[[1]](#footnote-1)

Household Energy Demand Predictions for IoT Systems

Leite, R.G., *P.G. Big Data Analytics, University of Liverpool, MSc. Computer Science student, University College Dublin, MSc. Electrical Engineering and Telecommunications student, Universidade Federal Fluminense*

***Abstract*—In modern societies, the ubiquity of computing devices highlights how data processing capabilities transitioned from revolutionary novelty to commodity. Lowering prices of smartphones, intelligent doorbells, or internet-enabled lightbulbs drive the creation of large software platforms responsible for the intensive computations requiring resources that would overwhelm more modest devices. Nonetheless, some use cases cannot cope with delays introduced by this distributed processing. For example, demand response proposals for smart electrical grids can expect millisecond-level thresholds. One possible way of mitigating this issue is by adding decision-making to the device. It is a case of edge computing that keeps a frequent communication channel with a large, distributed control plane open. However, having a concise, deterministic algorithm that can handle cases like consumption forecasting seems beyond our current understanding of electrical engineering. This restriction points to the need for some form of pattern matching process constantly updated by a training feedback loop. The edge device would handle decisions by applying this model trained by the central platform. This document investigates the options for proposals for time-series forecasting, including modern machine learning techniques. It examined the feasibility and trade-offs expected to satisfy the technical challenges posed by this scenario.**

***Index Terms*—Machine Learning, Internet of Things, Edge Computing**

# I. INTRODUCTION

The importance of demand-side control of consumer-grade electrical systems has increased in popularity in the past decade. It is now discussed as an integral component of research areas such as Smart Electrical Grids and Distributed energy generation. The underlying motivation for bringing this topic to a more central position is the overdue modernisation of energy distribution systems. There is a departure from a centralised, unidirectional control toward more fine-grained dispersed models.

Some technological breakthroughs did not fit this legacy top-down communication model. For example, the introduction of Solar and Wind generation requires a much more agile response given the unpredictable nature of their availability. The operator started dealing with greater levels of variance that more slow-moving forecasting models could not reliably manage.

Diagram

Description automatically generated

Figure - Demand Response, bidirectional communication

This landscape gave rise to energy aggregation services acting as independent coordinators. Industry players place bids on supply requirements advertised by system operators. It is the first step in decentralising control. However, decentralising the generation control might still require more fine-grained information dissemination.

This diffusion level enables more radical proposals from the demand-side of the energy distribution equation. Research is often not tied to the same constraints imposed by routine business concerns. There are countless proposals for communication architectures inspired by many disciplines, from Neighborhood-level microgrids to multi-layered energy distribution cells, similar to cellular telephone zoning.

This restructuring shares the traits of similar past modernisation endeavours. One often overlooked is moving consumers from passive receivers of information to active platform participants. The main driver is the move from simple broadcasting to much more immersive bidirectional channels. Other industries went through similar updates. The advertisement industry was the first to exploit bidirectionality to establish feedback loops, thus driving revenue increase.

There is no reason why Smart electrical systems could not follow this example. Updates in the distribution network require moving to the same infrastructure used by digital advertisement players. Therefore, there is little reason to reinvent consumer data extraction tools. Most importantly, this strategy enables leveraging tried and tested methods for handling data privacy and cybersecurity.

This document explores the beginning of this journey. It investigates the best strategy for understanding consumer behaviour and its relationship with the energy distribution system. The focus is developing a reliable tool for forecasting household demand as the first step toward creating a fully modernised digital communication medium.

# II. Related Work

The research starts by disambiguating the acronyms involved in developing such systems. It must interact with newer iterations of older concepts, like Fog Computing. Moreover, it must also provide guidance on terminology that seems well-defined but still suffers from conflicting details. J. Kampars et al., 2021, provide a comprehensive survey of the technological landscape. It explores the overlapping concepts between the Internet of Things (IoT), Edge Computing, and Cloud Platforms.

Another relevant survey is B. Sudharsan et al., 2021. The authors present several options for energy demand forecasting on embedded platforms. While this document cannot reach the same level of depth, it is essential to understand the practical implications of connecting and distributing forecasting models across distributed networks.

Finally, A. T. Eseye and M. Lehtonen, 2020, discuss the standard machine learning techniques employed for this project. The document performs a comprehensive analysis of Artificial Neural Networks in roughly the same terms stated by this research's objectives. It also provides parallels with competing implementations using Support Vector Machines.

# III. Methodology

Multiple academic disciplines are involved in proposing a solid implementation for an ambitious Smart Grid system. A fully fleshed-out plan requires much more than examining candidate machine learning implementations. However, this study cannot spare the resources necessary to perform due diligence in all needed aspects. Therefore, the scope will remain on the study of the decision-making process.

This focus on software implementation implies a primarily quantitative exploration of resource consumption. Since there is no candidate hardware platform, there are no expected limits. The scenario involves extra tasks such as defining credible baselines so the researcher can reach plausible conclusions.

Nonetheless, some theoretical guidelines can form a comfortable guardrail. We are discussing a time-series forecasting problem that fits a well-established statistical mould. There are clear baselines for how effective a model can perform. Most importantly, this framework allows experiments to quantify each proposal's effectiveness level.

The intersection between computer science and statistical model evaluation defines the point where automated decision-making becomes a viable option. For example, inspecting the behaviour of a dataset will allow us to compute the error margins of doing nothing. Assuming that previous readings are close enough to the subsequent measurement, there is little reason to waste computing cycles.

With this baseline, we can look for algorithms to improve this score. This step lets us put computational costs in context and define a price-per-point accuracy boost. This evaluation can rely on well-established time and space complexity analysis. There is no point in discussing strategies for proper assignment of responsibilities of components in the distributed architecture of a Smart Grid system if we do not know the costs involved in each stage.

Space complexity will inform the memory capacity required to build adaptative models and evaluate decisions. The size of these artefacts also impacts the required network bandwidth to operationalise an appropriate response time. Finally, it will inform the storage requirements of the platform. Keeping a database of consumer profiles may allow more sophisticated inferences, but it can also be prohibitively expensive.

Time complexity will dictate the level of responsiveness of the system. We discussed moving from a slow-moving, centralised forecasting plan to a distributed, fine-grained, responsive one. Generating and updating models can be a bottleneck that restricts response times. But also, forecasting times can hamper the agility of the solution.

Finally, the research will investigate the gains accrued by the automated decision model. It must look for relevant statistical criteria to quantify the confidence in results. The stationary baseline model, while trivial, has its structural capabilities that might not differ much from a complex estimator. An even worse case would be if the model could not reach the same reliability level as the one presented by the baseline.

# IV. Candidate Models

Chart

Description automatically generated with low confidence

Figure - Household consumption, daily seasonality

Household consumption injects a fair amount of human unpredictability into the problem. Some aspects can be reasoned at a coarse-grained level. For example, daily seasonal patterns are undeniable (Figure 2). This type of information drives decision processes in several disciplines, directly impacting the design of well-established distribution supply chains. These seasonal patterns are essential for supply-side planning in the energy distribution space.

This proposal explores a different pathway, more in line with the online connectivity patterns proposed by the modern Smart Electrical grid. The argument is that traditional seasonal windows do not capture single household demand behaviour with enough resolution to empower residents to drive energy conservation decisions. They are better suited to a centrally managed system, like the ones in the current Energy Distribution control planes. The main idea of this article is not to scrap the current model but refine it to include a more decentralised component that consumers could interact directly through their daily routines.

Unfortunately, capturing consumption patterns in this resolution is not a trivial matter. Human behaviour has a notoriously chaotic component that challenges conventional modelling. For this reason, the research explores non-mainstream applications. It borrows from modern digital marketing platforms and their successful applications of non-linear models such as Artificial Neural Networks (ANN).

Before discussing possibilities, it is crucial to place some guidelines that constrain available options. ANNs are a popular academic field with a prolific community. The number of possible architectures increases every year. Nonetheless, the proposed solution cannot reasonably expend the same resources as state-of-the-art Transformer solutions employed in large language models. Instead, it goes the opposite way, favouring lightweight structures transmitted over the network.

However, the other side of the spectrum does not provide a straightforward option either. While Feed Forward Neural Networks (FFNN) can be easily embedded in simple devices, they might not provide adequate structural support. The problem lies in treating an extrapolation problem that might not have a datapoint available to map its construction as a data interpolation problem if the process is not modelled at the required levels of sophistication.

Energy consumption forecasting requires a structure closer to a time series analysis. Patterns of neighbouring measurements inform the behaviour of future ones. FFNNs treat all data points in the same fashion if no structural inference mechanism is modelled explicitly. This research must rely on a decision framework that can adaptatively map sequential structures according to temporal invariances. Therefore, we will explore a few candidates between the simplest architectural choices and strong contenders with a track record of capturing time displacement patterns. The selected candidate must be simple enough to fit an embedded system but capture enough complexity to provide reliable forecasts.

## A. Convolutional Neural Networks

One of the simplest ways of looking for invariant behaviours across time is the application of adaptative filters. Instead of treating neurons as weight placeholders in a non-linear system, this architecture groups these structures into kernels and creates a lower resolution matrix on this convolution layer. This configuration allows the translation of a neighbouring pattern (a slight signal increase followed by a sharp drop) into a meaningful information unit to be manipulated by an upper layer. Since the experiment deals with a single signal, the best representative is the Single Dimension Convolution Neural Network (CNN1D). In this case, the kernel is a subset of the sequential input widow.

## B. Gated Recurrent Units

GRUs provide an extra layer of refinement on top of the standard Recurrent Neural Networks (RNN). These ANNs specialise in mapping sequential patterns found in their inputs. Standard RNNs use the composition of sigmoid units to capture context information in the input by applying the same structure to each input sequentially. Their limitation is their tendency to “forget” sequence members and exceptionally long input vectors. GRUs filter irrelevant members by adding a forget logical gate before proceeding with the signal propagation. Like the CNN1D, it will take neighbouring sequence values into account, thus creating a similar pattern grouping mechanic.

## C. Long-Term Short-Term Memory

LSTMs are the most sophisticated of the RNNs. This architecture's rationale is to construct context masks that propagate along with the inputs adaptatively. This structure allows the model to select arbitrary subsets of the input sequence during the prediction stage, thus enabling a sophisticated composition of simple pattern classification filters that rely on neighbouring indexes.

# V. Implementation

A picture containing diagram

Description automatically generated

Figure - Decision Support System, experiment summary

One striking contrast when considering the departure from centralised control systems toward more collaborative frameworks is the importance of concept drift. Large-scale planning dilutes individual shifts in consumer behaviour since it is concerned with putting together a macro view of the distribution system. In this case, significant perturbations on energy flow require events with widespread impacts, like events that capture the attention of a whole region (popular televised sporting events or season finales of shows on a schedule). Direct consumer engagement requires a much more individual approach. Residents will see little incentive to use a solution that is not responsive to their actions.

For this reason, this proposal must favour individualised decision-making models. At this point, some architectural considerations can rationalise this problem statement into a workable solution. Big datasets required to train these models might not conform to the processing restrictions of a more modest meter. However, this requirement might not be fulfilled on-site but in a fully distributed system (the control plane). The system’s design can accommodate delays in a model update if the existing solution performs at reasonable levels.

Following this premise, we can add three restrictions to the implementation:

1. The model must fit the embedded system after training.
2. The model must be responsive given the hardware restrictions of the embedded system.
3. Training times offsite must be fast enough so concept drift does not degrade model response.

Point 3 should not become an afterthought. Training millions of simple models on large datasets can become intractable even for robust, scalable platforms.

Our choice in working with ANN also introduces additional concerns that act as both benefits and sometimes as system constraints. Figure 3 provides a visual summary of this exercise:

1. The Global Active Power signal must be down-sampled to a workable discrete measurement, represented by the white circles in Figure 3.
2. The output of the predicted model will also represent the same type of discrete measurement of the same interval size, white diamonds in Figure 3.
3. The model will leverage a set of hidden units of one of the candidate architectures discussed in the previous section, white squares in Figure 3.

Although only one hidden layer is shown in Figure 3, the actual implementation might be more complex. For example, adding some noise to the system with a Dropout layer usually produces a more robust training phase (Srivastava N. et al., 2014). Additionally, some architectures, like CNN1D, require extra manipulation, such as model Flattening, to provide the expected output type.

Modern ANNs are highly modular and support the level of interchangeability required by this experiment. The source code was written using Python and the Tensorflow library (M. Abadi et al., 2015). The main reason for this choice was the possibility of progressing this research using Tensorflow Lite and directly embedding the trained model into a few candidate hardware platforms (B. Sudharsan et al., 2021).

Chart, line chart

Description automatically generated

Figure - Model Applicability, hypothesis test

Even though there is a significant amount of time spent on analysing the computational requirements involved in this proposal, the model’s applicability is the paramount concern. A highly scalable, computationally efficient, inaccurate solution will not warrant the resources spent on deploying it, even though costs were minimised. In this case, the proposal must provide the required level of observability. Figure 4 provides a visual summary of the data collected for validating applicability. Every prediction records the baseline (persistence model) used to compute the error accrued, assuming that current readings will not change (Figure 4, left). The model’s output provides the de-facto estimation of the future reading and its residual error (Figure 4, centre). Finally, we have the ground truth, or the labelled test data used to define the target for both error margins (Figure 4, right).

A picture containing text, clock

Description automatically generated As follows, our hypothesis for using a machine learning model hinges on the qualitative gains in terms of minimising the residual errors in comparison with not using any type of estimation, but also on the quantitative assessments of the structural differences between both cases (Figure 5).

Figure - Model Applicability, comparison

Chart, line chart

Description automatically generated

**W(1) P(1857):** MSE (baseline) 5.58394734

MSE (cnn1d) 1.44245937

p-value (student T) 0.00000003

Figure - Model Applicability, experiment's output

Figure 6 provides the output generated by the experiment. At training window one prediction number 1857 of this batch, 48 5-minute readings (blue line) were given as an input for prediction. The baseline (red line) used the last reading as the result of every position (the persistence model, or baseline). There were 13 5-minute interval predictions (orange line) adding up to a little over 1 hour (ANN model output). Both cases were compared with the 13-ground truth points (cyan line) with significant differences in error margins favouring the estimator. According to the student-t statistic, there are enough structural differences between the baseline and the estimator to accept that they are not the same model with high confidence (p-value).

# VI. Discussion

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure** | **Train T. (s)** | **Pred. T. (s)** | **Size (KB)** |
| FFNN | 28.500 | 0.205 | 13.254 |
| GRU | 1174.000 | 1.760 | 663.670\* |
| LSTM | 1731.000 | 1.580 | 768.541\* |
| CNN1D | 17.500 | 0.166 | 11.573 |

Table - Time and Space analysis

The starting point for a technical discussion is comparing candidate models and their applicability to the problem statement. Table 1 showcases the results of all ANN architectures' time and space complexity analysis. Most research in machine learning models focuses on the residual analysis of a model’s error when discussing regressions. For this research, the hardware constraints imposed by embedded systems demand a primary focus on processing power.

For example, trained Tensorflow models can be shrink-wrapped to fit this scenario with Tensorflow Lite. However, not all architectures are compatible with this transformation. GRUs and LSTMs rely on sophisticated data structures that cannot be trivially ported to an embeddable counterpart (Table 1, Size column). Even if the costs of sending a trained model over the network are disregarded, its size is directly correlated to its memory consumption. Using an ANN model to forecast energy demand is a CPU-intensive process, after all. There are bespoke hardware optimisations in the form of GPUs and TPUs that might not be readily available on simpler computing platforms.

According to the time and space criteria, the RNN-based models should be discarded. Both had consumption scores at least one order of magnitude larger than their simpler counterparts. The choice between FFNNs and CNN1Ds is less obvious. However, the latter has a slight edge in terms of model size.

The analysis of the candidate’s time complexity followed the same pattern. Given the CPU-intensive nature of these computations, their level of sophistication is reflected in both training and prediction times. The CNN1D was the clear winner in all criteria. Table 1 does not provide an accurate picture of the prediction times (Pred. T. column) since a candidate hardware platform was not used. However, training times (Train. T. column) provide a much more accurate simulation. In this case, we can speculate how long would it take to generate individual predictions for a large population. Table 2 presents these figures assuming a target audience of 100 thousand households and parallelism of 4. The choice between roughly 500 days (LSTM) versus five around days (CNN1D) puts options in more concrete terms.

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure** | **Train T. (s)** | **Serial (h)** | **Par. x4 (h)** |
| FFNN | 28.500 | 791.666 | 197.916 |
| GRU | 1174.000 | 32611.111 | 8152.777 |
| LSTM | 1731.000 | 48083.333 | 12020.833 |
| CNN1D | 17.500 | 486.111 | 121.527 |

Table - Training times

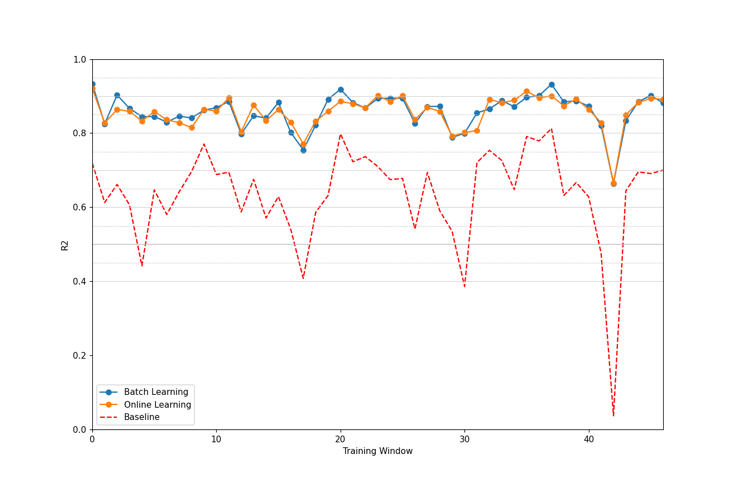
Once discussions about resource requirements are finished, the analysis must focus on the common concerns of machine learning systems. Figure 7 charts the CNN1D’s predictions across the whole data set. Since the chosen structure is a forecasting model that takes a sequence of 48 5-minute intervals and maps it into a series of 13 5-minute intervals, each point represents the expected and predicted output sequence averages.

Chart

Description automatically generated with medium confidence

Figure - Consumption forecasting

Figure 8 presents a more fine-grained view of the whole experiment. In this case, we see the consolidation of the R2 metric of each training window (3 months). This visualisation attempts to answer two questions. First, what are the gains of using a machine learning model (blue and orange lines) instead of just using the last consumption measurement during the forecasted period (red dashed line, the baseline)? Afterwards, which training strategy is the most effective: creating a new model from scratch on each training window (batch learning, blue line) or updating the existing model with the latest measurements (online learning, orange line)?



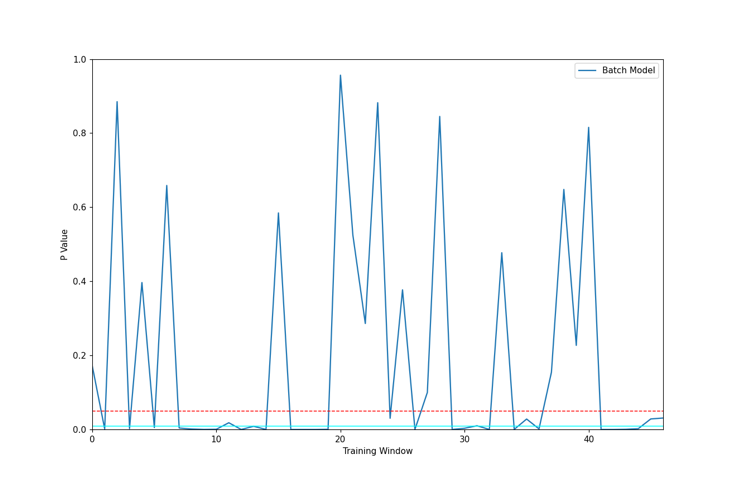
**Online:** mean(0.8560), gain(0.2208), var(0.0022)

**Batch:** mean(0.8583), gain(0.2232), var(0.0023)

Figure - Model gains over Baseline

The results showcase an average gain of 22% on the R2 metric. Moreover, the reported level of accuracy is aligned with current findings in the literature. The low mean-variance across the residual error is another fortunate finding since it signals that the models are not converging to these scores on average but make individual predictions that are significantly off the mark.

The next step is validating the hypothesis that the proposed model is structurally different from the baseline. The chosen process compared each prediction’s mean and variance using the z-score. A test data split of 10% was used at each training window to compute this work unit’s residual losses. The repeated last measurement stood in for the forecasts of the baseline for the same test cases. Both samples’ scores supplied the raw material for generating the p-value. The median value of this metric for all training windows could finally be compared to the gold standard of 0.05.



**Batch Model:** median p-value(0.0085)

Figure - Hypothesis testing: Batch x Baseline

Figure 9 provides the summary of the evolution of the p-value across the training windows for the Batch training strategy. On the positive side, the median score points to a high likelihood of rejecting the hypothesis that both the baseline model and a CNN1D trained in batches are structurally similar. The dashed red line marks this test's 0.05 threshold, the gold standard. Still, on several training windows, we encounter unclear behaviours with much higher p-values.

Chart

Description automatically generated with medium confidence

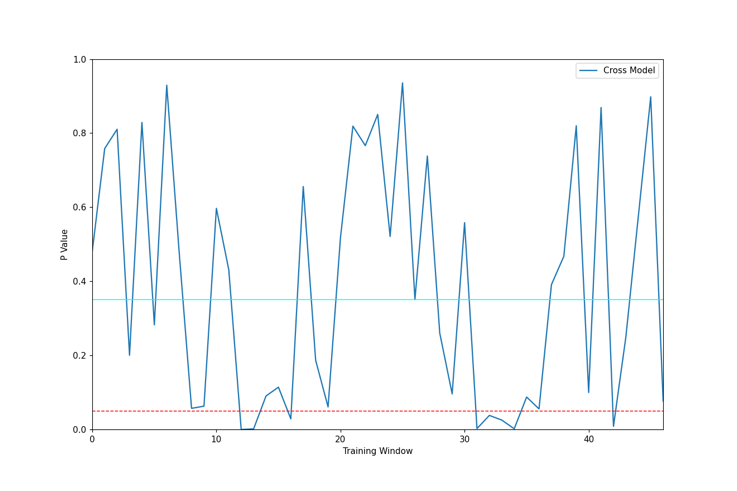
**W(1) P(930):** MSE (baseline) 0.00029548

MSE (cnn1d) 0.00025535

p-value (student T): 0.01607697

Figure - Model and Baseline similarities

One explanation for these discrepancies can be found in Figure 10. It showcases a period of low variation in consumption. In this case, it is expected that the mean squared error reported by baseline and CNN1D are drawn from the same neighbourhoods. Consequently, both models will be structurally similar. It is a key finding about the behaviour of the forecasting model as a whole: its gains in minimising residual errors are linked to periods of high variance in metering. In scenarios where consumption remains stable, we can expect degradation in overall predictive power.



**Cross Model:** median p-value(0.3508)

Figure - Hypothesis testing: Batch x Online

Finally, we perform the structural comparison between the Batch and Online training strategies. Figure 12 supplies an inconclusive picture. The hypothesis of both models being significantly different has much less consistent behaviour across training windows. Therefore, the Batch training strategy has a slight edge since it does not require sophisticated state keeping on the control plane and can be generated ad-hoc but provides roughly the same p-value score as Online when compared to the baseline.

# VII. Conclusion

This document outlined a possible strategy for enabling demand-side response in the context of smart electrical grids. It builds on two foundational aspects of a broader technological landscape. First is the move from a central control system that relies mainly on information broadcast to manage consumption toward a distributed, fully connected platform. Second, the opportunities for engaging with residential consumers directly, borrowing successful applications of machine learning systems on state-of-the-art marketing solutions.

The discussion surveyed the applicability of this toolkit to current IoT applications and, most importantly, smart residential meters. To that effect, the research focused on finding the most cost-effective algorithm in terms of both size (lightweight and distributable over the network) and processing times (to reduce resource consumption on the embedded side of the solution). ANNs provide a comfortable level of modularisation to enable solution interchangeability without affecting data preprocessing or output formatting. Out of all candidate architectures, CNN1D was the best choice in both criteria.

Next, the document compared the residual errors of the estimator against the persistence model. This baseline stood in as metering systems that report the current consumption profile. The results indicated an overall gain of 0.22 points in the R2 statistic, reaching a mean value of 0.85 on a test dataset spanning four years.

A more rigorous appraisal of the hypothesis presented some challenges. A stationary baseline provides meaningful insights into scenarios when constant consumption variation is the norm but acts as a confounding factor when consumption stabilises. This is an intractable problem since we need an estimator that captures signal shifts and stabilisation. Nonetheless, the experiment detected enough difference using this imperfect approach to highlight that the model did not perform like the baseline in well over 50% of the dataset.

Another research question was about possible operational gains enabled by different training strategies: batch and online training options performed at roughly the same level of accuracy. Moreover, the z-statistic test showed no apparent structural differences between these candidates.

This experiment was not an exhaustive exploration of the proposed scenario. One possible next step is to validate findings in candidate hardware platforms. Another serious issue that remains untouched is the *“cold-start problem”*. The reported scores used 3-month intervals for training, and the model uses the last 4-hour metering data to generate a 1-hour prediction. Newly installed meters will not have any historical data to work with. To cover these shortcomings, they must rely on additional strategies, such as swarm intelligence. As it stands now, little has been covered by the literature in this respect.

References

A. T. Eseye and M. Lehtonen, "Short-Term Forecasting of Heat Demand of Buildings for Efficient and Optimal Energy Management Based on Integrated Machine Learning Models," in IEEE Transactions on Industrial Informatics, vol. 16, no. 12, pp. 7743-7755, Dec. 2020, doi: 10.1109/TII.2020.2970165.

B. Sudharsan et al., "TinyML Benchmark: Executing Fully Connected Neural Networks on Commodity Microcontrollers," 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 883-884, doi: 10.1109/WF-IoT51360.2021.9595024.

J. Kampars, D. Tropins and R. Matisons, "A Review of Application Layer Communication Protocols for the IoT Edge Cloud Continuum," 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), 2021, pp. 1-6, doi: 10.1109/ITMS52826.2021.9615332.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Georges Hebrail (georges.hebrail '@' edf.fr), Senior Researcher, EDF R&D, Clamart, France

Alice Berard, TELECOM ParisTech Master of Engineering Internship at EDF R&D, Clamart, France

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software is available from tensorflow.org.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929-1958.

**Rodrigo Gonçalves Leite** (PG, University of Liverpool)

1. Submitted as part of the coursework required for the PPGET Pos-graduate course in Machine Learning, UFF-RJ. UFF [↑](#footnote-ref-1)