10th International Workshop on Applied Measurements for Power Systems (AMPS)*, 2019, pp. 1–6.

[10] G. Stamatescu, I. Stamatescu, N. Arghira, I. Făgărăsan, and S. S. Iliescu, "Embedded networked monitoring and control for renewable energy storage systems," in *2014 International Conference on Development and Application Systems (DAS)*, 2014, pp. 1–6.

[11] J. Brownlee, *Deep learning for time series forecasting:MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.

[12] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, "Dataport and nilmtk: A building data set designed for non-intrusive load monitoring," in *2015 ieee global conference on signal and information processing (globalsip)*. IEEE, 2015.