Progressive Short-Term Household Load Demand Forecast: A Dual-Model Approach

João Silva
DETI
Universidade de Aveiro
Aveiro, Portugal
joao.npsilva@ua.pt

Paulo C. Bartolomeu
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal
bartolomeu@ua.pt

***Abstract*—The rapid evolution in the most recent years has raised power consumption in today's world. Alternative energy sources are gaining increasing importance and, with that, so are the studies on how to better utilize renewable energy. This work was conducted within the scope of a project that aims at developing an intelligent energy management system for residential buildings. One of the requirements for said system is to accurately predict the overall household energy consumption. The objective of this study is to utilize machine learning techniques to develop a model that can learn the consumption habits and patterns of a household in order to forecast its future electrical load. Firstly, a study comparing Support Vector Regression (SVR) and LSTM (Long Short Term Memory) neural network is employed. It was found that SVR slightly outperforms LSTM when the training and prediction is performed utilizing only one house. Secondly, utilizing a dataset with multiple households and LSTM, a generic model is built. The generic model will be utilized in the early stages of the final system, while enough information about the specific house is being collected, in order to train a specific model with SVR.**

***Index Terms*—energy consumption, load, forecast, machine learning, prediction, short-term, hourly, residential, household**

I. Introduction

According to the UN, two thirds of the world's population is projected to live in urban areas by 2050 [1], which would inherently increase the demand for energy in a particular region.

There is an increasing effort to keep energy distribution sustainable and to find ways of reducing its price. This study in part of such a project, to build a energy management systems designed to aid in reducing the electric bill of households. One input that can be of value to an algorithm like this is the future household demand. Certainly, the mentioned task can benefit from an accurate energy prediction to better perform load scheduling.

Energy consumption within a household will frequently follow a pattern, especially when it comes to periods of high consumption (in most cases during the day and afternoon) and low consumption (in most cases during the night). A family can have repeated behaviors and, probably, somewhat constant lows and peaks of energy usage. Historical energy usage records and Machine Learning (ML) can be utilized as a way of building knowledge on the consumption habits of a household and based on the most recent consumption, predict the near future energy demand.

The objective of this study is to utilize machine learning techniques to build a model that can learn the consumption habits and patterns of a household in order to forecast the future electrical load of that particular household. This is called short term time series forecasting: a technique in machine learning, which analyzes data and the sequence of time to predict future events. It is a short term forecast due to small time step that is being predicted (one hour). This means that the records of the train data were collected on hourly intervals and the model will predict the consumption of the next hour.

Machine Learning models usually need to be trained with historical data, which in some cases might not be available and needs to be collected first. Additionally, if the amount of data is small, the machine learning model cannot be expected to perform well. This might not be a problem for research purposes. However, when trying to build a product that can be sold, the user will expect results right away. To overcome this issue, this study also proposes to build a generic model that can predict the consumption of any household with good enough precision and no training. This generic model will be trained using several house's historical data. On a real world scenario, when prediction future house consumption of a household, this model is meant to be used in the early stages of its deployment, while not enough data has been collected to train a machine learning model specifically for that household. Once the performance of the specific model surpasses the generic model, the former replaces the later in forecasting household energy consumption.

This study is divided into three parts. Firstly, a description of previous studies in this area is presented. Based on the literature review, a set of machine learning algorithms are chosen and present in the next section. Then, utilizing a dataset with multiple households, a specific and a generic model are built. The specific model is trained with data relative to a single house. The generic model will be utilized in the early stages of the final system, while not enough information about the specific house is collected to train the specific model.

II. LITERATURE REVIEW

Certain independent factors are expected to have an impact on energy consumption, such as the relationship between outdoor temperature and electricity demand. In Nordic countries, this relationship is easily visible, especially during cold seasons, as the use of space and water heating appliances increases.

The study in [2] analyses the problem of energy consumption prediction at single household level. The proposed model consists on a Support Vector Regression (SVR) (a version of SVM for regression problems) with both daily and hourly data granularity. The model was trained with past hourly electricity consumption (obtained with smart meters), weather conditions, temperature, humidity, hour of the day, day of the week, month, season and electricity time of use (TOU) rate.

Most of the 15 households used in this research had a good mean average percentage error (MAPE) value for daily forecast, with the best value being 12.78% and the worst 34.95%. The average MAPE was of 22.64%. The accuracy for hourly forecast suffers from high fluctuation, with the best value being 23.31% and the worst 64.38% (average of 37.3%). In addition, results show that five households, whose consumption patterns were more unpredictable, benefited from random sample splitting, in contrary to the other households, with similarities in their consumption patterns over time, that performed better with time-based training and testing subsets splitting.

The authors of the study in [3] propose a long short-term memory (LSTM) recurrent neural network (RNN) based framework for short-term load forecasting for individual residential households.

The proposed approach was compared against other ML models (backpropagation neural network (BPNN), k-nearest neighbour (KNN) regression, extreme learning machine (ELM) and input selection combined with hybrid forecasting (IS-HF)). In addition to individual household forecasts, the authors also tested the models for aggregated household prediction. The LSTM model was the best model for individual (MAPE: 44.06%) and aggregated forecast (MAPE: 8.58%). The difference in MAPE values shows the difference between individual and aggregated household prediction as the aggregated load smooths the profiles, while individual households are prone to unpredictable peaks.

Electrical consumption is highly related with factors such as temperature, day of the week, etc. However, one cannot disregard the correlation that residents' behaviour has with energy consumption throughout the day, for example, events like shower and laundry will often occur around similar times during the several days.

The authors of the study in [4] included appliance measurements in the training data of a LSTM recurrent neural network to tackle this issue. The objective is to better classify the lifestyle of the household, improving the interpretation of volatile peaks. Out of 19 appliances, 6 were selected as the most important ones and used to train the model, they were: clothes dryer, clothes washer, dishwasher, heat pump, television, and wall oven.

The performance of the model was tested against a feed forward neural network (FFNN) and a K-nearest neighbors (KNN) algorithm, also, the models were trained only with total house measurements and, for comparison, total measurement plus appliances. The proposed LSTM outperformed the FNN and KNN algorithm and in all cases. Models that were trained with appliance data outperformed house measurements only.

The study in [5] compares energy forecasting techniques that allow individual or aggregated prosumers to anticipate their future energy consumption. To this end, a set of models were compared. One of the tested models is a time-series based linear regression model, Seasonal Auto regressive Integrated Moving Average (SARIMA). Secondly, a feedforward neural network, fully-connected multi-layer perceptron (MLP). In addition, a LSTM neural network and a support vector regression (SVR) model were also compared. An ensemble model, containing all other modules in addition to one that combines their outputs was also implemented.

Regarding aggregated consumption, results shown that the ensemble model performed the best (14.4% mean absolute percentage error (MAPE) with weather data), followed by the SVR model (interestingly, SVR performed better without weather data: 15.4% MAPE without weather data). As for the individual households, the SVR model, despite still having a not so good accuracy, performed better than the other models (53.45% MAPE), including the ensemble model (68.51% MAPE) and the LSTM model (100.84%).

Forecasting household energy consumption is a difficult subject. Unlike the forecast of the electrical consumption in large buildings or aggregated city loads, where the energy demand curve is smooth due to unpredictable events not having a great impact in the overall picture, single family homes have very volatile peaks and minimums in their energy consumption. As presented, SVR is a not so complex machine learning technique based on support vector machines that is used in these problems and presents a realistic way of predicting future energy demand. LSTM is a more complex neural network based approach, however, studies that compare these approaches, in relation to energy consumption prediction claim that SVR achieved a better score than LSTM. Both approaches rely on historical data to train a model what hinders their ability to work when no past records are available.

III. ML TECHNIQUES

According to the previous exploration on the techniques used to forecast electrical loads, Support Vector Regression (SVR) and Long Short Term Memory (LSTM) are the most commonly used techniques to solve this problem.

Support Vector Machines (SVM) [6] is a supervised machine learning technique used for classification problems. SVM utilizes a kernel and an optimizer algorithm. The kernel transforms the data, increasing its dimensionality. This aims at transforming non-linear data into a high dimensional space where data can be linearly separable. The optimizer finds the hyper plane (decision boundary) that separates data. SVR is version of SVM that is utilized for regression problems. Here, the model tries to find the best fit line by reducing the error between observed and predicted values. The model was implemented with the scikit-learn python library [7]. The chosen parameters were set to default, which utilizes the Radial-Basis Function (RBF) kernel, which is used for non-linear problems.

Neural Networks (NN) or Artificial Neural Networks (ANN) are a machine learning technique that mimics the behavior of the human brain [8]. This model is used to model complex patterns by passing the data through layers of neurons. Neurons will receive features from the training set and apply an activation function, the result is outputted to another neuron and so on. The output of the neural network represents the combined input of all neurons.

Recurrent Neural Networks (RNNs) implement feedback loops where information from prior inputs is used to influence the current output. Basically, the information a layer had in the previous step is remembered by a memory function. This method is widely used with sequential data.

However, it is difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This occurs because the gradient of the loss function decays exponentially with time (vanishing gradient problem) [9]. Long Short Term Memory (LSTM) networks are a type of RNN that use special units which include a memory cell that can maintain information in memory for long periods of time, solving the vanishing gradient problem.

In this paper, the architecture we selected contained two LSTM layers. The first, is called the input layer and contains neurons equal to the amount of inputs (number of features * past window, where past window is the amount of historical records the model will look at in order to predict the next value), in this case, 72 neurons. The second layer is called a hidden layer. The number of units in this layer is usually between the size of the input and size of the output layers [10]. Therefore, the number of neurons was set to two thirds of those of the input layer, more exactly, 48 neurons. An output layer is also part of the architecture. This last layer provides the output of the model and, since it is only one value (energy consumption), only one neuron is required. Every layer neurons' are fully connected to the previous neurons. A visual representation is present in Fig. 1. The LSTM neural network was implemented using keras library [11].

The exploration of state of the art articles confirmed that LSTM is widely utilized to make predictions in time series problems such as the forecast of household consumption. However, some papers also suggested that SVR is a very competent technique that can event outperform LSTM [5]. Studies also shown that models gained from additional features like weather and individual appliance consumption [4].

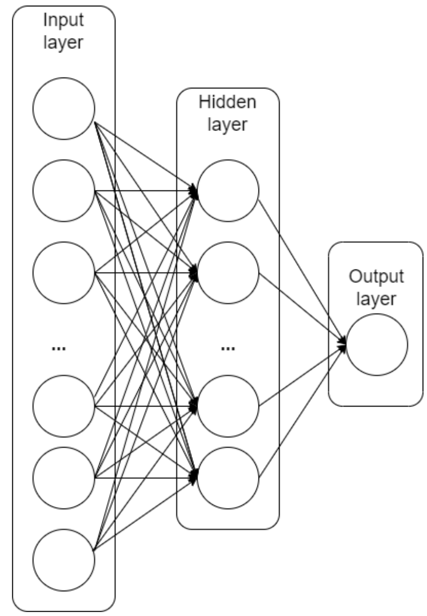

Fig. 1: LSTM Neural Network Architecture

IV. Generic Model

Machine learning models are usually tested by splitting the data into train and test sub-datasets. This means that the test results were obtained when the model already had many samples, e.g., at least a few months of historical data. On a real world scenario, the user will expect results from a given machine learning model right away. Therefore, it cannot afford to wait for the model to have sufficient historical data before start providing adequate forecasts. At the same time, it should not be expected from a model with little to no training data to achieve acceptable results.

In this section we present the option of having a generic model, which does not require the knowledge of the prior behavior of a specific house to predict future electrical consumption values. This model was trained with the load demand data of several houses, so that it can represent the consumption behavior of typical households. This model was utilized in the early stage of deployment, when not enough data has been collected to generate a specific model. Once the specific model outperforms the generic one in terms of its accuracy, the later is no longer used. The specific model will be trained solely with the specific data pertaining to the household. A brief test is also performed, comparing the LSTM and SVR models, to understand which technique is better, when it comes to training exclusively with one house and making predictions for the same house.

A. Dataset

The dataset [12] contains energy consumption readings for a sample of 5567 London Households that took part in the UK Power Networks led Low Carbon London project between November 2011 and February 2014. The dataset contains only information about the date and power readings. Firstly, the data was separated into multiple files, one for each house. This was necessary since several users were mixed in the same file and, often times, the same user had reading across multiple files. Also, since the original dataset contained reading with the time steps of half an hour, all the records were aggregated into one hour intervals. The dataset was also subject to a data treatment process, where we made sure the households with empty records would not be used. Additionally, this process also ensured that all the records were continuous and spaced with the same interval.

B. Metrics

In order to evaluate the performance of the resulting models several metrics were used, as described below.

MAPE: Mean Absolute Percentage Error represents accuracy as a ratio.

$$MAPE = \frac{100}{n} \sum_{j=1}^{n} \left| \frac{Actual_j - Forecasted_j}{Actual_j} \right| \quad (1)$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

WAPE: Weighted absolute percentage error equally penalizes for under-forecasting or over-forecasting, and doesn't favor either scenario.

$$WAPE = \frac{\sum_{j=1}^{n} |Actual_j - Forecasted_j|}{\sum_{j=1}^{n} |Actual_j|} \quad (2)$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

MAE: Mean Absolute Error expresses the absolute error between observations and predictions.

$$MAE = \frac{\sum_{j=1}^{n} |Forecasted_j - Actual_j|}{n} \quad (3)$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

MaxAE: Maximum Absolute Error represents the maximum error between one observation and one prediction across all observations.

$$MaxAE = max(|Forecasted_j - Actual_j|) \quad (4)$$

where *Actual* represents the actual consumption value, *Forecasted* represents the predicted value and n corresponds to the number of observations.

C. Specific Model

The specific model will be trained with the load demand data of a single household. The model will then be used to forecast the power demand of that particular household. The LSTM and SVR algorithms were compared regarding their ability to train exclusively with data from one house and making predictions for that same house. The assumed window of past values is 24. Hence, the models will receive the data from the last 24 hours to make a prediction. The models were also compared with a baseline model. The baseline prediction is the same as the last observed value (current value).

Results in Table I show that the SVR algorithm achieved a better performance in terms of MAPE, WAPE and MAE. Therefore, SVR will be used has the specific model. Interestingly, the LSTM model achieved a better maximum absolute error in almost all the test houses. The baseline, which predicts the last observed value, scored far worst results in Table II, the only exception being the maximum absolute error of the first test house.

D. Generic Model

As discussed, the objective of the generic model is to predict the energy consumption while historical data is still being collected to feed a ML model trained with data from that particular household.

Previous research has shown that SVR is the best model when it comes predicting household energy usage on one specific house. The SVR model in question is, however, implemented with the *sklearn* library and does not support incremental learning. This means that all the training must be performed in one go, with all the training data stored in memory. Hence, the model is not be able to continue training incrementally. Additionally, the SVR model is quite slow for larger amounts of data, making it impractical to build this generic model. Since the previously tested LSTM model, implemented using the keras library, performed decently and does support incremental learning, it was chosen to be the base for the generic model, which is to be trained with several households.

Initially, 500 users where selected at random to be the training pool for the generic model. However, the training process was shown to be very time consuming, with the model taking about 10 minutes per house. Therefore, only 50 households were used to train this generic model. This decision will not seriously harm the model since in further testing it was found that more train data beyond a certain point will not necessarily improve the performance of the model. However, since the model supports incremental learning, it is possible to keep training the model. This feature will also be used to keep training an instance of the generic model with the same data that will be used for the specific model. This way, while the specific model is being trained, the generic model will also improve its results for a specific household. The generic model will use the last 24 hours of consumption, hour and weekday values as features, similarly to the specific model.

The training process requires feature scaling. This process aims at centering at value zero and helps the machine learning model achieving better results. For this purpose the *sklearn* library was used to instantiate a standard scaler. This scaler works with the mean and standard deviation of each feature and all the training dataset should be considered to retrieve these values. Therefore all the 500 houses were used to build

House	SVR				LSTM			
	MAPE	WAPE	MAE	MaxAE	MAPE	WAPE	MAE	MaxAE
MAC000205	0.36	0.38	0.19	3.18	0.51	0.44	0.23	3.02
MAC002086	0.20	0.21	0.04	0.81	0.24	0.24	0.05	0.77
MAC000387	0.30	0.35	0.17	2.38	0.36	0.36	0.17	2.47
MAC000276	0.24	0.23	0.15	1.17	0.27	0.24	0.16	0.96
MAC001251	0.4	0.4	0.28	2.58	0.5	0.42	0.3	2.36

TABLE I: SVR vs LSTM performance in 5 test Houses

House	BaseLine			
	MAPE	WAPE	MAE	MaxAE
MAC000205	0.48	0.5	0.24	2.44
MAC002086	0.31	0.34	0.07	1.04
MAC000387	0.46	0.43	0.22	2.5
MAC000276	0.27	0.27	0.18	1.24
MAC001251	0.5	0.47	0.32	2.81

TABLE II: Baseline performance in 5 test Houses

this scaler even though some of the houses that contributed to the scaler did not actually train the model.

E. Generic Model and Specific Model Simulation

To emulate the behavior of both models, a simulation was performed utilizing 5 randomly selected households. This process will iterate over the records of each house being tested, emulating the passage of time, and evaluating the performance of both models as more and more historical data is available.

Since the specific model will only be trained with data from one specific user, and since that, at the beginning of the experiment there is no prior data from this user, the specific model is bound to not achieving good results at start. In a real world scenario, it would be possible to retrain the specific model every hour, when a new record is created. However, for time reasons, the model was only retrained every week. This means that on the first week, only the generic model will be available. On theory, with the passage of time, the specific model should outperform the generic model, since it is being trained for the specific user that is being tested.

For each hour, both specific and generic models will make their predictions on the next energy consumption value. Each record is stored in a dataset that contains the observed values in the simulation. This dataset is used to train the specific model and to provide the latest records for the models to make predictions (since the models take into account the last 24 steps of data to predict the next value). Every week the specific model will be retrained with the observed reading of that week. Since the generic model supports incremental learning, an instance of this model will also be retrained with the same data. The objective is to understand if the generic model is good enough to be a substitute model in the early stages and, also, how long does it take for the specific model to start providing valuable predictions.

A product based on this approach will harness the all-encompassing nature of the generic model and use its prediction in the early stages of the deployment. This model will then be replaced by the specific model, that was trained especially for that household. This replacement is performed as soon as the system recognizes that the specific model is making less errors in its predictions than its generic counterpart. To attain this, the MAE metric was chosen, comparing the last 72 hours of predictions from both models. As soon as the specific model has a smaller MAE over the last three days, the generic model is replaced and is no longer used.

F. Results

The results presented in the Table III show that the specific model achieves better results in all the metrics, except in the second test house, where the specific model ended up with higher maximum absolute error. Despite this, it is safe to say that the specific model has a better performance than the generic model, as expected. It is worth mentioning that these metrics encompass the whole house dataset, which means that the first week (where all the specific model's predictions were zero) and the following days (where the specific model did not have much training) also count for these results.

To remove this factor, another test was performed with the same scenario, but this time only the predictions after 1 year were evaluated (about 6 months of data left). The purpose of this is to see the evolution of both models as more data from the specific household is fed to the models. The results show, for the first test house MAC000032, for the generic model, the WAPE value is 0.35 and for the specific model 0.28. This shows that with time, both models can improve their performances with the passage of time. Additionally, Table IV shows the results obtained using the baseline approach. We can see that these results are, in their majority, worst than both the generic and specific models.

Figures [2, 3] document the final model consumption curve for two test houses. These graphs show how the final load forecasting mechanism will behave: utilizing the generic model in the beginning of the deployment and, later, after the green vertical line, replacing it with the specific model. This substitution is performed comparing the MAE values of both models on the last 72 hours. It is possible that the specific model outperforms the generic model temporally. However, it does not seem likely, unless in a extreme scenario, since all the tests indicate that the specific model will always eventually achieve better results. Therefore, the replacement of the forecasting model is permanent. On some test houses, this replacement occurs as soon as possible. For example, in the first test house [2, the green line appears at hour 240, meaning that the generic model was used for 10 days (7 days while the specific model was only gathering data + 3 days of the system gathering

House	Generic Model				Specific Model			
	MAPE	WAPE	MAE	MaxAE	MAPE	WAPE	MAE	MaxAE
MAC000032	1.95	0.44	0.25	4.69	1.83	0.39	0.22	4.64
MAC000091	0.32	0.33	0.09	1.79	0.24	0.28	0.08	2.56
MAC000112	0.42	0.44	0.24	3.59	0.3	0.36	0.2	3.45
MAC000258	0.46	0.36	0.12	2.43	0.39	0.32	0.11	2.3
MAC000283	0.42	0.37	0.11	2.11	0.3	0.3	0.09	2.07

TABLE III: Generic model vs Specific model in 5 test Houses

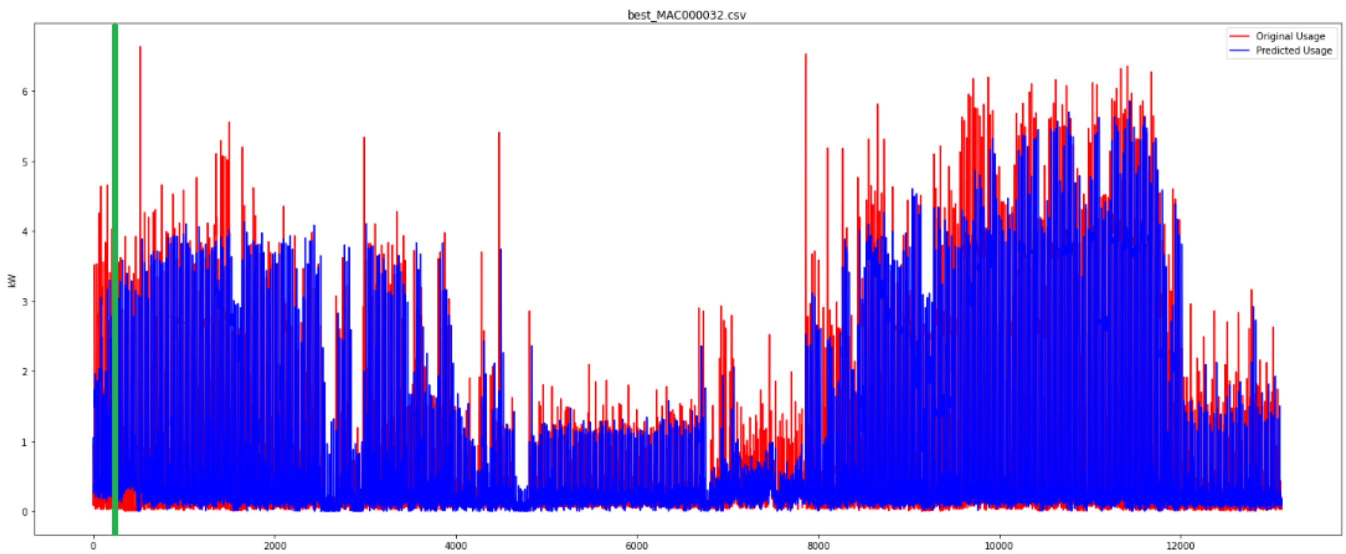

Fig. 2: MAC000032 Model Prediction

House	BaseLine			
	MAPE	WAPE	MAE	MaxAE
MAC000032	2.32	0.68	0.36	6.24
MAC000091	0.28	0.33	0.09	2.08
MAC000112	0.4	0.45	0.24	4.19
MAC000258	0.71	0.71	0.24	3.04
MAC000283	0.49	0.47	0.14	2.96

TABLE IV: Baseline performance in 5 test Houses

error metrics for comparison). However in other houses, such as MAC000112 3, this replacement only occurred after 533 hours (more than 22 days). This occurs because the specific model required more time to provide better predictions than the generic model.

V. Conclusion

In this study, a household load demand forecast system is built. Initially, two different techniques (LSTM and SVR) were compared on five test houses. Results show that SVR achieved better results than LSTM except in terms of maximum absolute error, where in some cases LSTM was better.

The final system makes use of both discussed ML techniques. One of them, LSTM, is used to build a generic model, trained with several houses in order to have a general concept of the average house behavior. Since the model is trained beforehand, it is used while not much information is known about a test house. The second technique, SVR, shown better results for training with information relative to only one house and predicting consumption records for the same house. Therefore, this model was used as the specific model, which is trained with data from a single test house. In a initial stage, as there is not much training data available, it is expected that this model to perform poorly. However, as time goes on, this model will outperform the generic model and thus, replace it. The generic model predicted future house consumption values with acceptable precision, achieving better results than the baseline approach. Eventually, in all cases, the specific model outperformed the generic model. Once the specific model gets better than the generic model, the former replaces the later. In some cases this substitution was carried out very early (10 days) and in another case more than 22 days were necessary for the specific model to achieve better performance than the generic model.

VI. Future Work

In this study, a LSTM neural network was built based on the researched related work. The devised architecture was not subject to more variations seeking further improvements. The SVR ended up outperforming the LSTM, but the potential in NN is much higher. A more in depth study on how many neurons and layers a NN can benefit from, in the energy consumption prediction area, will surely contribute to improve the results.

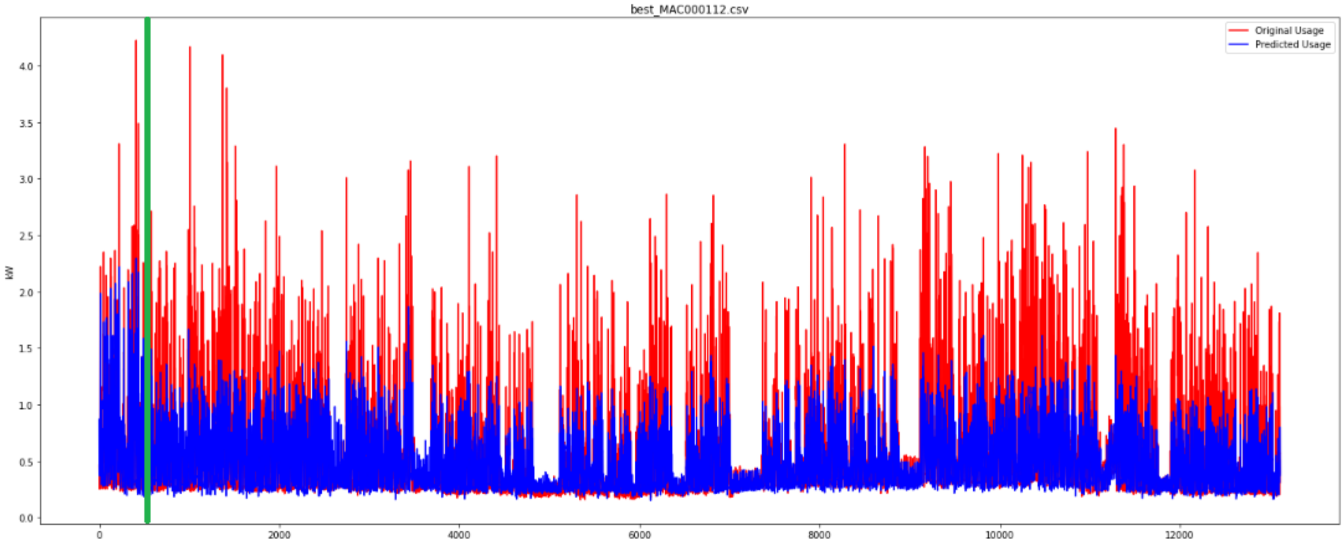

Fig. 3: MAC000112 Model Prediction

Despite the SVR algorithm achieving better performance in predicting energy consumption, it was not chosen to implement the generic model. This is because the utilized library did not implemented incremental learning, which would have forced the training process to be performed at once. Further research on how to implement a SVR based model, that supports incremental learning, might improve the results.

Additionally, the utilized dataset to build the generic and specific models contained several houses, however, it did not contain a high number of features. This study could have benefited from a more detailed dataset to build the generic model, something to have in account when designing the data acquisition system underlying the forecasting algorithm.

Acknowledgment

The present study was partially supported by FCT/MCTES through national funding, by the Portuguese EV4Energy project and, when applicable, co-financed by European funds from project UIDB/50008/2020-UIDP/50008/2020.

References

[1] United Nations Department of Economic and Social Affairs Population Division, "68% of the world population projected to live in urban areas by 2050, says UN — UN DESA — United Nations Department of Economic and Social Affairs," pp. 2–5, 2018. [Online]. Available: https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

[2] X. M. Zhang, K. Grolinger, M. A. Capretz, and L. Seewald, "Forecasting Residential Energy Consumption: Single Household Perspective," *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 110–117, jan 2019.

[3] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, jan 2019.

[4] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, 2018.

[5] T. Petrican, A. V. Vesa, M. Antal, C. Pop, T. Cioara, I. Anghel, and I. Salomie, "Evaluating forecasting techniques for integrating household energy prosumers into smart grids," *Proceedings - 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing, ICCP 2018*, pp. 79–85, oct 2018.

[6] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing 2004 14:3*, vol. 14, no. 3, pp. 199–222, aug 2004. [Online]. Available: https://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] S. Shanmuganathan, "Artificial neural network modelling: An introduction," *Studies in Computational Intelligence*, vol. 628, pp. 1–14, feb 2016.

[9] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 6, no. 2, pp. 107–116, 1998.

[10] A. Blum, "Neural networks in C++ : an object-oriented framework for building connectionist systems," 1992.

[11] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[12] "Smartmeter Energy Use Data in London Households." [Online]. Available: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households