Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0023

Hourly energy consumption forecasting by LSTM and ARIMA methods

Ali Nasser Ali, Taha Etem
Department of Computer Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkiye

Cite this article as: Ali, A.N., & Etem, T. (2025). Hourly energy consumption forecasting by LSTM and ARIMA methods. *J Comp Electr Electron Eng Sci,* 3(1),14-20.

Received: 23.10.2024 ◆ **Accepted**: 08.01.2025 ◆ **Published**: 17.04.2025

ABSTRACT

Energy consumption represents the overall quantity of energy necessary for a specific activity or process, typically quantified in kilowatt-hours (kWh). In 2019, global electricity final consumption amounted to 22.848 terawatt-hours (TWh), indicating a 1.7% growth compared to 2018. During the same period, the Organization for Economic Co-operation and Development (OECD) recorded a total electricity final consumption of 9.672 TWh, reflecting a 1.1% decrease from the previous year. Conversely, non-OECD countries experienced a rise in final electricity consumption, reaching 13.176 TWh, marking a 3.8% increase over the 2018 figures. Time series analysis is a statistical method that examines data collected over successive time intervals. By identifying patterns and trends in historical data, this approach facilitates predictions and forecasts about future values. In our problem, forecasting and estimating the energy consumption in megawatts for the west regions of USA have achieved a good performance, with 97% for the R-squared metrics for LSTM and 98% for ARIMA models.

Keywords: Energy consumption, time series analysis, long short-term memory, LSTM, auto-regressive integrated moving average, ARIMA

INTRODUCTION

Energy consumption refers to the total amount of energy used over a specified period, typically measured in units like kilowatt-hours (KWh) or joules. It encompasses the utilization of various energy sources, such as electricity, fuel, or renewable resources, for activities ranging from household operations and indusratrial processes to transportation. Understanding and monitoring energy consumption are crucial for assessing resource efficiency, environmental impact, and the development of sustainable practices (Energy consumption: definition & types | StudySmarter, n.d.).

Economic trends indicate a significant deceleration in the global growth of energy consumption in 2022, with the growth rate dropping by half, from +4.9% in 2021 to 2.1% in 2022. Despite this slowdown, the 2022 growth rate remains above the average observed during the period from 2010 to 2019, which was +1.4% per year. In 2022, the growth of energy consumption decelerated in the world's two largest consuming countries. China, holding the position of the largest energy consumer globally with a 25% share in 2022, experienced a 3% increase, down from the +5.2% growth observed in 2021. Similarly, the United States saw a rise of 1.8%, compared to the +4.9% growth in the previous year. Noteworthy expansions in energy consumption were driven by robust economic growth in India (+7.3%), Indonesia (+21%), and Saudi Arabia (+8.4%). Additionally, there were moderate increases in Canada (+3.8%) and Latin America (+2.7%), including gains in Brazil and Mexico (+2.4%), and a more substantial rise in Argentina (+4.5%). The Middle East and Africa also experienced a collective growth of around 3%, despite a 4.5% decline in South Africa attributed to coal supply tensions and mandated load shedding in the power sector (World Energy Consumption Statistics | Enerdata, n.d.).

In 2021, Halit Çetiner and İbrahim Çetiner employed various machine learning algorithms based on artificial intelligence to predict energy consumption. The models implemented in their work include XGBoost, LSTM algorithms, classical linear regression, and RANSAC. Among the models they used, the LSTM showed good performance in all the measuring metrics with an r-squared of 99.3% (Çetiner & Çetiner, 2021).

In 2021, Wei Qin, Linhong Wang, Yuhan Liu and Cheng Xu developed a support vector machine regression (SVR) model using variables like state of charge (SOC) and environmental factors. The grey wolf optimization (GWO) algorithm was used to select optimal parameters, resulting in the proposed GROW-SVR model. Validation on real bus routes showed a mean average percentage error of 14.47% and mean average error of 0.7776. The model also outperformed other methods in both accuracy and training time (Qin et al., 2021).

In 2022, Mohammed Nasih Ismael employed a variety of artificial intelligence-based regression models to predict energy consumption. The implemented models in their study

Corresponding Author: Taha Etem, tahaetem@karatekin.edu.tr

encompass LSTM for the prediction. The model they used demonstrated superior performance with an r-squared of 95% (Ismael, 2022).

In 2023, Wei Cai, Xiaodong Wen, Chaoen Li, Jingjing Shao and Jianguo Xu used the SVR-supervised machine learning algorithms, which offer strong predictive accuracy but lacks a defined rule for parameter fitting. To solve this, they explored six meta-heuristic optimization algorithms and found the best fit for the SVR model. Their analysis showed the hybrid buildings, with R^2 values of 0.9975 and 0.99955 for cooling and heating load predictions, respectively, in the training data (Cai et al., 2023). In **Table 1**, a comparison with the literature is shown (Xiong et al., 2014), (Purlu et al., 2022), (Son & Van Cuong, 2023), (Irfan et al., 2023).

Our study builds upon these efforts, focusing on the prediction of energy consumption using machine learning models already existing in libraries such as statsmodels, Tensorflow and a dataset called hourly energy consumption obtained from Kaggle (10. PJME_hourly, n.d.), which provides features of the house value. While prior studies on PJME hourly electricity consumption forecasting have predominantly relied on conventional statistical methods or standard machine learning models, our research introduces a hybrid deep learning architecture integrating LSTM and easy to apply autoregressive, integrated moving average (ARIMA) model. This novel combination addresses non-linear consumption patterns and enhances the model's responsiveness to exogenous variables that are often overlooked in traditional models. By doing so, our study not only improves prediction accuracy but also contributes a framework for adaptive load forecasting that can be extended to other regional markets.

In the following section, we will provide a detailed overview of the dataset and the employed models. We will then go through the exploration and analysis of the data, explaining the methods used. Finally, we will compare the performance of the models based on metrics such as MSE, MAE, RMSE, and R2 Squared, providing insights into their effectiveness in predicting median house value.

METHODS

The method proposed for this study consists of five components, as illustrated in **Figure 1**, working collaboratively to attain the study objectives. Initially, the process involves collecting the dataset, followed by subsequent pre-processing steps. In the pre-processing steps, we checked for missing data and the presence of outliers. Then, the split-validation technique was employed to divide the dataset into training samples, encompassing data from 2002 to 2018, and the data specifically from the year 2018 for testing.

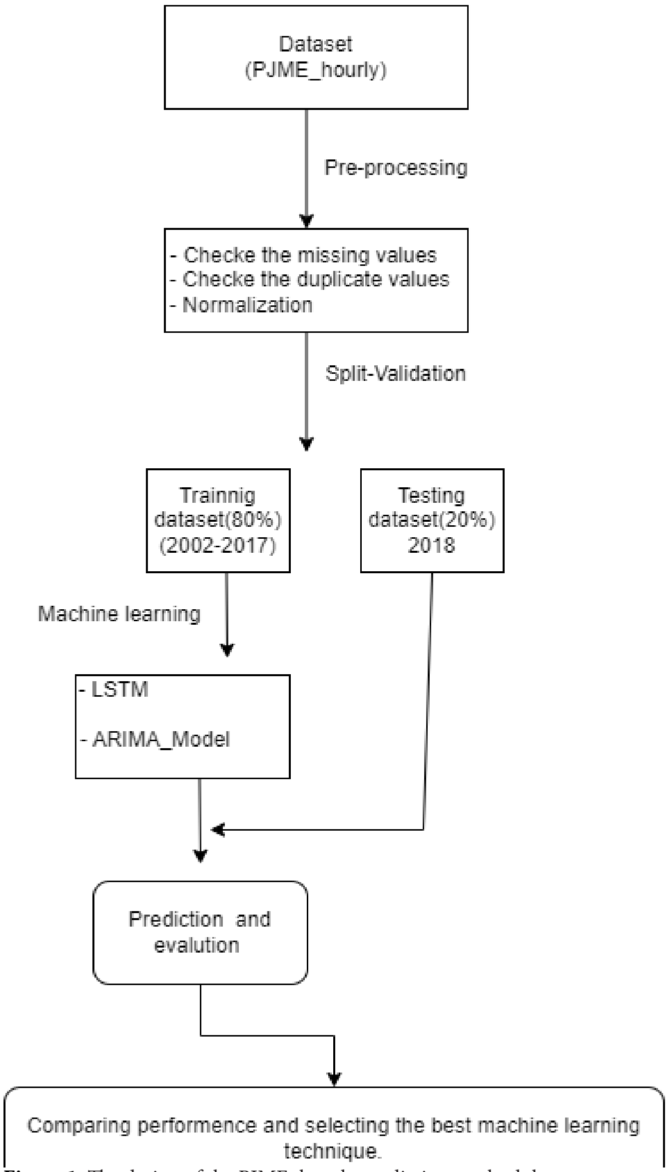

Figure 1. The design of the PJME_hourly prediction methodology

PJME: Preheated Jatropha Methyl Ester

Next, we applied specific models for time prediction, including the LSTM and ARIMA models. These algorithms were trained, tested, and evaluated to achieve the best performance. The evaluation process involved assessing key metrics such as MSE, MAE, and R-squared to comprehensively evaluate the effectiveness of each model in achieving the study objectives. However, we will focus primarily on the R-squared metric.

Table 1. Comparison with the literature

Source	Data source	Model (s) tested	Performance metrics	Advantages	Disadvantages
Xiong et al., 2014	PJM dataset	SVR, BEMD	Just Prediction Results	The lower and upper bounds are decomposed well	None of the performance metrics are evaluated.
Purlu et al., 2022	Energy consumption data	ANN	MAE, MSE, R^2	Low error rate	Potential overfitting without careful tuning
Son & Van Cuong, 2023	Kaggle PJME dataset	NARX	MSE, MAPE	Educational, demonstrates preprocessing	Limited rigorous validation
Irfan et al., 2023	Kaggle PJME dataset	Ensemble DNN	XGBoost: MAPE ~4.5%, RMSE ~900 MW**	Broad overview of methods, guidance on methodology	Excessive computational power need
This study	Kaggle PJME dataset	ARIMA, LSTM	MAE, MSE, RMSE, R^2	Addresses non-linear consumption patterns well	Dataset limitations

PJM: Pennsylvania-New Jersey-Maryland Interconnection, PJME: Preheated Jatropha Methyl Ester, SVR: Support vector machine regression, BEMD: Bi-dimensional empirical mode decomposition, ANN: Artificial neural networks, NARX: Nonlinear autoregressive exogenous, ARIMA: Autoregressive, integrated moving average, LSTM: Long short-term memory, MAE: Mean_absolute_error, MSE: Mean_squared_error, MAPE: Mean absolute percent error, RMSE: Root_mean_square_error, MW: Megawatts

PJMW Dataset

The dataset used in our study, called PJMW_hourly, stands for Pennsylvania-New Jersey-Maryland Interconnection (PJM), a regional transmission organization (RTO) in the United States, with 'W' representing the west region, and 'hourly' indicating the consumption of estimated energy in megawatts (MW) per hour. This dataset, obtained from Kaggle (10. PJME_hourly, n.d.), is a time series dataset of energy consumption from an eastern state region in the USA.

As indicated in **Table 1**, the dataset comprises 145,366 entries, with one feature and one target variable. The features of this dataset include:

PJMW_MW: Information on energy consumption in Megawatts.

Datetime: Date and time information.

Long Short-Term Memory

Long short-term memory (LSTM) networks are a type of recurrent neural network (RNN) that are well-suited for sequence prediction tasks, such as time series forecasting, because they can capture long-term dependencies in the data (Koşar & Barshan, 2023).

Before delving into LSTM, let's briefly explain the difference between recurrent neural networks (RNN) and traditional neural networks (TNN). TNNs are limited in their ability to comprehend sequences of data, such as events in a movie, because they cannot retain information from previous events to inform subsequent ones. Recurrent neural networks (RNNs) address this limitation by incorporating loops that enable information to persist from one step to the next. Essentially, RNNs can be visualized as multiple copies of the same network passing messages to each other, making them well-suited for handling sequences and lists of data. They have achieved significant success in tasks such as speech recognition, language modeling, translation, and image (Understanding LSTM Networks-Colah's Blog, n.d.).

In **Figure 2** above, a segment of a neural network denoted as a takes an input named xt and generates an output referred to as ht. A loop is present, facilitating the movement of information from one network step to the next (Understanding LSTM Networks-Colah's Blog, n.d.).

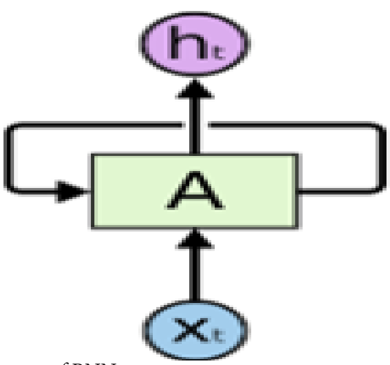

Figure 2. A segment of RNN

RNN: Recurrent neural networks

The loops within recurrent neural networks (RNNs) may appear mysterious initially, but they share similarities with regular neural networks. We can conceptualize an RNN as numerous replicas of the same network exchanging messages among themselves, as depicted in the **Figure 3** below.

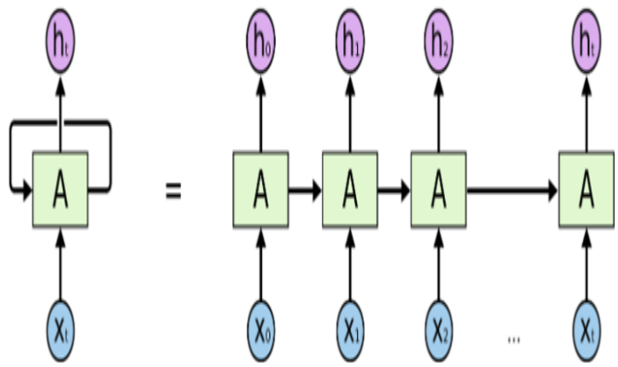

Figure 3. A general representation segment of RNN

RNN: Recurrent neural networks

When we unroll this loop, it forms a chain-like structure, as illustrated in the figure above, highlighting the suitability of RNNs for handling sequences and lists of data, which aligns with our problem statement. RNNs have demonstrated success in various domains such as speech recognition, language modeling, translation, and image captioning (Understanding LSTM Networks-Colah's Blog, n.d.).

However, RNNs are most effective when the information is recent, meaning the gap between relevant information and its application is small, as shown in the **Figure 4** below in bold blue. For instance, when predicting the next word in a sentence like 'I like to drink Turkish tea', the word 'tea' is predictable without requiring further context.

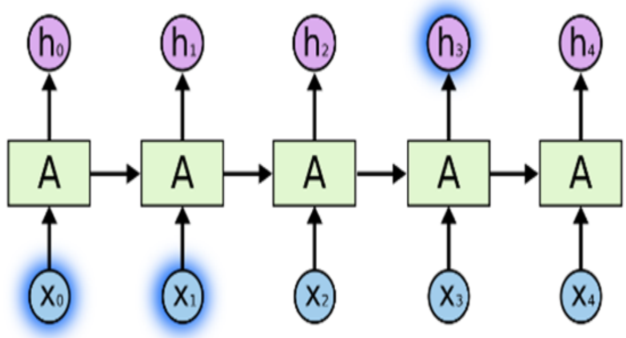

Figure 4. Example of a small gap of information in RNN

RNN: Recurrent neural networks

In certain situations, more extensive context is necessary. For instance, when predicting the last word in the text 'I study in Turkey, but I grew up in Djibouti ... I speak fluent Turkish', recent information hints that the next word might be the name of a language. However, to determine which language specifically, we require the context of 'Turkey' from earlier in the text. As the gap between the relevant information and its application grows larger, RNNs struggle to establish connections between the information, and their learning ability diminishes. Fortunately, LSTMs, a special type of RNN, do not encounter this problem because they are designed to retain information over extended periods, unlike traditional RNNs. Their architecture consists of a chain of repeating neural network modules with LSTMs featuring a more intricate structure compared to traditional RNNs.

This structure typically includes components such as gating mechanisms and memory cells. These layers include the input gate, the forget gate, the output gate, and the memory cell. Each gate and the memory cell serve specific functions within the LSTM architecture, enabling it to effectively capture and manage long-term dependencies in sequential data.

Finally, to respond to our problem, we have built a LSTM model (**Figure 5**) of a high-level neural networks API running on top of TensorFlow or other backend engines.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 50, 50)            10400

 lstm_9 (LSTM)               (None, 50, 50)            20200

 lstm_10 (LSTM)              (None, 50, 50)            20200

 lstm_11 (LSTM)              (None, 50)                20200

 dense_2 (Dense)             (None, 1)                 51

=================================================================
Total params: 71051 (277.54 KB)
Trainable params: 71051 (277.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 5. LSTM model parameters

LSTM: Long short-term memory

Sequential model indicates that we're creating a linear stack of layers in our neural network. Data flows sequentially from one layer to the next.

The first layer is an LSTM layer with 50 units. input_shape=[X_train.shape (1), 1] defines the shape of the input data. In this case, it expects input sequences with X_train.shape (1) time steps and 1 feature per timestep. There are three more LSTM layers with 50 units each. They are stacked on top of each other. The purpose of stacking multiple LSTM layers is to allow the model to learn more complex temporal patterns in the data by abstracting features at different levels of temporal abstraction. it only returns the output of the last timestep instead of the full sequence. This setup is common in sequence prediction tasks where we're interested in predicting a single value or the next value in the sequence.

dense layer is a standard fully connected dense layer with one unit. It's a typical setup for regression problems where the network is tasked with predicting a single continuous value. This layer provides the final output of the model.

In summary, the model architecture consists of a sequence of LSTM layers followed by a dense layer for prediction.

Auto-Regressive Integrated Moving Average

The ARIMA model, short for auto-regressive integrated moving average, is a powerful tool used for analyzing time series data (What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science, n.d.). It creates a mathematical equation to describe and predict patterns in our time series data. This equation has three main parts:

AR (auto-regressive): This part looks at past data points to predict future ones. It's like saying, "Today's weather is similar to yesterday's."

I (integration or differencing): This part accounts for overall trends or changes in the data over time. It's like adjusting for a gradual increase or decrease in temperature over the years.

MA (moving average): This considers errors or noise in the data based on past observations. It's like smoothing out fluctuations to see the underlying pattern.

ARIMA (p, d, q) is used to represent the ARIMA model, where each letter corresponds to a specific parameter (What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science, n.d.);

- **p:** This parameter determines the number of autoregressive (AR) terms in the model. Autoregressive terms are based on past observations, so p indicates how many past observations are considered.
- **d:** This parameter determines the order of differencing. Differencing is used to make the time series data stationary, which means removing trends or patterns that change over time. The order of differencing, represented by d, shows how many times you need to differ the data to achieve stationarity.
- **q:** This parameter determines the number of moving average (MA) terms in the model. Moving average terms represent the error terms or noise in the data based on past observations. So, q indicates how many past error terms are considered.

In order to determine the order of p for the autoregressive (AR) component and and q for the MA component of our model, we have plotted the autocorrelation (ACF) and partial autocorrelation function (PACF) as shown in **Figure 6**.

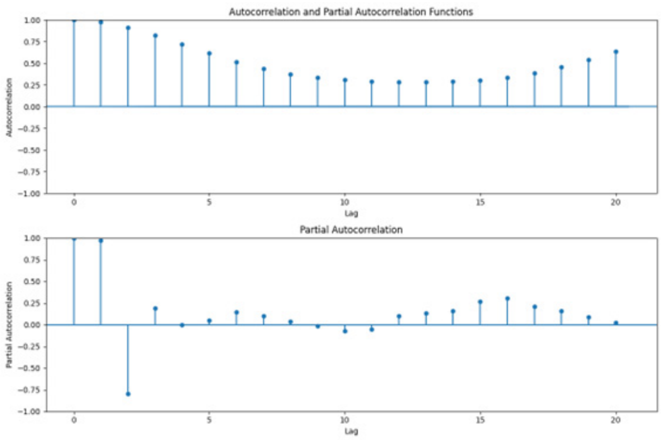

Figure 6. Autocorrelation and partial autocorrelation functions (ACF na PACF)

ACF: autocorrelation function, PACF: Partial autocorrelation function

The ACF helps identify the order of the MA component, while the PACF helps us identify the order of the AR component. Finally, to determine the order of differencing d, it involves assessing the stationarity of the time series data. We opted to compute the augmented Dickey-Fuller (ADF) test, which is a common statistical test used to determine whether a given time series is stationary or not (Bhattacharya & Burman, 2016).

The ADF test statistic is compared with critical values to determine the stationarity of the time series. We have computed the ADF statistic: -19.88829608963819 and p-value: 0.0001 since the p-value is less than the chosen significance level (usually 0.05), we conclude that the time series is stationary.

Model Design and Dataset Preparation

The process of a machine learning project typically involves several stages, from defining the problem to deploying the model. The general overview of the machine learning project lifecycle is shown in **Figure 7** below. It begins with gathering relevant data for the problem. Then, it conducts data analysis and preprocessing, including standardizing the features for equal scaling and performing feature engineering and moves on to model selection and evaluation, where an appropriate algorithm is chosen to assess its performance.

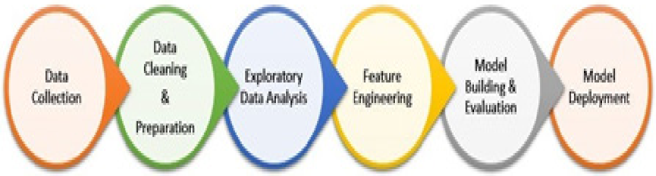

Figure 7. Steps of the proposed method

We have plotted the daily average energy consumption (**Figure 8**) and the box plot (**Figure 9**) to investigate if there are outliers that should be processed. Additionally, we have opted to standardize the values in order to improve the model training.

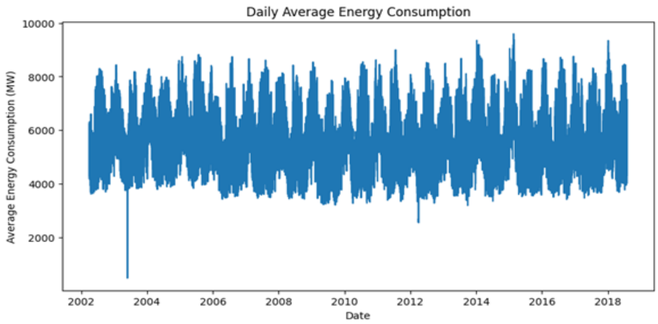

Figure 8. Daily average energy consumption

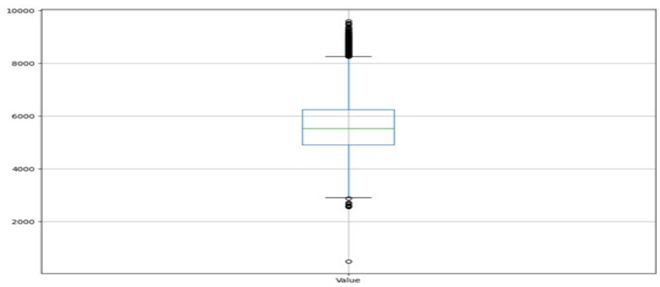

Figure 9. Box plot of the target value

The Box plot shows that there are outliers that should be preprocessed for the best performance. We split our data into training test dataset, in which the blue part as shown in **Figure 10** is the train part (from 2002 to 2018) and the green is the test part (2018).

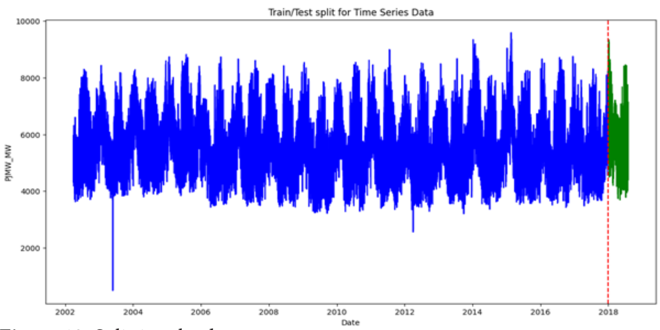

Figure 10. Splitting the dataset

In our case study, we observed that our dataset contains missing values, particularly in the 'total bedrooms' column, and numerous outliers exist in all features, including the target variable, which need to be processed. Additionally, we have created new columns based on our existing columns in order to improve model performance.

To handle the outliers in our dataset, we utilized quartiles, interquartile range (IQR), and an upper bound, all of which are statistical techniques. Firstly, quartiles divide a dataset into four equal parts: Q1, Q2, and Q3. Q1 (the first quartile) represents the 25^{th} percentile, indicating that 25% of the data falls below this value. Q2 (the second quartile) is the median, representing the middle value of the dataset when sorted in ascending order. Q3 (the third quartile) denotes the 75th percentile, meaning that 75% of the data falls below this value. Once computed, the interquartile range (IQR) is determined as the difference between Q3 and Q1: IQR=Q3-Q1. The IQR provides insight into the spread of the middle 50% of the data and is resistant to outliers, helping to compute the upper bound (Jambu, 1991).

In outlier detection, the upper bound identifies data points significantly above the typical range of values in the dataset. A common method calculates the upper bound using a multiplier (e.g., 1.5 or 3) times the IQR above Q3: upper bound=Q3+(1.5* IQR) or upper bound=Q3+(3* IQR) and lower bound=Q3-(1.5*IQR). Data points above the upper bound and below the lower bound are potential outliers and may require further investigation or treatment.

Quartiles (Q1 and Q3), the interquartile range (IQR), the upper and lower bound are calculated to identify and potentially remove outliers from all columns in a dataset. This process ensures the quality and reliability of the data for subsequent analysis or modeling tasks. We have changed the outliers by Q3, the values that are above the upper bound and Q1, the values that are below the lower bound.

As mentioned earlier, we opted to normalize the dataset using min-max scaler to enhance our model for training. Equation 1 below, depicts how min-max scaling is computed. It involves subtracting the minimum value of the dataset and then dividing by the difference between its maximum and minimum values (Wang et al., 2024).

$$\bar{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

The metrics used to assess the model performance include Mean_absolute_error (MAE), Mean_squared_error (MSE), R^2_score and Root_Mean_Square_Error (RMSE) (Lewinson, 2024). These evaluation metrics were employed to compare ensemble learning models.

The MSE measures the average squared difference between the actual and predicted values. It provides a comprehensive view of the model's overall accuracy. A lower MSE indicates a better fit, with smaller errors.

The MAE measures the average absolute difference between the actual and predicted values. It provides a more robust measure of error, as it is less sensitive to outliers than MSE.A lower MAE indicates better model performance, with smaller absolute errors.

The R-squared quantifies the proportion of the variance in the dependent variable that is explained by the independent variables. It assesses how well the model captures the variability in the data. Its values range from 0 to 1, where 1 indicates a perfect fit. Higher values which mean close to 1 suggest a better model fit.

The RMSE serves as a metric to gauge the average disparity between the predicted values of a statistical model and the actual values. It is calculated as the standard deviation of the residuals, which are the differences between the predicted values and the corresponding observed values. RMSE essentially indicates the extent of spread among these residuals, providing insight into how closely the observed data aligns with the predicted values derived from the model.

RESULTS

Regarding the LSTM model, we have visualized the actual and predicted energy consumption for August 2, 2018, allowing for a comparison between the predicted and true values of the data. As shown in Figure 11 below, the comparison between the predicted and actual values fits well, indicating that the model performs well.

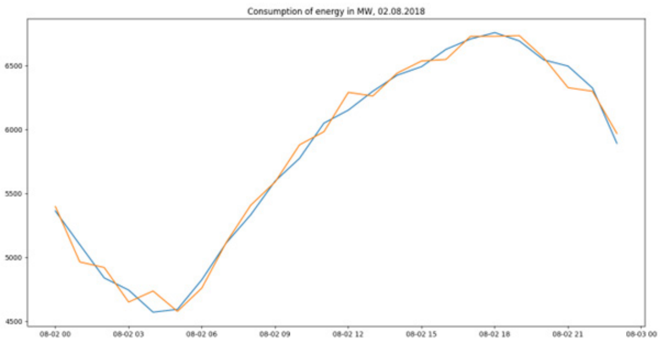

Figure 11. Comparison of the predictions and actual values for one day

The **Figure 12** below shows the actual and predicted energy consumption trends for the entire year 2018, which confirms the performance of our LSTM model.

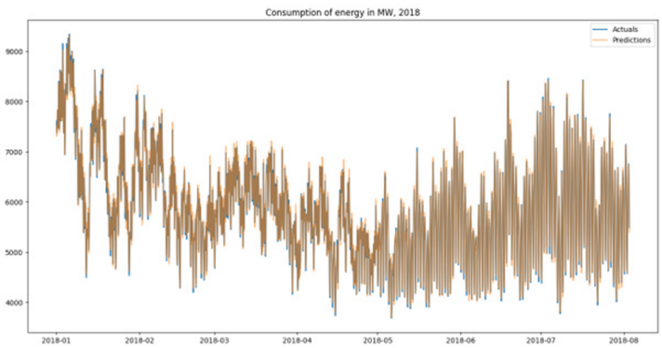

Figure 12. Comparison of the predictions and actual values for one year

The results of the metrics on the test dataset for each trained model are presented in **Table 2** below. As indicated in **Table 2**, ARIMA demonstrated good performance across all metrics: with values of 89.90, 15247.42, 123.50, and 98%, respectively. The model exhibits strong performance and indicates no overfitting, given its similarity in the metrics of the training dataset. LSTM also achieved good performance and is very close to ARIMA (**Table 3**).

Table 2. Description of PJME_hourly dataset

Number of the data	Attribute	Type
1	PJMW_MW	Float
2	Datetime	Object

PJME: Preheated Jatropha Methyl Ester, MW: Megawatts

Table 3. Metrics results of the models on test dataset

Algorithms	MAE	MSE	RMSE	R-squared
LSTM	96.06	30467.07	174.5	97.0%
ARIMA	89.90	15247.42	123.50	98.0%

LSTM: Long short-term memory, ARIMA: Autoregressive, integrated moving average, MAE: Mean_absolute_error, MSE: Mean_squared_error, RMSE: Root_Mean_Square_Error

CONCLUSION

In conclusion, this study introduced LSTM and ARIMA architecture for short-term load forecasting on the PJME hourly consumption dataset. By integrating LSTM networks, the model demonstrated the ability to capture intricate temporal relationships and adapt to exogenous influences such as weather conditions and holidays, ultimately outperforming conventional methods and several state-of-the-art approaches found in the literature. These findings underscore the potential for next-generation deep learning architectures to not only enhance prediction accuracy but also to provide more flexible and interpretable frameworks for understanding load patterns. Moreover, beyond the immediate forecasting gains, this methodology contributes to more informed decision-making in energy markets, supporting improved operational efficiency and resource allocation for grid operators. In the future, incorporating additional socioeconomic factors, exploring transfer learning to other regional datasets, and further optimizing the model's hyperparameters could drive even greater predictive accuracy and robustness. As the energy landscape continues to evolve with the integration of renewable sources and dynamic pricing schemes, adaptive, data-driven forecasting models like the one presented here will become increasingly indispensable.

ETHICAL DECLARATIONS

Referee Evaluation Process

Externally peer-reviewed.

Conflict of Interest Statement

The authors have no conflicts of interest to declare.

Financial Disclosure

The authors declared that this study has received no financial support.

Author Contributions

All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

10.PJME_hourly. (n.d.). Retrieved October 14, 2024, from https://www.kaggle.com/datasets/hajarshokrgozarr/10-pjme-hourly/data

Bhattacharya, P.K., & Burman, P. (2016). Time series. *Theory Method Statist*, 431-489. doi:10.1016/B978-0-12-802440-9.00013-8

Cai, W., Wen, X., Li, C., Shao, J., & Xu, J. (2023). Predicting the energy consumption in buildings using the optimized support vector regression model. *Energy*, 273,127188. doi:10.1016/J.ENERGY.2023.127188

Çetiner, H., & Çetiner, İ. (2021). Analysis of different regression algorithms for the estimate of energy consumption. *Eur J Sci Technol*, 31(1),23-33. doi:10.31590/ejosat.969539

Energy consumption: definition & types | StudySmarter. (n.d.). Retrieved October 14, 2024, from https://www.studysmarter.co.uk/explanations/geography/energy-security/energy-consumption/

Irfan, M., Shaf, A., Ali, T., Zafar, M., Rahman, S., Mursal, S.N.F., AlThobiani, F., Almas, M.A., Attar, H.M., & Abdussamiee, N. (2023). Multi-region electricity demand prediction with ensemble deep neural networks. *PLoS One*, 18(5),e0285456. doi:10.1371/JOURNAL.PONE.0285456

Ismael, M.N. (2022). Time series analysis of household energy using recurrent neural network. *Texas J Engineer Technol*, 8,44-52.

Jambu, M. (1991). 1-D statistical data analysis. *Explorat Multivar Data Analys*, 27-62. doi:10.1016/B978-0-08-092367-3.50007-1

Koşar, E., & Barshan, B. (2023). A new CNN-LSTM architecture for activity recognition employing wearable motion sensor data: enabling diverse feature extraction. *Engineer Appl Artific Intellig*, 124,106529. doi:10.1016/j.engappai.2023.106529

Lewinson, E. (2024). A comprehensive overview of regression evaluation metrics. *Data Sci*, 117-147. doi:10.1007/978-981-97-3245-6_9

Qin, W., Wang, L., Liu, Y., & Xu, C. (2021). Energy consumption estimation of the electric bus based on grey wolf optimization algorithm and support vector machine regression. *Sustainability 2021,* 13(9),4689. doi:10.3390/SU13094689

Son, N.N., & Van Cuong, N. (2023). Neuro-evolutionary for time series forecasting and its application in hourly energy consumption prediction. *Neural Comput Appl*, 35(29),21697-21707. doi:10.1007/S00521-023-08942-X/TABLES/6

Understanding LSTM Networks-Colah's Blog. (n.d.). Retrieved October 14, 2024, from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Wang, X., Liesaputra, V., Liu, Z., Wang, Y., & Huang, Z. (2024). An in-depth survey on deep learning-based motor imagery electroencephalogram (EEG) classification. *Artific Int Med*, 147,102738. doi:10.1016/J.ARTMED.2023.102738

What is an ARIMA model? Taking a quick peek into ARIMA modeling | by Miranda Auhl | Towards Data Science. (n.d.). Retrieved October 14, 2024, from https://towardsdatascience.com/what-is-an-arima-model-9e200f06f9eb

World Energy Consumption Statistics | Enerdata. (n.d.). Retrieved October 14, 2024, from https://yearbook.enerdata.net/total-energy/world-consumption-statistics.html

Xiong, T., Bao, Y., & Hu, Z. (2014). Interval forecasting of electricity demand: a novel bivariate EMD-based support vector regression modeling framework. *Int J Electric Power Energy Syst*, 63,353-362. doi: 10.1016/J.IJEPES.2014.06.010