e-Prime - Advances in Electrical Engineering, Electronics and Energy 2 (2022) 100030

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

Performance evaluation of LSTM neural networks for consumption prediction

Davi Guimarães da Silva [a,b,c,*], Marla Teresinha Barbosa Geller [c], Mauro Sérgio dos Santos Moura [c], Anderson Alvarenga de Moura Meneses [a,c]

[a] Federal University of Western Pará – Graduate Program in Society, Nature and Development, R. Vera Paz, s/n, Salé, CEP 68.035-110, Santarém, PA, Brazil
[b] Federal Institute of Education, Science and Technology of Pará, Brazil
[c] Laboratory of Computational Intelligence, Federal University of Western Pará, Institute of Geosciences and Engineering, Brazil

ARTICLE INFO

Keywords:
Deep learning
Internet of things
Monitoring and prediction
Energy efficiency
Neural networks
Long short-term memory
Prediction module

ABSTRACT

Energy consumption and energy efficiency are topics that have attracted the attention of researchers in recent years, in order to seek sustainable solutions for energy production and reduction of costs, aiming to provide a balance between development and protection of natural resources. One of the alternatives that have obtained satisfactory results is the use of technologies based on Internet of Things (IoT) and Deep Learning systems. Based on this, we assessed the performance of Long Short-Term Memory (LSTM) neural networks in time series electric energy consumption prediction, for a forecasting module of an IoT system. Three time series were used and we compared LSTM to the algorithms Extreme Boost Gradient and Random Forest. Computational results indicate that the LSTM model showed a tendency of better RMSE performance in the first data set, and statistically significant better results in other two data sets, according to the Kruskal-Wallis test ($p < 0.0001$ in both cases). Thus, the proposed model was implemented and validated from experiments, presenting accurate prediction for monitoring and estimating power consumption, being applicable to Energy Efficiency and decision making.

1. Introduction

Electricity has great importance for the lives of individuals, companies, institutions, and is directly linked to the development of countries. Nowadays, the extreme dependence on this resource is indisputable, since the performance of various economic activities and social welfare are directly linked to the consumption of electricity.

According to the Brazilian Energy Research Company (Empresa de Pesquisa Energética – EPE) [1], the industrial, commercial, residential and public buildings were the main electricity demands of the country, being responsible for the consumption of about 79,4% of the total energy consumed in the country. However, it is important to note that the achievement of quality of life should not compromise the integrity of the planet, that is, sustainable attitudes are required so that it is possible to maintain the acquired social comfort and also minimize the damage to natural resources, with actions towards energy efficiency.

Among several researches that seek to enable the efficient use of electric energy, the use of Internet of Things (IoT) systems stands out. According to Serpanos and Wolf [2], IoT is characterized by making use of heterogeneous technologies as the devices are being inserted, new requirements of scalability, interoperability and connectivity are added. For Wong and Kim [3], IoT is a new industrial ecosystem that combines intelligent and autonomous machines, advanced predictive analytics, and human-machine collaboration to improve productivity and efficiency.

Another important concept to highlight that can be applied to the efficient use of electricity is Deep Learning (DL) [4,5]. DL is a set of Artificial Intelligence (AI) techniques that use mainly deep neural networks (DNNs), with many layers and units (neurons) between the input and output layer, processed with High Performance Computing hardware (such as Graphic Processing Units).

Currently, DL models have been used for tasks such as image classification and segmentation, forecasting, speech recognition, and object detection. Several systems use DL, such as Siri (Apple's virtual assistant) and Cortana (Microsoft's virtual assistant). In the present work, the Long Short-term Memory (LSTM) DNN developed by Hochreiter and Schmidhuber [6] in 1997 is investigated for load forecasting within an IoT framework. Such DNN is able to learn short and long-term dependencies, being used for processing and forecasting Time Series (TS).

According to Frank et al. [7], a TS is a sequence of vectors $\boldsymbol{x}(t)$, $t = 0, 1,\ldots$, where t represents elapsed time. So, theoretically, $\boldsymbol{x}$ may be a value which varies continuously with t, such as a temperature or load, in our

* Corresponding author at: Federal University of Western Pará – Graduate Program in Society, Nature and Development, R. Vera Paz, s/n, Salé, CEP 68.035-110, Santarém, PA, Brazil.

E-mail address: davi.guimaraes@ifpa.edu.br (D.G. da Silva).

https://doi.org/10.1016/j.prime.2022.100030
Received 30 December 2021; Received in revised form 17 February 2022; Accepted 27 February 2022

case. In practice, for any given physical system, $\boldsymbol{x}$ will be sampled to give a series of discrete data points, equally spaced in time. Over the years, forecasting energy short and long-term TS has become a fundamental activity for industries and public policies. In this sense, Hong [8] discusses the origins of electricity forecasting, as well as its present and future.

Arjomandi-Nezhad et al. [9] and Al Mamun et al. [10] adopt the following categories of forecasting, depending on the time horizon: very short-term loading forecasting (VSTLF), ranging from few seconds to few minutes; short-term loading forecasting (STLF), ranging from few minutes to few hours; medium-term loading forecasting (MTLF), ranging from few days to few months; and long-term loading forecasting (LTLF), ranging from few weeks to several years. More specifically, in the domain of the present article, which is the STLF, Gross and Galiana [11] presented the state of the art in the decade of 1980, when the literature comprised, for example, ARMA (Auto-Regressive Moving Average) and state-space models.

Hong et al. [12] review more recent works on energy forecasting domains such as wind and solar power systems, as well as electricity prices and demands. The authors also discuss the research frontiers in energy forecasting which are forecast combination and ensemble forecasting, hierarchical forecasting, probabilistic forecasting, as well as Artificial Intelligence (AI) and Machine Learning (ML) techniques, which are our focuses in the present work.

By 2000, Hippert et al. [13] in their article stated that “the reports on the performance of NNs in forecasting have not entirely convinced the researchers in this area” and reviewed 40 papers on the application of NNs to short-term load forecasting. It is worth to remind that back then, NNs such as conventional Multi-layer Perceptron (MLP) and Kohonen Maps [14] were at their apogee in several research areas. Nevertheless, Hippert et al. [13] conclude that more research with more rigorous standards were necessary for a sounder discussion. For building energy consumption prediction, Zhao and Magoulès [15] reviewed articles with the techniques used, classifying them in engineering methods (which calculates thermal dynamics according to physics principles), statistical methods (particularly regression methods which correlates energy consumption or energy index with other variables), and ML methods such as Neural Networks (NNs) and Support Vector Machines.

With the breakthrough of DNNs and more rigorous methodologies for development as well as the adoption of statistical methods for assessing the performance of ML algorithms [16,17] the NNs became not only more acceptable and reliable but more reproducible and accurate as well.

For example, Hou et al. [18] applied the LSTM DNN in the residential load forecasting process of an adaptive load aggregation method with a number of selected households. The Mean Absolute Percentage Errors (MAPEs) of the forecasting load were 14.3% and 10.2% to 50 and 100 households, respectively. And when the number of households reaches 150, the MAPE of both the traditional method and proposed method are both below 0.1, which meets the requirement of load forecasting accuracy. Shaqour et al. [19] use load aggregation and DNNs for the forecasting process. The authors investigate the effects of electrical demand aggregation size on STLF. DL architectures were used for STLF, such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and bi-directional LSTM/GRU. The results show that the lowest MAPE errors was achieved with DNN (2.47–3.31%) at a 479-aggregation level, and MAPE errors below 10% can be achieved at a 30-aggregation level.

As another example of DNN applied to energy forecasting, Ozer et al. [20] presented the LSTM DNN with transfer learning as a solution to the problem of load forecasting when there is not enough historical data. This situation can happen in recent buildings and newly installed meters. Data for transfer of learning were obtained from different parts of the world. The critical issue was to determine which dataset for transfer learning has the best similarity to the real dataset.

Integrated LSTM and CNN models are being used with good results for short-term load forecasting studies. Rafi et al. [21] performed the STLF of the Bangladesh power system, integrating the LSTM and CNN models, considering the advantages of each. The motivating issue cited by the authors is that often a model alone does not perform well for load predictions. The hybrid model (CNN-LSTM) is composed of a module CNN, an LSTM module and a feature-fusion module. The model validation was performed with the evaluation metrics mean average error (MAE), root mean square error (RMSE), MAPE and the coefficient of determination (R^2). The model CNN-LSTM was compared with the approaches LSTM, radial basis functional network (RBFN) and Extreme Gradient Boosting (XGBoost). According to the research, the results of the model CNN-LSTM were superior to the models tested in all validation cases. An LSTM and CNN integration model was also used by Farsi et al. [22] for load forecasting in Malaysia (hourly consumption) and Germany (daily consumption). According to the authors, the integrated network which they called PLCNet, improved the accuracy from 83.17% to 91.18% for the data in Germany and 98.23% of accuracy for the data from Malaysia, reinforcing that the model has good performance for be used for STLF.

Despite the rising complexity in DL models for load forecasting, our focus was the assessment of a DNN model for an IoT framework. In this sense, we assessed the performance of LSTM DNNs applied to energy time series consumption prediction, for a prediction IoT module that performs from the collection of consumption data.

The originality of this refers to application of LSTM to the prediction of TS of electricity, as well as the evaluation of its performance through statistical tests in three datasets with real consumption data (one of them completely “unknown” for the models), enabling the integration of diverse knowledge involved for energy monitoring in industries, commercial establishments and public buildings, considering the use of DL, since there is no similar system being developed in the Amazon region according to the bibliographic research carried out. Besides, the present LSTM results will form a baseline for future tests, with more complex architectures for IoT systems.

Due to the recent successful applications of tree-based algorithms, we chose two of them in order to compare the performance of the LSTM NNs, namely the XGBoost and Random Forest (RF). For example, Ahmadi et al. [23] compared tree-based algorithms, including XGBoost and RF for long-term wind power forecasting, demonstrating a successful application of such algorithms. Arjomandi-Nezdah et al. [9] also compared several tree-based algorithms applied to the country-level demand in Germany day-ahead forecast in order to analyze the energy consumption TS during the covid-19 pandemic.

Thus, the main contributions of this work are: (i) statistical performance analysis of LSTM DNN architectures with different numbers of layers in different datasets; (ii) prediction of time series of electricity consumption for real data collected in a city in the Brazilian Amazon region, since studies of this type are scarce; (iii) experiments to verify the generalization ability of the proposed LSTM model, including data not used for training the models, and comparison to the algorithms XGBoost and RF.

The remaining of the article is organized in the following sections. Section 2 presents the related works. Section 3 describes the theoretical background. Section 4 presents the EnergySaver software. Section 5 describes the methodology for assessment of the LSTM DNNs for consumption prediction. Section 6 presents the computational experimental results and discussion. The results are discussed in Section 7, and Section 8 provides the conclusions.

2. Related work

The development of systems for monitoring electricity consumption is extremely important for the construction of policies aimed at energy efficiency. Researchers have sought sustainability solutions for energy production and reduction of its costs, aiming to provide a balance

Table 1
Summary of related works.

Authors	Year of publication	Models	Major contributions
de Sousa et al. [24]	2020	NNs	Used conventional NNs with Prediction Intervals (PIs) for forecasting solar irradiance.
Aswani et al. [25]	2012	Model Predictive Control	A model system for managing air conditioners aiming to provide an efficient use of electricity by reducing its consumption, using IoT sensors.
Ruano et al. [26]	2018	SPWS (Self-powered wireless sensors)	An implementation and testing of self-powered wireless sensors (SPWS), specifically designed for residential energy management applications, as well as the use of an IoT platform.
Moura et al. [27]	2021	WSN (Wireless sensor network)	An IoT platform based on wireless sensor network infrastructure (WSN), which supports an intelligent system to control heating, ventilation and air conditioning (HVAC).
Geller and Meneses [28]	2021	UML for an IoT/AI system	Proposals for using Unified Modeling Language (UML) extensions to model an artificial intelligence energy monitoring system to predict consumption.
Silva et al. [29]	2021	LSTM DNN	A prediction module with Deep Learning for the IoT-based EnergySaver framework, which aims to monitor electric power consumption, from data capture to consumption prediction for the following month.
Ozer et al. [20]	2021	LSTM (XCORR cross-correlation)	Used LSTM in a transfer learning approach based on XCORR cross-correlation.

between development and protection of natural resources, especially from the use of systems based on IoT and ML. Table 1 shows the main articles related to the present work. de Sousa et al. [24] used conventional NNs with Prediction Intervals (PIs) for forecasting solar irradiance, providing an assessment of future uncertainty in different scenarios. PIs consist of lower and upper bounds that a future unknown value will lie within, with a predetermined probability, called confidence level, becoming more reliable and informative for decision-makers, giving support to select the best action under uncertain conditions. The present work goes one step further, using LSTM DNNs for prediction, and applying them to energy consumption prediction.

In [25] the authors presented a model system for managing air conditioners aiming to provide an efficient use of electricity by reducing its consumption. The proposed system works as follows: micro controllers with sensors are used to obtain data, which are stored on a local server in sMAP (Simple Measurement and Actuation Profile) database. A control computer accesses the internet to get up-to-date weather forecasts as well as data obtained from the room sensors, then runs a MPC (Model Predictive Control) scheme, based on DL that calculates a control input, which is sent from the local server to the thermostat which in turn transmits a corresponding signal to the air conditioner thus adjusting the room temperature. The results of the experiments show that there has been a 30% to 70% reduction in energy consumption while maintaining a comfortable room temperature. The energy savings are due to the control compensating for variable occupancy, while considering transient and stable electrical consumption.

Another IoT model was proposed by Ruano et al. [26], from an approach for prototyping and validating two components of an integrated system, as part of a solution for a predictive control of Heating, Ventilating and Air Conditioning (HVAC) systems. They presented the results of an implementation and testing of self-powered wireless sensors (SPWS), specifically designed for residential energy management applications, as well as the use of an IoT platform, and its use in a real application. The results pointed out that the use of small devices, allow seamless integration into building energy efficiency applications, as well as easy to configure and install, inexpensive compared to available alternatives, providing alerts for diagnostics, and most importantly, enabling perpetual autonomous operation under reasonable conditions and lighting conditions common in buildings and homes.

Moura et al. [27] designed and installed an IoT platform based on wireless sensor network infrastructure (WSN), which supports an intelligent system to control heating, ventilation and air conditioning (HVAC) and lighting systems in buildings. The goal was to find options to achieve more sustainable uses of energy, considering the integration of renewable generation sources and the use of an intelligent load control system. As a result, the approach points out how to provide the right technology to existing old buildings, contributing to near-zero energy consumption (nZEB) at a low cost.

Geller and Meneses [28] presented some proposals for using Unified Modeling Language (UML) extensions to model IoT systems already available in the literature [30–32]. An artificial intelligence energy monitoring system to predict consumption was used as a case study and was modeled with UML resources. It was possible to conclude that UML diagrams and their extensions can represent the different views of an IoT application (static, behavioral, security, etc.), motivating researchers to use them, giving rise to several proposals.

Silva et al. [29] presented an approach for the creation of a prediction module with deep learning for the IoT-based EnergySaver framework, which aims to monitor electric power consumption, from data capture to consumption prediction for the following month. Experimental results indicate that the software obtained satisfactory results from data collection with IoT to consumption prediction using the module with LSTM DNN, with preliminary case use tests. However, a closer look regarding the LSTM DNN statistical performance was still required, giving origin to the present work.

Ozer et al. [20] used LSTM in a transfer learning approach based on XCORR cross-correlation. In this way, the data sets (real and for transfer learning) were individually normalized and XCOOR was applied to determine which would be the most suitable. LSTM model is trained with the data and the weights of the resulting models are transferred and trained with the original data, ending with the tests. The results obtained with the RMSE, MAPE and MAE metrics showed that the proposed model using LSTM and transfer learning was successful compared to models that were known to work well on small datasets such as RF, XGBoost, and Light Gradient Boosting Machine (LGBM) algorithm.

The work proposed herein differs from the other articles in the literature once we propose the use of an LSTM DNN to perform the consumption prediction developed for an IoT module (see also [29]) in a framework for monitoring the electricity consumption of a specific equipment or a building, for example. We also perform the comparison between the LSTM DNN and the well-known algorithms XGBoost and RF using Time Series Cross-Validation (TS-CV), as well as using data not used for training the models, using statistical tests in order to demonstrate the robustness of our results.

3. Theoretical background

3.1. Long-short term memory (LSTM) deep neural networks

The concepts of Artificial Neural Networks (ANNs) emerged from the paradigm of AI, where it is believed that building a system that simulates the structure of the brain will present intelligence, that is, it will be able to learn, assimilate, make mistakes, and learn from its mistakes [6]. Thus, the first reference to ANN theory was around 1943, when Warren McCulloch and Walter Pitts published the article "A Logical Calculus of the Ideas Immanent in Nervous Activity", where they presented what

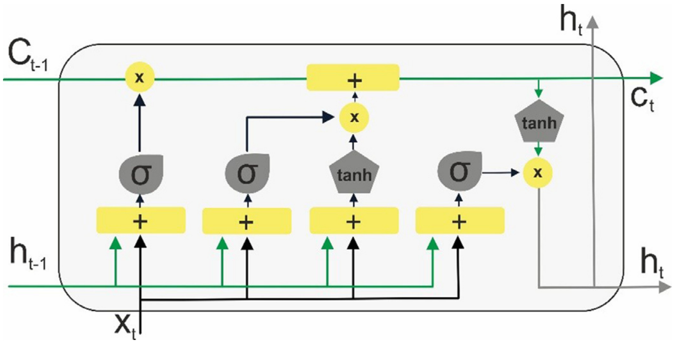


Fig. 1. LSTM DNN structure.

became known as McCulloch's Boolean neuron [14]. ANNs are based on the structure of biological neurons and are composed of devices called artificial neurons that, despite not being a substitute for biological neurons, can solve several problems [6].

LSTM DNN was used in this research. This is the use of a variation of the recurrent network with so-called LSTM units. The goal was to propose a solution to the vanishing gradient problem, common in RNNs. As a result of the experiments, the LSTM obtained more successful runs, learning much faster [6]. The LSTM DNN structure is shown in Fig. 1.

An LSTM unit is composed of a neuron and input, forgetting, and output gates. The neuron is responsible for the memory of the network, represented by weighted sum-based activation, and the gates are a way to optionally enable information.

LSTM network consists of several critical components [33], which are: (a) Hidden state: used to determine what to forget, enter and exit at the next step; (b) Input state: combination of the hidden state and the current input; (c) Internal state: values with memory function; (d) Input gate: decides whether the input reaches the internal state; (e) Forgetting gate: decides whether the internal state disregards the previous one; (f) Output gate: decides whether the internal state is passed to the output and the hidden state at the next step.

The structure of gates used by LSTM networks is what allows the proposed solution for several problems related to sequential models with dependencies, as is the case of the electricity consumption prediction problem, used in this research. It is important to highlight that several works in the literature have used LSTM network models and obtained satisfactory results relative to consumption forecasting.

3.2. XGBoost algorithm

The XGBoost algorithm is an open source library that provides an efficient and scalable implementation of the Gradient Boosting framework, being an ensemble tree method, initially developed by Chen and He [34]. XGBoost is described by Chen and Guestrin [35] and its main objective is to attain fast execution and model performance, trying to decrease overfitting chances. The XGBoost algorithm can be used for either regression or classification, because it is flexible, with many hyperparameters, adjusting it to the problem to be solved and to different data types.

3.3. Random forest

The Random Forest (RF) algorithm was introduced by Breiman [36] and can be used for both classification and regression. It is a classifier consisting of a large set of simple classifiers (Decision Trees), and its representation is in the form of identically distributed random independent vectors [37]. RF is a term used for ensemble methods that use tree-type classifiers, which from a training set using bagging (a meta-algorithm that aims to improve the classification and regression of models), builds a large number of decision trees out of the subset of data. Using bagging for training avoids over-fitting and reduces variance. For regression, the results obtained by Random Forest are based on the average of the numerical result returned by the different trees [37].

4. The EnergySaver software

As mentioned, EnergySaver is a framework aimed at monitoring electric power, where the data capture system uses open source technologies applied to IoT, embedded systems, and a prediction module with LSTM DNNs [28,29].

In Fig. 2, the general flow of the system operation is described as follows: Initially, the consumption data is captured by the multimeter, then it is sent to the Raspberry Pi3, which in turn sends by Mosquitto Broker (MQTT) to the Flask server. On the server, this data is stored in the MongoDB database and displayed in real time on an existing web page on the local server. With the consumption data obtained by the server daily, every first day of the month, the LSTM network is trained with data from the previous months and tested with the new data from the month that just ended. From this, the system sends a message to the System Administrator with information related to the forecast for the following month.

In order to model EnergySaver we used diagrams from the UML, considered a standard for modeling different types of systems [28]. The modeling of IoT systems with UML is still the subject of research for the adaptation of this visual language to the specificities of these systems. Some works such as [30–32] have consolidated UML as a resource for modeling IoT systems.

One of the benefits of using a language already recognized as a standard is the ease of resources available and the possibility of extensions to represent the specific characteristics of each system. The challenge of modeling IoT systems with UML is to represent a sufficient level of detail for engineers to implement them while abstracting the complexities so that they can be deployed with ease. More details on modeling EnergySaver can be found in [28]. In [29] the technologies used are presented, such as Multimeter, Arduino with an electric current sensor, Raspberry Pi, Mosquitto broker, MongoDB and Flask micro-framework, as well as the creation of a prediction module for the EnergySaver framework.

4.1. Creating the forecasting module for EnergySaver

In [29] an approach to create a prediction module using an LSTM model was presented. The process begins with the collection of electricity consumption data in equipment or in a building, to create a dataset that will be used to train and test the network. With this data available and treated, the network is trained and tested to predict the next points. Fig. 3 shows the general flow of the LSTM module.

LSTM module described in Fig. 3 performs both data cleaning and the LSTM DNN execution for forecasting. This process is performed automatically on the first day of each month. The data received and stored in the database must go through a cleaning process before being used by the net for training and testing. Afterwards, the LSTM DNN is run to perform the consumption forecast, as well as a concatenation process between the training and testing file, creating a new training file with the old data and the data used to test the net. At the beginning of the next month the entire process described in the previous steps is performed again [29].

5. Methodology

Topics of the methodology used in this work are described in the flowchart in Fig. 4, with the analysis of the data sets, their preprocessing, the models and architectures for time series prediction, as well as the statistical tests and results of their application for prediction (see Sections 5.1.1 and 5.1.2). For the experiment with the "not-known"

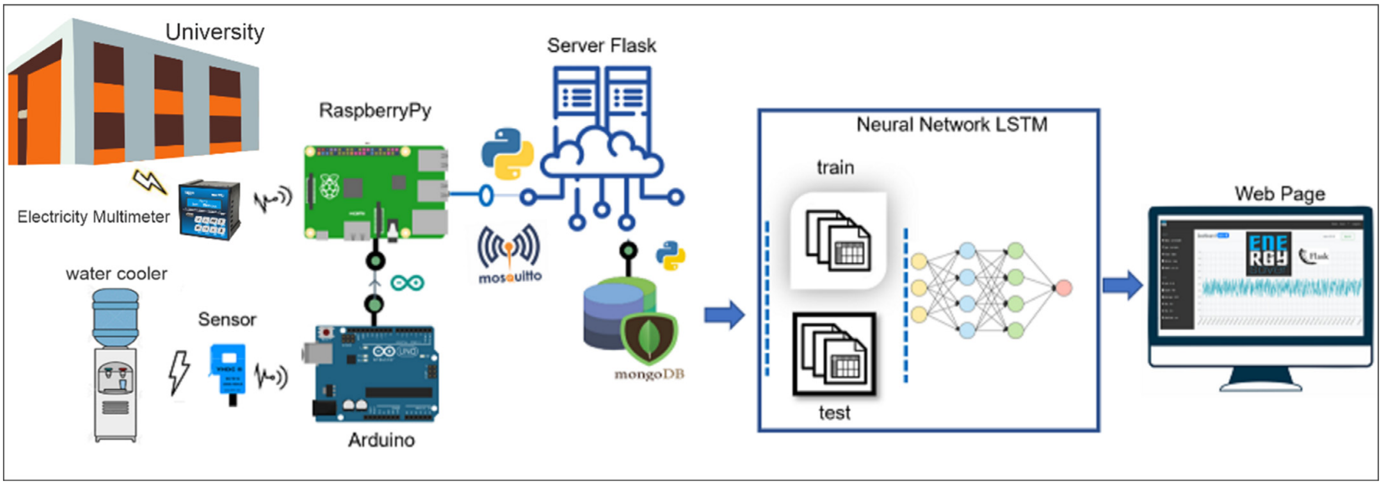


Fig. 2. EnergySaver conceptual scheme. (Adapted from [28]).

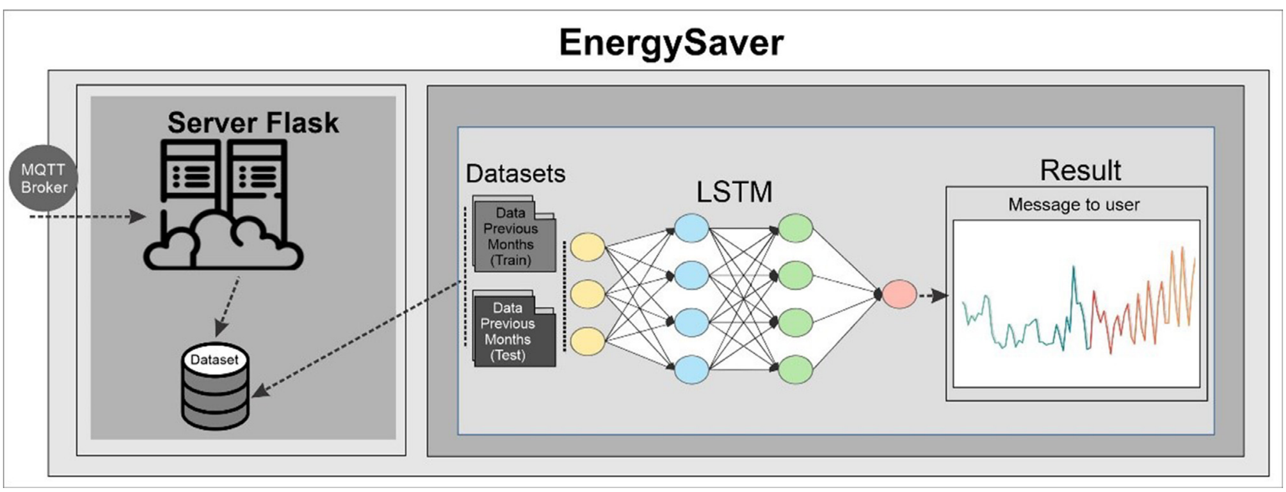


Fig. 3. Proposed LSTM Module [29].

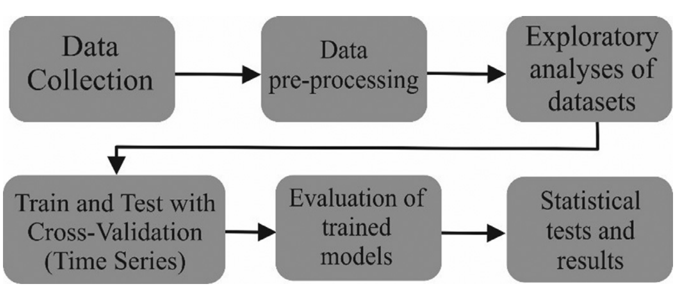


Fig. 4. Flowchart of the methodology used in this work.

data set (Section 5.1.3), the TS-CV was not performed, since one already trained model was used.

5.1. *Case study and datasets*

In the present work, the LSTM prediction was evaluated from different points of view: (a) the consumption of an individual residence; (b) the consumption of a university building with several departments (classrooms, auditoriums, cafeteria, laboratories, among others); and (c) the Singapore electric demand, which is a data set not used for the present models' training or testing, also used by de Sousa et al. [38].

Thus, the experimental results were conducted on three datasets, (a) DS1-UCI (Individual Household Electric Power Consumption - UCI Repository[1]), which contains data from a single residence in Paris, France; (b) DS2-LABIC which is a real consumption data set collected from a building of the Federal University of Western Pará, in the city of Santarém, state of Pará, in the Brazilian Amazon; and (c) DS3-SED (Singapore Electric Demand), which is available at the Energy Market Company website,[2] which operates Singapore's wholesale electricity market. In the data sets described in (a) and (b) the intended time horizon was one month, with the 90 observations in the sliding window being used to forecast the next point (10 minutes intervals between observations for both data sets). The data set in (c) had 10 minutes intervals and all the period of the TS was predicted, as described in the next section.

5.1.1. *Power consumption time series DS1-UCI dataset*

The DS1-UCI data set was chosen because it is widely used in the literature for predicting electric power consumption [39–41], among others, so that its reproducibility and comparison with other works can be verified.

[1] https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption.

[2] https://www.emcsg.com/MarketData/Priceinformation.

The DS1-UCI dataset has data regarding the individual residential electricity consumption of a household in Sceaux, France, provided by the UCI Machine Learning Repository (www.archive.ics.uci.edu), which is commonly used by researchers for the validation of their electricity consumption prediction architectures. The considered collection period is from December 2006 to November 2010 (47 months), displayed in 1-minute units with a total of 2075259 points in kilowatts (kW).

In our tests, the univariate time series was used, and the data refers to global active power, being resized to 10-minute intervals (207,526 records), where the periods from December 2006 to October 2010 (203,801 records) were used for network training and November 2010 (3725 records) for validation.

5.1.2. Power consumption time series DS2-LABIC dataset

The DS2-LABIC data set is related to one of the university buildings that the EnergySaver software will monitor. This TS has characteristics such as the consumption highly influenced by the air conditioning systems, highly demanded in the Amazon region, as well as seasonality due to the University activities.

The DS2-LABIC data were obtained in the city of Santarém, Pará, Brazil, located in the western Amazon region. The data were collected by researchers from the Laboratory of Computational Intelligence (LabIC) in a building of the Federal University of Western Pará (Universidade Federal do Oeste do Pará, UFOPA). The collection period was from January to August 2019.

The consumption data for the period has 256,092 records, based on the aggregate global active power in Watts (W), referring to the total consumption in each period. Thus, after transforming the raw data to index on a fixed basis (for 10 minutes intervals), the dataset was left with a total of 33,830 records. From this total, 29,492 records referring to the months of January to July 2019 were used for cross-validation, corresponding to approximately 87% of the data set, and for validation 4338 records referring to the month of August 2019, corresponding to 13% of the total.

5.1.3. Power consumption time series DS3-SED dataset

The DS3-SED data set is a large scale demand TS, used to test the capacity of generalization of the models in data that are not “known” by the forecasting models, as performed in [23] in the context of long-term wind power forecasting. This step will allow tests to verify if the models are able of making predictions on data not presented for training.

The columns present in the dataset are: price type, date, period, price ($/MW) and demand (MW). In this case the univariate time series of demand was used with 30 minutes intervals. The considered collection period is from January 1, 2010 to March 30, 2011, according to Sousa et al. [38], with 21,701 records.

5.2. Training and testing

In order to perform the training and testing, a computer with the following configurations was used: Intel® Core™ i5 - 7400 processor, 16GB of RAM, and Nvidia RTX 2080Ti GPU with 11GB of VRAM. The LSTM DNN was implemented in Python using the library Keras 2.6.0 using Tensorflow 2.6.0 as backend. The main settings of the LSTM were: normalization between 0 and 1; 100 epochs; batch_size 32; the size of the sliding window (90 observations for both training and testing); and sequential model. The libraries XGBoost 1.5.2 and Scikit-Learn 1.0.2 were also used. XGBoost was implemented with 1000 estimators for DS1-UCI and 500 estimators for DS2-LABIC. RF was implemented with 100 estimators in all cases.

5.2.1. Time series cross-validation (TS-CV)

One way to evaluate the behavior of an out-of-sample model is to use cross-validation [42,43]. However, in a conventional k-fold cross-validation, training and test data are randomly selected. Such typical CV as those used in classification problems, are not applicable in the TS case because of the data temporal dependency. Thus, the training set must contain only data observed before the test set data [44]. In the present work, we used the function TimeSeriesSplit from the Scikit-learn library (https://scikit-learn.org), which performs the partition of a time series into $n+1$ parts, where n is the number of partitions informed, seeking to find an optimal number of elements to be used for model training, as shown in Fig. 5.

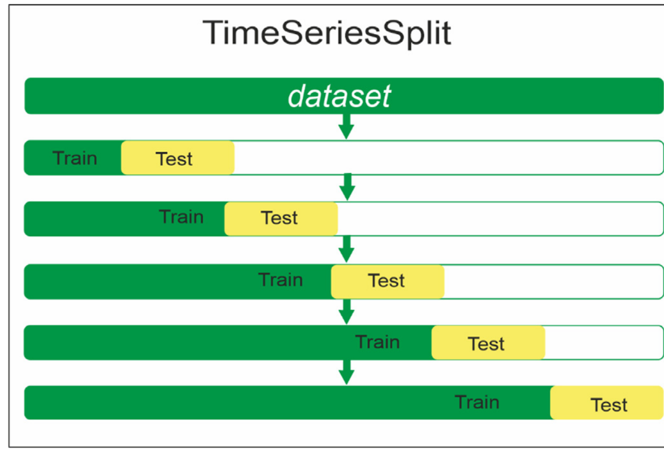


Fig. 5. Example of 5-fold TS cross-validation with TimeSeriesSplit.

In Fig. 5, at each training/test execution, training data is always prior to test data, so that the time dependency is respected. The horizontal axis represents the size of the data set and the vertical axis represents the iterations of the cross-validation. The CV was performed with 10 partitions ($k=10$), as Witten and Frank [45] state that extensive tests performed on different datasets showed that 10 is a value close to the number of partitions at which the best estimates can be obtained. For each TimeSeriesSplit fold, the models generated were saved in .h5 format for later use.

5.3. Performance metrics for evaluation

Four performance metrics for TS prediction were used for performance evaluation, namely, the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the MAPE and the coefficient of determination R^2-Score. These metrics are calculated by the Eqs. (1)–(4).

$$RMSE = \sqrt{\left[\frac{1}{n}\sum_{i=1}^{n}\left(|x_i - y_i|\right)^2\right]} \tag{1}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \tag{2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right| * 100 \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{4}$$

Where y_i, $\hat{y}_i$, and n, are respectively the measured value, the predicted value, and the total number of samples, and $\bar{y}$ is the average value of the observed output over samples existing. RMSE is the standard deviation of the prediction errors, which points out how far the data points are spread from the regression line. MAE measures the average magnitude of the prediction errors and ignores their directions and is the average of the absolute differences between the predicted and actual values for all instances in the test set, considering that all individual differences have the same weight. MAPE is a relative metric, which expresses average value of relative error as a percentage of the actual data. The R^2-Score is used to evaluate the performance of a linear regression model, resulting in the amount of the variation in the output dependent attribute which is predictable from the input independent variables.

5.4. Statistical analysis

Aiming to determine the best model and assuming that the results do not have a normal distribution, non-parametric statistical tests of Kruskal-Wallis [46] is used to compare three or more groups of data were applied. This test is used when you want to test the hypothesis that several samples have the same distribution. The null hypothesis is that there is no statistically significant difference between groups ($p < 0.05$). The alternative hypothesis indicates that at least one group is different. Thus the Dunn's post-hoc test [46] is needed for pairwise comparison between the models when there is statistically significant difference in the Kruskal-Wallis test. For the statistical tests the metric RMSE was used.

The Kruskal-Wallis tests were performed at SAS OnDemand for Academics platform (https://welcome.oda.sas.com/home), with the procedure NPAR1WAY. Dunn's post-hoc tests were performed in the same platform, with a SAS macro [46]. The threshold 0.05 was adopted for the statistical tests.

5.5. Computational experimental design

We have divided the computational experiments in three groups: (i) a preliminary TS-CV experiment, in order to verify if there is statistically significant difference between the LSTM models regarding their number of layers (using DS1-UCI and DS2-LABIC); (ii) a TS-CV experiment in order to compare the performances of LSTM DNN to the ML algorithms XGBoost and RF (also using DS1-UCI and DS2-LABIC); and (iii) the forecasting of DS3-SED with the LSTM, XGBoost, and RF models saved in the previous DS2-LABIC TS-CV experiment, in order to verify the robustness of the models regarding a data set that was not used for training the models.

In the preliminary tests in (i), for each dataset (DS1-UCI and DS2-LABIC), four LSTM architectures (number of layers) were used (with 2, 4, 6, and 8 layers), namely L2, L4, L6 and L8. For each training of the 10-fold TS-CV, one model was generated and saved (10 in total), and applied to the test set, generating 10 metrics MAE and RMSE each. Since these were preliminary tests, only two metrics were assessed, based on the normalized predicted values.

Since there was no statistically significant difference between the models in the preliminary tests, the model L2 (see results in Section 6.2.1) was chosen for the train-test experiment in (ii), since it would be the fast model to be trained, with lower number of weights. The LSTM models saved after the training with the DS2-LABIC (see results in Section 6.2.2) were used for forecasting the DS3-SED data set.

6. Results

6.1. Exploratory data analysis

6.1.1. DS1-UCI data set

Fig. 6 shows the samples per day and week, referring to the DS1-UCI dataset.

Fig. 7 presents the boxplots aiming to verify the distribution of the data related to active power. In the boxplot in Fig. 7, the statistics related to active power are presented, and reveal that the average load is almost constant over the years, while the quarterly graph shows lower consumption in the second and third compared to the others because the demand for power in France increases in the first and fourth quarters due to the heating load when it is winter.

Fig. 8 presents the autocorrelation (ACF) of active power, where the observation at a given instant is related to past observations. According to Fig. 8, a sliding window with 90 observations is adequate for training and forecasting (see also Figs. 11 and 14).

Table 2
Average metrics obtained by the network models in each dataset.

Dataset	Model	MAE	RMSE
DS1-UCI	L2	0.0332	0.0498
	L4	0.0290	0.0442
	L6	0.0291	0.0444
	L8	0.0285	0.0444
DS2-LABIC	L2	0.0202	0.0283
	L4	0.0217	0.0299
	L6	0.0220	0.0301
	L8	0.0258	0.0339

6.1.2. DS2-LABIC data set

Fig. 9 displays the samples by day and week, referring to the DS2-LABIC dataset. Data presented refers to the global active power.

In Fig. 9, there are some variations related to weekdays, which have higher consumption, whereas the demand on weekends is lower. There are also some consumption peaks (outliers) that deviate from the average. Fig. 10 presents the boxplot aiming to verify the distribution of data related to active power where high values outliers can be observed. Fig. 11, presents the active power ACF, corroborating for a sliding window with 90 observations.

6.1.3. DS3-SED data set

Fig. 12 shows the samples per day and week, referring to the DS3-SED dataset.

The data presented in Fig. 12 also provides information regarding the days and weeks electric power consumption, which indicate the TS behavior. Fig. 13 presents the boxplots aiming to verify the distribution of the data related to active power.

In Fig. 13, the statistics related to active power are presented, and reveal that the average load is almost constant over the years (Years), while the quarterly graph shows lower consumption in the first and third compared to the others (Quarters). Fig. 14 presents the autocorrelation of active power.

6.2. Computational experimental results

6.2.1. Preliminary experiments (number of LSTM layers) results

Figs. 15 and 16 show the boxplots of the results obtained from the RMSE metric for the normalized values.

In Fig. 15, for the DS1-UCI dataset, it seems that model L2 had a worst performance, however the null hypothesis of similar performance could not be rejected in the Kruskal-Wallis test ($p = 0.0833$). In Fig. 16, for the DS2-LABIC dataset, the L8 architecture presents larger amplitudes. In this case, there was no statistically significant difference between the number of the layers of the models ($p = 0.7801$).

Table 2 presents the average MAE and RMSE metrics in the preliminary tests, obtained by the 10 models, for each dataset.

In Table 2, for the DS1-UCI dataset, the 8-layers model obtained lower average RMSE (0.0285). However, the lowest average RMSE was 0.0202 correspondents to the 2-layers model.

6.2.2. Prediction of data from DS2-LABIC data set not used for training

With the DS2-LABIC dataset, we sought to predict the electricity consumption for the month of August 2019 (the last 4.338 observations of the TS), using the model L2 to July (with lower overall RMSE). Fig. 17 shows the results for this part of the data set, which showed lower averages in the errors of the metrics used.

Fig. 17, shows that the LSTM algorithm was able to learn from the training data, and forecast the month of August 2019 in DS2-LABIC data set. The prediction of August 2019 obtained RMSE = 0.0289 and MAE = 0.0205. With these results presented it is possible to highlight that the proposed network model was able to learn from the train-

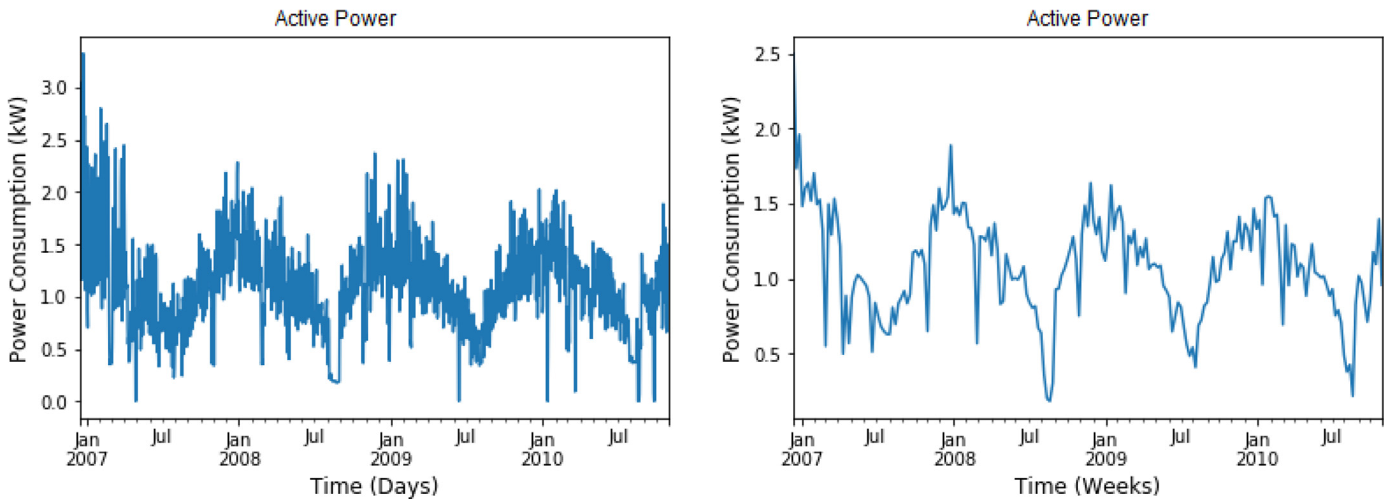


Fig. 6. Samples of the electrical consumption data from the DS1-UCI dataset.

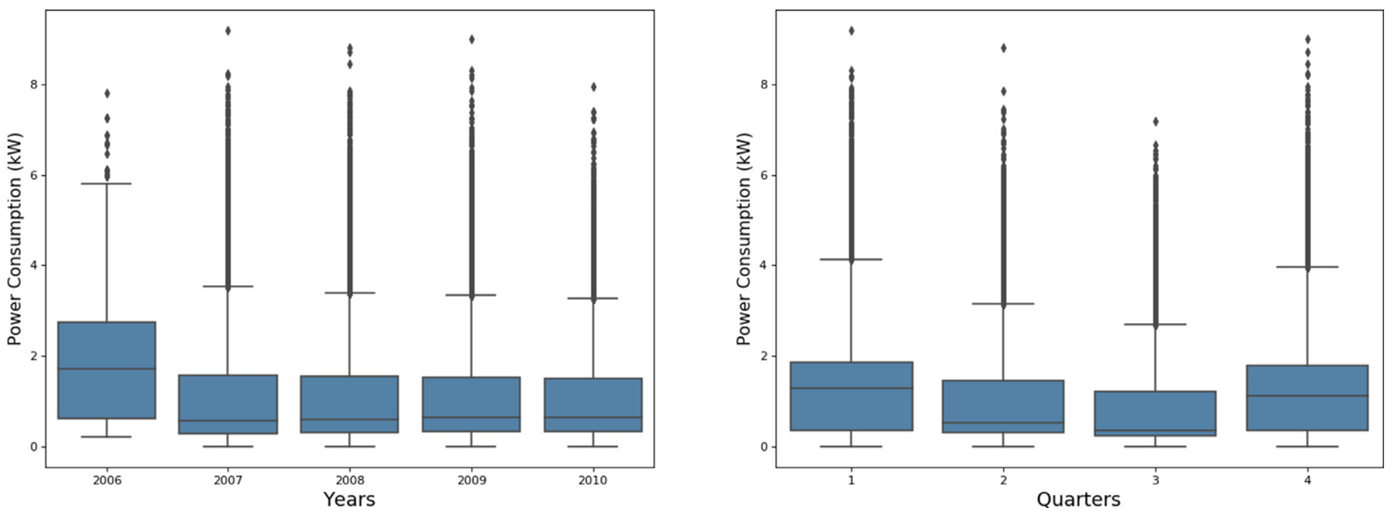


Fig. 7. DS1-UCI dataset box-plots of electric load (Yearly and Quarterly).

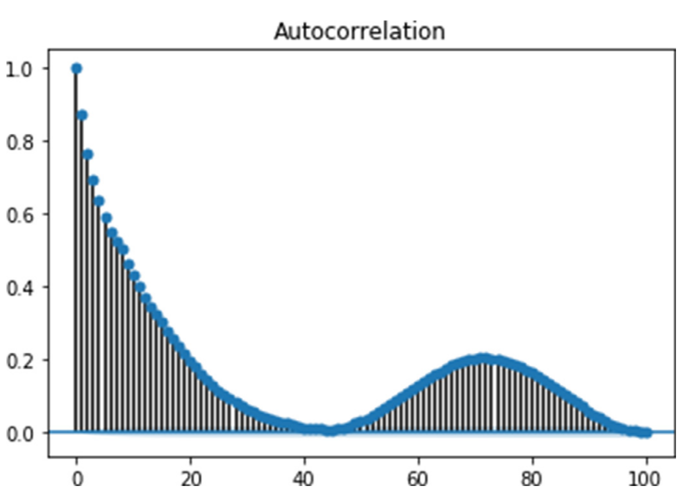


Fig. 8. Autocorrelation function - DS1-UCI dataset.

Table 3
LSTM results for DS1-UCI dataset.

k (TS-CV)	DS1-UCI LSTM Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	0.472	0.262	28.0	75.5
2	0.463	0.262	25.7	74.4
3	0.471	0.269	28.0	75.6
4	0.458	0.245	28.6	75.0
5	0.464	0.250	24.0	76.3
6	0.457	0.242	29.5	78.2
7	0.445	0.242	29.8	77.3
8	0.463	0.251	23.0	76.5
9	0.457	0.254	26.2	77.1
10	0.461	0.255	29.5	74.0
Average	0.461	0.253	27.2	76.0
Median	0.462	0.252	28.0	75.9
Std-dev	0.008	0.009	2.4	1.3
Minimum	0.445	0.242	23.0	74.0
Maximum	0.472	0.269	29.8	78.2

ing data, and when applied in the tests, the predictions were close to the actual value, which demonstrated the success of preliminary tests.

6.2.3. Train-test comparison of LSTM, XGBoost, and random forest results DS1-UCI data set

Tables 3–5, respectively show the test scores for LSTM, XGBoost, and RF, containing each one of the 10-fold TS-CV metrics RMSE, MAE,

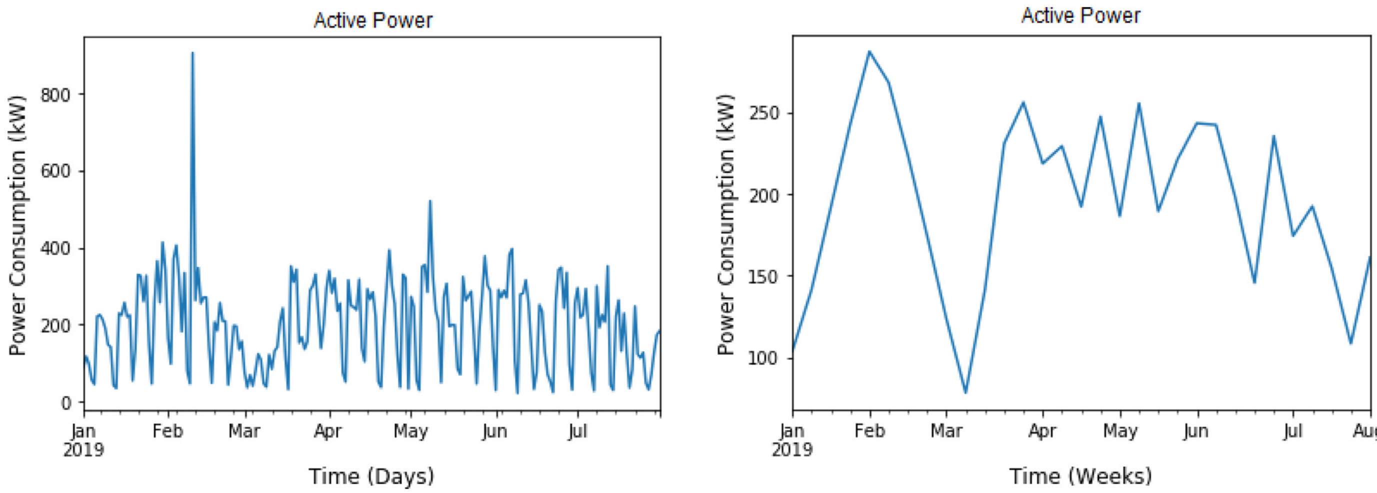


Fig. 9. Samples of the electrical consumption data from the DS2-LABIC data set.

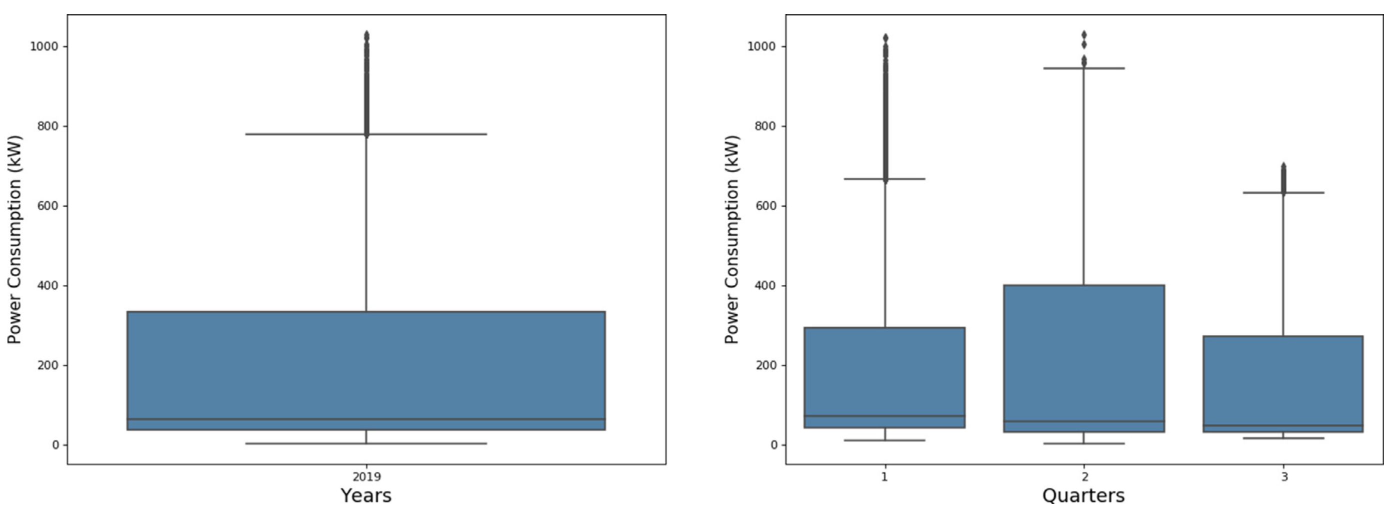


Fig. 10. DS2-LABIC dataset box-plot of electric load (Yearly and Quarterly).

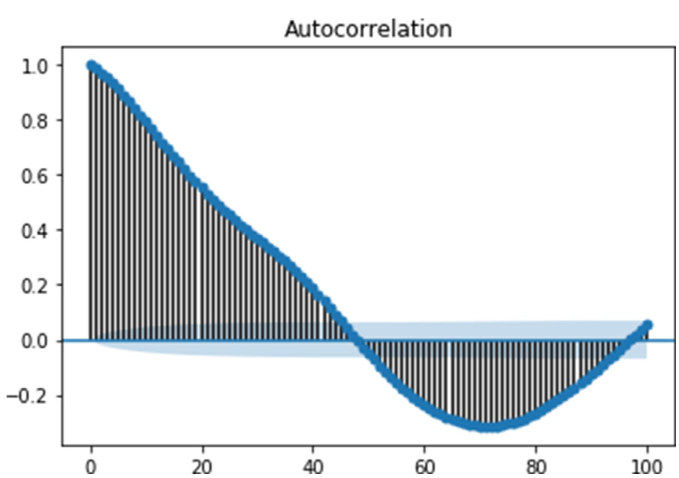


Fig. 11. Autocorrelation function - DS2-LABIC dataset.

Table 4
XGBoost results for DS1-UCI dataset.

k (TS-CV)	DS1-UCI XGBoost* Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	0.484	0.279	29.2	74.2
2	0.474	0.270	27.7	75.3
3	0.472	0.269	27.8	75.6
4	0.469	0.265	27.4	75.8
5	0.468	0.264	27.1	76.0
6	0.467	0.262	27.0	76.1
7	0.464	0.260	26.7	76.4
8	0.463	0.260	26.7	76.5
9	0.461	0.257	26.4	76.7
10	0.462	0.257	26.4	76.6
Average	0.468	0.264	27.2	75.9
Median	0.467	0.263	27.0	76.0
Std-dev	0.007	0.007	0.8	0.7
Minimum	0.461	0.257	26.4	74.2
Maximum	0.484	0.279	29.2	76.7

* estimators = 1000.

MAPE, and R^2 for the DS1-UCI data set. Fig. 18 shows the box-plots for the RMSE of the LSTM, XGBoost, and RF models.

Despite LSTM's lower average and median RMSE (respectively 0.461 and 0.462), according to the Kruskal-Wallis test there is no statistically significant difference between the LSTM, XGBoost, and RF algorithms' RMSE for DS1-UCI data set ($p = 0.0718$), showing a tendency of lower LSTM results, but without the confirmation of statistical tests.

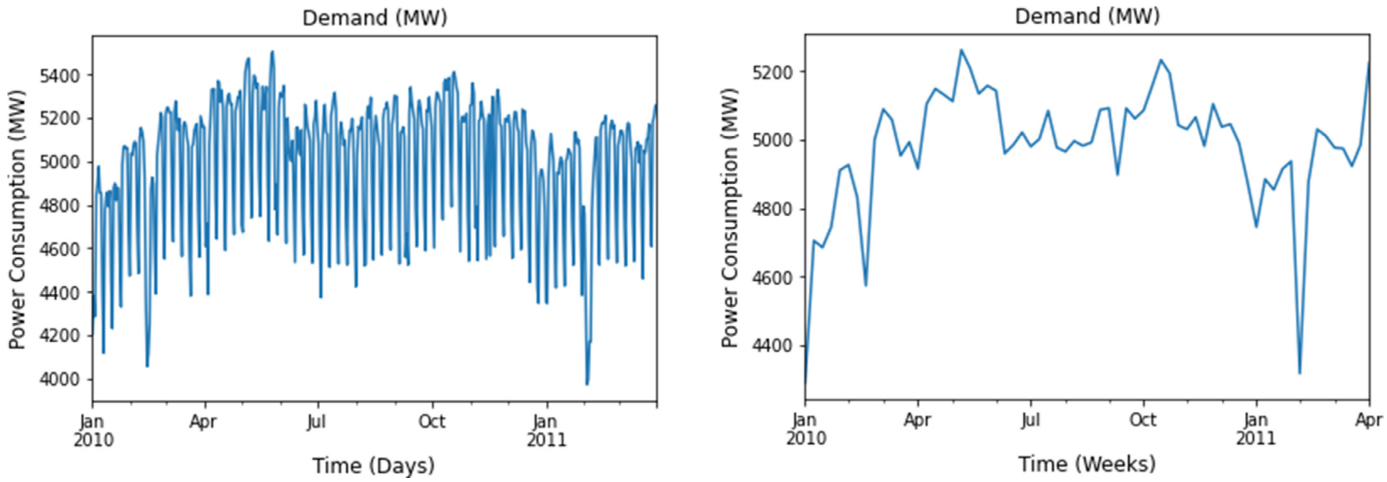


Fig. 12. Samples of the electrical consumption data from the DS3-SED dataset.

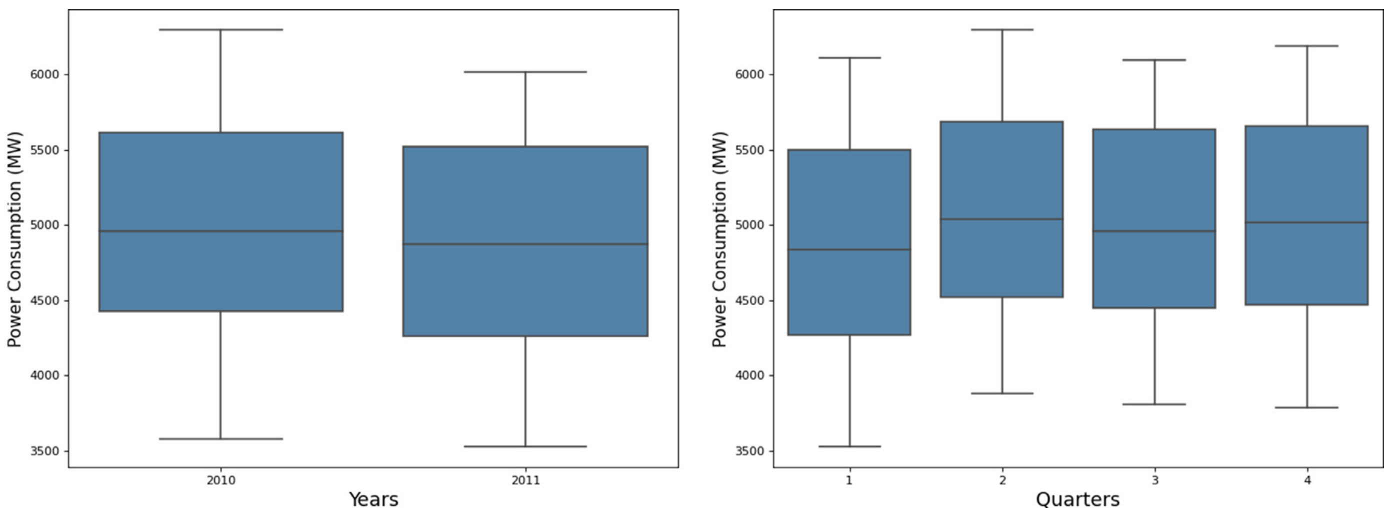


Fig. 13. DS3-SED dataset box-plots of electric load (Yearly and Quarterly).

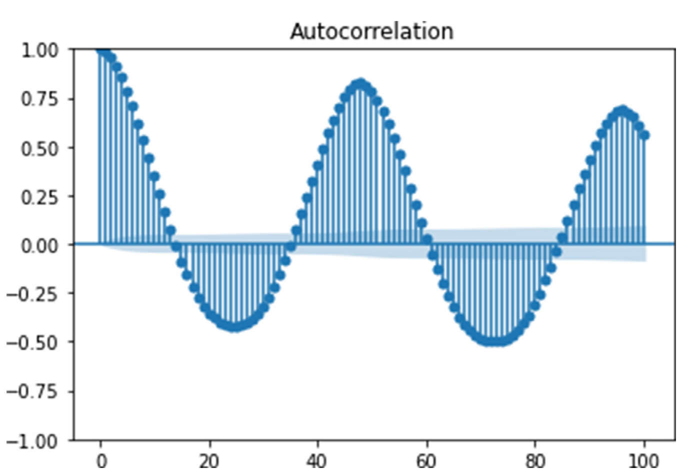


Fig. 14. Autocorrelation Function - DS3-SED dataset.

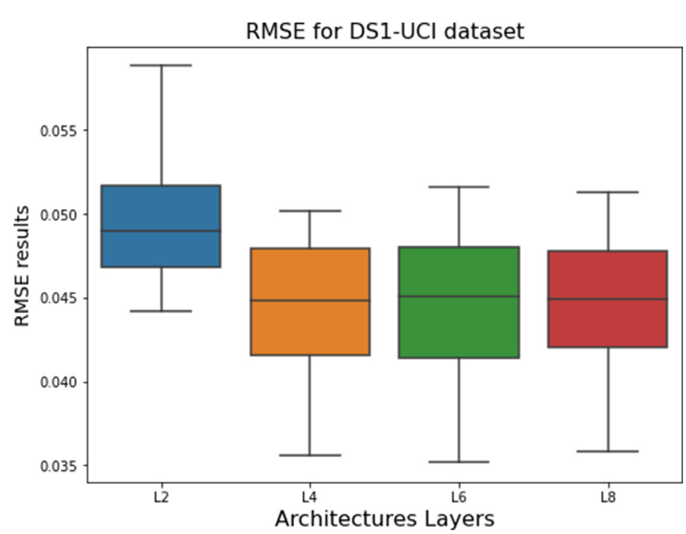


Fig. 15. Boxplots of the RMSE results obtained for the DS1-UCI dataset.

DS2-LABIC data set

Tables 6–8, respectively show the test scores for LSTM, XGBoost, and RF, containing each one of the 10-fold TS-CV metrics RMSE, MAE, MAPE, and R^2 for the DS2-LABIC data set. Fig. 19 shows the box-plots for the RMSE of the LSTM, XGBoost, and RF models.

According to the results of the Kruskal-Wallis test for DS2-LABIC, there is statistically significant difference between the LSTM, XGBoost, and RF algorithms ($p < 0.0001$). Dunn's test indicated the statistically

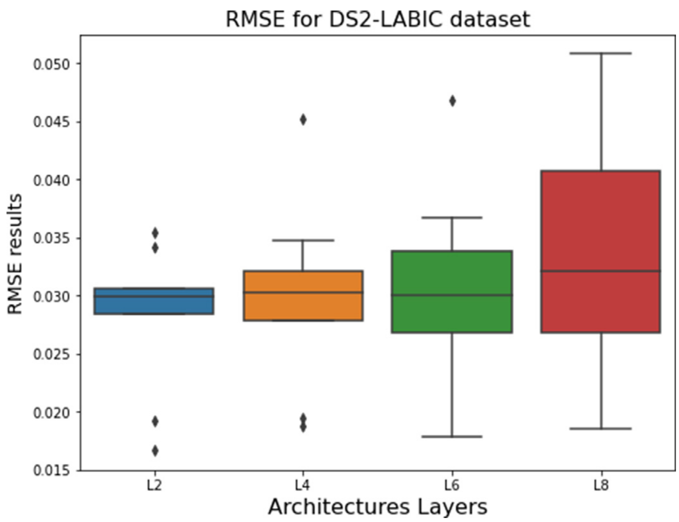


Fig. 16. Boxplots of the RMSE results obtained for the DS2-LABIC dataset.

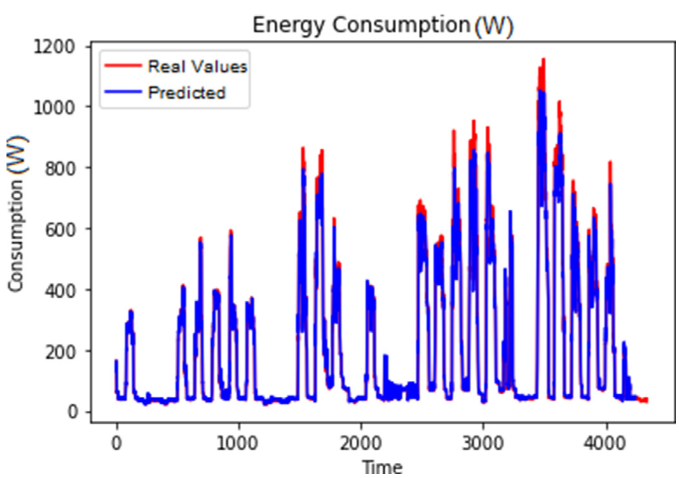


Fig. 17. Result of the LSTM Network run (observations of DS2-LABIC dataset not used for training - August 2019).

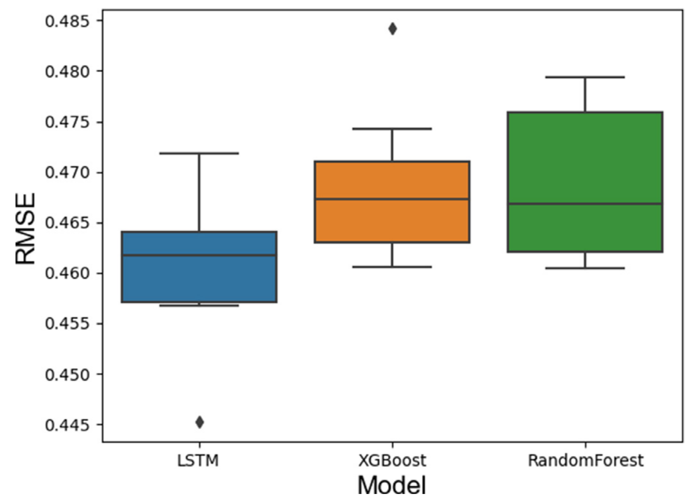


Fig. 18. Box-plots of the RMSE obtained with the LSTM, XGBoost, and RF models for the DS1-UCI.

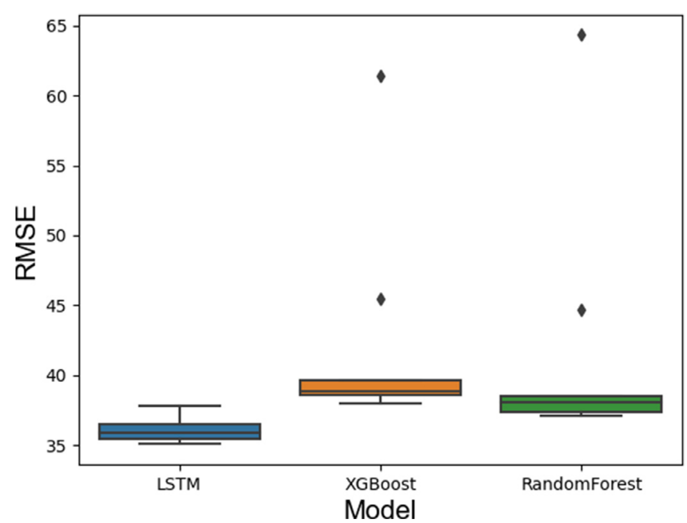


Fig. 19. Box-plots of the RMSE obtained with the LSTM, XGBoost, and RF models for the DS2-LABIC.

significant difference between LSTM in relation to both RF and XGBoost algorithms, as shown in Table 9, and the LSTM outperformed the other algorithms for the case tested (DS2-LABIC data set).

6.2.4. *Summary of results for the train-test experiment (DS1-UCI and DS2-LABIC data sets)*

Table 10 shows the average test scores consolidated for LSTM, XGBoost, and RF models, for both DS1-UCI and DS2-LABIC data sets, based on Tables 4–9 data.

6.2.5. *"Unknown" data test comparison of LSTM, XGBoost, and random forest results (DS3-SED data set)*

Tables 11–13, respectively show the test scores for LSTM, XGBoost, and RF, containing each one of the 10-fold TS-CV metrics RMSE, MAE, MAPE, and R^2 for the DS3-SED data set, which was not used for training. Fig. 20 shows the box-plots for the RMSE of the LSTM, XGBoost, and RF models.

According to the results of the Kruskal-Wallis test, there is statistically significant difference between the algorithms' RMSE metrics for the DS3-SED data set ($p < 0.0001$). Dunn's test indicated the statistically significant difference between LSTM in relation to both RF and XGBoost algorithms, as shown in Table 14, and the LSTM outperformed both algorithms for the case tested (DS3-SED data set).

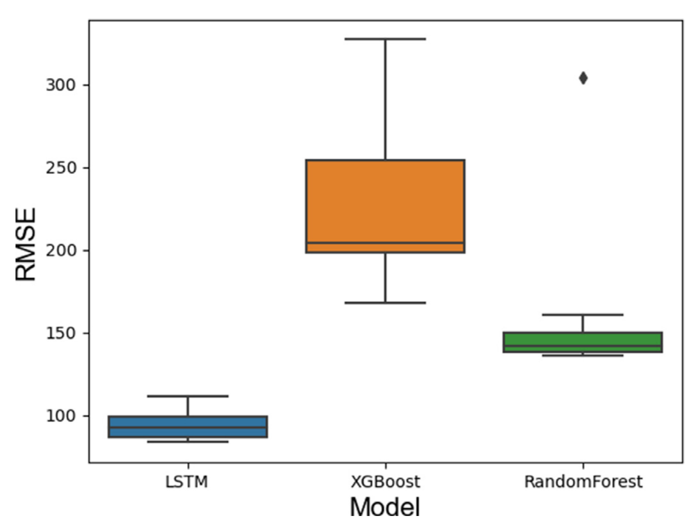


Fig. 20. Box-plots of the RMSE obtained with the LSTM, XGBoost, and RF models for the DS3-SED data set.

Table 5
RF results for DS1-UCI dataset.

k (TS-CV)	DS1-UCI RF Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	0.479	0.264	26.0	74.8
2	0.477	0.262	25.8	75.0
3	0.477	0.262	25.6	75.0
4	0.473	0.259	25.6	75.4
5	0.469	0.257	25.1	75.8
6	0.464	0.255	25.3	76.3
7	0.462	0.256	25.7	76.6
8	0.461	0.258	25.9	76.6
9	0.463	0.256	25.5	76.4
10	0.461	0.256	25.6	76.7
Average	0.469	0.259	25.6	75.9
Median	0.467	0.258	25.6	76.1
Std-dev	0.007	0.003	0.3	0.8
Minimum	0.461	0.255	25.1	74.8
Maximum	0.479	0.264	26.0	76.7

Table 6
LSTM for DS2-LABIC dataset.

k (TS-CV)	DS2-LABIC LSTM Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	35.517	17.122	10.5	97.9
2	36.167	21.485	22.6	97.8
3	37.761	18.137	10.1	97.6
4	35.388	17.387	12.6	97.9
5	35.348	17.956	12.8	97.9
6	37.509	21.297	18.4	97.6
7	35.843	17.498	11.4	97.8
8	35.883	18.915	14.8	97.8
9	36.544	19.617	14.9	97.8
10	35.058	17.778	12.7	97.9
Average	36.102	18.719	14.1	97.8
Median	35.863	18.047	12.7	97.8
Std-dev	0.916	1.590	3.8	0.1
Minimum	35.058	17.122	10.1	97.6
Maximum	37.761	21.485	22.6	97.9

Table 7
XGBoost for DS2-LABIC dataset.

k (TS-CV)	DS2-LABIC XGBoost* Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	61.420	27.576	13.7	93.7
2	45.504	21.206	13.3	96.5
3	39.425	19.724	12.9	97.4
4	39.666	19.142	11.6	97.4
5	38.766	18.571	11.1	97.5
6	38.468	18.580	11.1	97.5
7	38.034	18.334	10.9	97.6
8	37.979	18.025	10.5	97.6
9	38.823	18.442	10.3	97.5
10	38.931	18.401	10.4	97.5
Average	41.702	19.800	11.6	97.0
Median	38.877	18.576	11.1	97.5
Std-dev	7.264	2.886	1.3	1.2
Minimum	37.979	18.025	10.3	93.7
Maximum	61.420	27.576	13.7	97.6

* estimators = 500.

Table 8
Random Forest for DS2-LABIC dataset.

k (TS-CV)	DS2-LABIC RF Test Scores			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
1	64.347	29.863	16.0	93.0
2	44.696	20.867	11.4	96.6
3	37.944	18.799	10.8	97.6
4	37.379	18.193	10.3	97.7
5	37.061	18.142	10.6	97.7
6	37.437	18.561	11.4	97.6
7	37.215	18.200	11.0	97.7
8	38.132	18.358	10.8	97.6
9	38.516	18.364	10.6	97.5
10	38.427	18.366	10.6	97.5
Average	41.115	19.771	11.3	97.1
Median	38.038	18.365	10.8	97.6
Std-dev	8.463	3.637	1.7	1.4
Minimum	37.061	18.142	10.3	93.0
Maximum	64.347	29.863	16.0	97.7

Table 9
Dunn's post-hoc test for DS2-LABIC data set.

Comparison number	Group comparisons	Difference in average ranks	Cutoff at alpha=0.05	Significance difference = **
1	LSTM-RF	10.9	9.42511	**
2	LSTM-XG	16.7	9.42511	**
3	RF-XGB	5.8	9.42511	

7. Discussion

In the first experiment (preliminary tests), we assessed LSTM network models with different number of layers for consumption prediction on two datasets (DS1-UCI and DS2-LABIC) using the TS-CV. The number of layers did not cause a statistically significant difference between the models, and we chose the model L2 for conducting further tests, which had the lowest RMSE.

Another point to be highlighted is the comparison to the results obtained in the validation presented in the work of Silva et al. [29], which also used the DS2-LABIC dataset, although without the use of TS-CV to find the best network architecture. In such work, average values for ten runs were RMSE = 0.0410 and MAE = 0.0263. However, when searching for the best models with TS-CV, the metric values were RMSE = 0.0289 and MAE = 0.0205, presenting lower prediction error values. Thus the application of TS-CV for searching the best models allowed the improvement the overall LSTM performance.

Regarding the comparison of LSTM to the ML algorithms XGBoost and RF with TS-CV, for the DS1-UCI data set, it was shown a slight tendency of better results of LSTM although not confirmed with statistical tests. Conversely, a better performance of the LSTM model was confirmed statistically for DS2-LABIC data set.

Table 10
Consolidated table with average test scores of LSTM, XGBoost, and RF models for DS1-UCI and DS2-LABIC data sets.

Models	DS1-UCI dataset - Test scores (Average)				DS2-LABIC dataset - Test scores (Average)			
	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)	RMSE (W)	MAE (W)	MAPE (%)	R^2 (%)
XGBoost*	0.468	0.264	27.2	75.9	41.702	19.800	11.6	97.0
Random Forest	0.469	0.259	25.6	75.9	41.115	19.771	11.3	97.1
LSTM	0.461	0.253	27.2	76.0	36.102	18.719	14.1	97.8

* For DS1-UCI, XGBoost estimators = 1000 and for DS2-LABIC, XGBoost estimators = 500.

Table 11
LSTM for DS3-SED dataset.

k (TS-CV)	DS3-SED LSTM Test Scores			
	RMSE (MW)	MAE (MW)	MAPE (%)	R^2 (%)
1	84.569	66.561	1.4	98.3
2	94.487	72.511	1.4	97.9
3	100.646	81.925	1.7	97.6
4	94.270	72.275	1.5	97.9
5	91.165	68.292	1.4	98.1
6	111.197	84.924	1.6	97.1
7	87.165	69.484	1.4	98.2
8	86.583	69.484	1.4	98.3
9	101.615	76.413	1.5	97.6
10	83.729	60.984	1.2	98.4
Average	93.543	72.285	1.4	97.9
Median	92.717	70.880	1.4	98.0
Std-dev	8.821	7.164	0.1	0.4
Minimum	83.729	60.984	1.2	97.1
Maximum	111.197	84.924	1.7	98.4

Table 12
XGBoost for DS3-SED dataset.

k (TS-CV)	DS3-SED XGBoost* Test Scores			
	RMSE (MW)	MAE (MW)	MAPE (%)	R^2 (%)
1	327.069	243.296	4.5	75.1
2	167.770	122.405	2.4	93.4
3	198.113	153.103	2.9	90.9
4	191.727	147.914	2.9	91.4
5	207.963	160.920	3.1	89.9
6	200.712	154.539	3.0	90.6
7	222.423	181.283	3.5	88.5
8	200.495	154.465	3.0	90.6
9	264.528	212.426	4.1	83.7
10	296.167	228.821	4.4	79.6
Average	227.697	175.917	3.4	87.4
Median	204.338	157.729	3.1	90.3
Std-dev	51.170	39.456	0.7	6.0
Minimum	167.770	122.405	2.4	75.1
Maximum	327.069	243.296	4.5	93.4

* estimators = 500.

Table 13
Random Forest for DS3-SED dataset.

k (TS-CV)	DS3-SED RF Test Scores			
	RMSE (MW)	MAE (MW)	MAPE (%)	R^2 (%)
1	304.108	216.217	4.0	78.5
2	135.990	107.247	2.1	95.7
3	136.536	109.375	2.2	95.7
4	137.500	109.059	2.2	95.6
5	144.186	113.207	2.2	95.2
6	142.799	113.476	2.2	95.3
7	140.273	109.836	2.2	95.4
8	140.715	109.920	2.2	95.4
9	160.574	124.877	2.5	94.0
10	151.704	117.890	2.3	94.6
Average	159.438	123.110	2.4	93.5
Median	141.757	111.563	2.2	95.3
Std-dev	51.395	33.129	0.6	5.3
Minimum	135.990	107.247	2.1	78.5
Maximum	304.108	216.217	4.0	95.7

Table 14
Dunn's post-hoc test for DS3-SED data set.

Comparison number	Group comparisons	Difference in average ranks	Cutoff at alpha=0.05	Significance difference = **
1	LSTM-RF	10.9	9.42511	**
2	LSTM-XGB	19.1	9.42511	**
3	RF-XGB	8.2	9.42511	

For the "unknown" observations of DS3-SED the LSTM model statistically outperformed XGBoost and RF, corroborating the good performance of forecasting "unknown" data illustrated in Fig. 17, for the DS2-LABIC data set. Therefore, we can conclude that LSTM DNN has a good response in predicting energy consumption TS, given that the forecasting results in the datasets, showing the LSTM ability.

Finally, it is important to highlight that this forecasting module with LSTM networks is coupled to the EnergySaver framework [29]. This framework performs from the collection of consumption data in equipments and buildings to the display of data in real time in the web application. Thus, a continuous monitoring system of energy consumption provides several advantages, including the possibility of decision making for reducing expenses and carbon footprint.

8. Conclusion

In the present work, we took a closer look to the statistical performance of LSTM NNs for electric energy consumption in the context of an IoT system. The results corroborate to the well-known LSTM accurate performance, with statistical validation. We analyzed the robustness of the LSTM regarding a different number of layers, in two different data sets, for a TS prediction module of the software EnergySaver. We also contributed for the TS forecasting of electric consumption in a building located at a city at the Brazilian Amazon region, with peculiar climate and weather characteristics, since works on such matter are scarce. The LSTM NNs reached an outstanding generalization of capacity with data that was not used in the training phase, even for observations of a different TS.

Declaration of Competing Interest

Silva D. G. reports financial support was provided by Conselho Nacional de Pesquisa (CNPq).

Acknowledgments

D.G.S. and M.S.S.M. acknowledge Conselho Nacional de Pesquisa (CNPq) for financial support. D.G.S. and A.A.M.M. acknowledge the Banco da Amazônia S.A. (BASA) for partially supporting this research (Project "Energy Monitoring System using Internet of Things, Big Data and Machine Learning"). The authors would like to thank the reviewers for their valuable comments and suggestions.

References

[1] Empresa de Pesquisa Energética, "Balanço Energético Nacional 2020: Ano base 2019," Rio de Janeiro : EPE, pp. 7–295, 2020.
[2] D. Serpanos, M. Wolf, Internet-of-Things (IoT) Systems, Springer International Publishing, Cham, 2018, doi:10.1007/978-3-319-69715-4.
[3] K.S. Wong, M.H. Kim, Privacy protection for data-driven smart manufacturing systems, Int. J. Web Serv. Res. 14 (3) (2017) 17–32 Jul., doi:10.4018/IJWSR.2017070102.
[4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444 May, doi:10.1038/nature14539.
[5] F. Chollet, Deep Learning with Python. 2018.
[6] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780 Nov., doi:10.1162/neco.1997.9.8.1735.
[7] R.J. Frank, N. Davey, S.P. Hunt, Time series prediction and neural networks, J. Intell. Robot. Syst. Theory Appl. 31 (1–3) (2001) 91–103, doi:10.1023/A:1012074215150.
[8] T. Hong, Energy forecasting: past, present and future, Foresight Int. J. Appl. Forecast. 32 (2013) 43–48.
[9] A. Arjomandi-Nezhad, A. Ahmadi, S. Taheri, M. Fotuhi-Firuzabad, M. Moeini-Aghtaie, M. Lehtonen, Pandemic-aware day-ahead demand forecasting using ensemble learning, IEEE Access 10 (2022) 7098–7106, doi:10.1109/ACCESS.2022.3142351.
[10] A. Al Mamun, M. Sohel, N. Mohammad, M.S. Haque Sunny, D.R. Dipta, E. Hossain, A comprehensive review of the load forecasting techniques using single and hybrid predictive models, IEEE Access 8 (2020) 134911–134939, doi:10.1109/ACCESS.2020.3010702.
[11] G. Gross, F.D. Galiana, Short-term load forecasting, Proc. IEEE 75 (12) (1987) 1558–1573, doi:10.1109/PROC.1987.13927.

[12] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, H. Zareipour, Energy forecasting: a review and outlook, IEEE Open Access J. Power Energy 7 (2020) 376–388 October, doi:10.1109/OAJPE.2020.3029979.

[13] H.S. Hippert, C.E. Pedreira, R.C. Souza, Neural networks for short-term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55, doi:10.1109/59.910780.

[14] S. Haykin, Neural Networks: a Comprehensive Foundation, 2nd ed., Pearson, 1998.

[15] H.X. Zhao, F. Magoulès, A review on the prediction of building energy consumption, Renew. Sustain. Energy Rev. 16 (6) (2012) 3586–3592, doi:10.1016/j.rser.2012.02.049.

[16] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[17] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064, doi:10.1016/j.ins.2009.12.010.

[18] T. Hou, et al., A novel short-term residential electric load forecasting method based on adaptive load aggregation and deep learning algorithms, Energies 14 (22) (2021), doi:10.3390/en14227820.

[19] A. Shaqour, T. Ono, A. Hagishima, H. Farzaneh, Electrical demand aggregation effects on the performance of deep learning-based short-term load forecasting of a residential building, Energy AI 8 (2022) 100141 November 2021, doi:10.1016/j.egyai.2022.100141.

[20] I. Ozer, S.B. Efe, H. Ozbay, A combined deep learning application for short term load forecasting, Alexand. Eng. J. 60 (4) (2021) 3807–3818, doi:10.1016/j.aej.2021.02.050.

[21] S.H. Rafi, N. Al-Masood, S.R. Deeba, E. Hossain, A short-term load forecasting method using integrated CNN and LSTM network, IEEE Access 9 (2021) 32436–32448, doi:10.1109/ACCESS.2021.3060654.

[22] B. Farsi, M. Amayri, N. Bouguila, U. Eicker, On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach, IEEE Access 9 (2021) 31191–31212, doi:10.1109/ACCESS.2021.3060290.

[23] A. Ahmadi, M. Nabipour, B. Mohammadi-Ivatloo, A.M. Amani, S. Rho, M.J. Piran, Long-term wind power forecasting using tree-based learning algorithms, IEEE Access 8 (2020) 151511–151522, doi:10.1109/ACCESS.2020.3017442.

[24] A.F.G. de Sousa, H.C.M. Furtado, W.N. Macêdo, A.A.D.M. Meneses, Analysis of artificial neural network point forecasting models and prediction intervals for solar irradiance estimation, Am. J. Eng. Appl. Sci. 13 (3) (2020) 347–357 Mar., doi:10.3844/ajeassp.2020.347.357.

[25] A. Aswani, N. Master, J. Taneja, D. Culler, C. Tomlin, Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control, Proc. IEEE 100 (1) (2012) 240–253 Jan., doi:10.1109/JPROC.2011.2161242.

[26] A. Ruano, S. Silva, H. Duarte, P.M. Ferreira, Wireless sensors and IoT platform for intelligent HVAC control, Appl. Sci. 8 (3) (2018) 370 Mar., doi:10.3390/app8030370.

[27] P. Moura, J.I. Moreno, G. López, M. Alvarez-Campana, IoT platform for energy sustainability in university campuses, Sensors 21 (2) (2021) 357 Jan., doi:10.3390/s21020357.

[28] M.T.B. Geller, A.A.D.M. Meneses, Modelling IoT systems with UML: a case study for monitoring and predicting power consumption, Am. J. Eng. Appl. Sci. 14 (1) (2021) 81–93 Jan., doi:10.3844/ajeassp.2021.81.93.

[29] D. da Silva, M. Geller, M. Moura, and A.A.M. Meneses, "A deep learning prediction module for the IoT system EnergySaver for monitoring and estimating power consumption," 2021.

[30] G. Reggio, A UML-based proposal for IoT system requirements specification, in: Proceedings of the 10th International Workshop on Modelling in Software Engineering - MiSE '18, 2018, pp. 9–16, doi:10.1145/3193954.3193956.

[31] K. Thramboulidis, F. Christoulakis, UML4IoT—a UML-based approach to exploit IoT in cyber-physical manufacturing systems, Comput. Ind. 82 (2016) 259–272, doi:10.1016/j.compind.2016.05.010.

[32] D.A. Robles-Ramirez, P.J. Escamilla-Ambrosio, T. Tryfonas, IoTsec: UML extension for internet of things systems security modelling, in: Proceedings of the 10th International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), 2017, pp. 151–156, doi:10.1109/ICMEAE.2017.20. Nov..

[33] Y. Sun, P. Wang, S. Zhai, D. Hou, S. Wang, Y. Zhou, Ultra short-term probability prediction of wind power based on LSTM network and condition normal distribution, Wind Energy 23 (1) (2020) 63–76 Jan., doi:10.1002/we.2414.

[34] T. Chen and T. He, "XGBoost : eXtreme gradient boosting," R package version 0.71-2. pp. 1–4, 2015.

[35] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi:10.1145/2939672.2939785. Aug..

[36] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[37] G. Iannace, G. Ciaburro, A. Trematerra, A wind turbine noise prediction using random forest regression, Machines (2019), doi:10.3390/machines7040069.

[38] A.F.G. de Sousa, H.C.M. Furtado, and A.A.D.M. Meneses, "Estimation of prediction intervals for time series by neural network constructed with particle swarm optimization," 2017.

[39] T.Y. Kim and S.B. Cho, "Predicting the household power consumption using CNN-LSTM hybrid networks," 2018, pp. 481–490. doi: 10.1007/978-3-030-03493-1_50.

[40] T. Le, M.T. Vo, B. Vo, E. Hwang, S. Rho, S.W. Baik, Improving electric energy consumption prediction using CNN and Bi-LSTM, Appl. Sci. 9 (20) (2019) 4237 Oct., doi:10.3390/app9204237.

[41] N. Khan, I.U. Haq, S.U. Khan, S. Rho, M.Y. Lee, S.W. Baik, DB-Net: a novel dilated CNN based multi-step forecasting model for power consumption in integrated local energy systems, Int. J. Electr. Power Energy Syst. 133 (2021) 107023 Dec., doi:10.1016/j.ijepes.2021.107023.

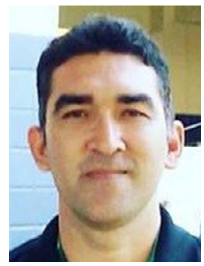
[42] P. Dangeti, Statistics for machine learning. Packt.

[43] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, Int. Jt. Conf. Artif. Intell. (1995) June.

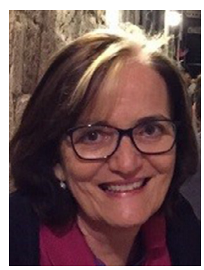
[44] R. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed., OTexts, Melbourne, Australia, 2021 [Online]. Available: OTexts.com/fpp3.

[45] I. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. 2005.

[46] A. Dmitrienko, C. Chuang-Stein, R. D'Agostino, Pharmaceutical Statistics using SAS, 53, SAS Publishing, 2007.



Davi Guimarães da Silva is a doctorate student at the Federal University of Western Pará (UFOPA-2019). Master in Informatics from the Federal University of Amazonas (UFAM - 2016). Specialization in Teaching for Vocational Education (2018). Specialization in Systems Engineering (2013). MBA Specialization in Environment (2011). Graduated in Data Processing Technology from the Federal University of Pará (UFPA - 2009). He is currently Professor at the Federal Institute of Education Science and Technology of Pará (IFPA). Areas of interest: Deep Learning (Time Series Prediction), Energy (Computational Intelligence applied to monitoring electrical energy consumption), Database and Information Retrieval area, Machine Learning, Data mining, Information Technologies applied to Education, Virtual Learning Objects, Augmented Reality Applied to Education.



Marla Teresinha Barbosa Geller: Graduated in Informatics from Instituto Luterano de Ensino Superior de Santarém (2000) and master's at Electric Engineering from Universidade Federal do Pará (2006). She has experience in Computer Science, acting on the following subjects: software engineering, object oriented programming, databases, software processes, and UML.

Mauro Sergio dos Santos Moura: Bachelor of Science and Technology from Federal University of Western Pará (UFOPA, 2020). Currently he is an undergraduate student in Physical Engineering at UFOPA. He has experience in the areas of programming with Python and C++, related to the Internet of Things (IoT), parallel programming with CUDA, Artificial Intelligence/Machine Learning/Deep Learning for Energy Efficiency. Currently he develops Deep Learning solutions for computer vision for segmentation of biomedical images.

Anderson Alvarenga de Moura Meneses: Graduated in Physics from the Federal University of Rio de Janeiro (2000), Master (2005) and Doctorate (2010) in Nuclear Engineering from COPPE/Federal University of Rio de Janeiro, with a fellowship (2009) at the Dalle Molle Institute for Artificial Intelligence (IDSIA, University of Lugano, Switzerland). Specialist in Analysis, Design and Systems Management at Pontifical Catholic University-Rio (2008). Coordinator of the Computational Intelligence Laboratory (LabIC/UFOPA). Leader of the Computational Intelligence and Optimization research group. Associate Professor at the Federal University of Oeste do Pará (UFOPA) and permanent member of the Postgraduate Program in Natural Resources in the Amazon (PPGRNA/UFOPA). Collaborating member of the Graduate Program in Society, Nature and Development (PPGSND/UFOPA). Areas of interest: Nuclear Engineering (optimization of reactor reload), Deep Learning (Time Series Prediction and Image Processing) and Energy (Computational Intelligence applied to electrical energy consumption monitoring.)