Time Series Forecasting of Electrical Energy Consumption Using Deep Learning Algorithm

E. O. Edoka, V. K. Abanihi, H. E. Amhenrior*, E. M. J. Evbogbai, L. O. Bello, V. Oisamoje

Department of Electrical and Electronic Engineering, Edo State University Uzairue, Uzairue, Nigeria.

ABSTRACT: Energy consumption forecasting is an operation of predicting the future energy consumption of electrical systems using previous or historical data. The Long Short-term Memory (LSTM) Model; a deep learning model was used in this project to analyze the Short-term consumption forecast performance. This was carried out by using an energy consumption dataset obtained from the Transmission Company of Nigeria (TCN) Benin City regional 132/33KV transmission station. The dataset were daily load readings recorded in the half-hourly format from August to December 2021. The model was used to demonstrate the feasibility of generating an accurate short-term load forecast for the case study despite the peculiarity and insufficiency of the energy consumption readings. Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are the statistical evaluation metrics used. The approach produces exceptional levels of accuracy, with MAPE of 0.010 and RMSE of 19.759 for a 100 time-step. The findings imply that the LSTM model can make accurate predictions with minimal error, and this Deep learning model may be a useful tool for short-term forecasting demand. This finding serves as a baseline for future research in this field of study and power system planning.

KEYWORDS: Energy consumption, Load forecasting, Short-term forecast, Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN).

[Received Oct. 27, 2022; Revised July 2, 2023; Accepted July 26, 2023] Print ISSN: 0189-9546 | Online ISSN: 2437-2110

I. INTRODUCTION

Electricity is vital to people's lives, businesses, and organizations, and it plays a crucial role in the growth of nations. It is a crucial metric for assessing the development-related well-being of a nation (Okakwu *et al*., 2019). Since the success of many economic operations and social welfare are strongly related to electricity usage, today's excessive dependence on this resource is undeniable (da Silva *et al*., 2022). Electrical energy is another name for what is commonly referred to as electricity and the load consumption here is electrical energy.

Nigeria is a developing country with a rapidly expanding population and economy, necessitating a rise in energy consumption. Thus, recognizing future energy consumption trends is crucial for designing, running, and exploring any nation's electrical power system (Ade-Ikuesan *et al*., 2019). As a result, there is need for individuals, academics, nations, and the entire global economy to develop intellectual models for controlling energy consumption processes. Selecting appropriate techniques for evaluating, modelling, and predicting time series of consumption and integrating them with existing information systems for management decisions across individual firms, cities, industries, and states (Kalimoldayev *et al*., 2020).

The invention of Artificial Intelligence has helped in fostering technological advancement across various aspect of human life. Artificial intelligence has provided a means for computer systems to carry out tasks requiring human intelligence. This leads us to Machine Learning where computers are trained and given the ability to learn without being explicitly programmed. Machine learning algorithms uses historical data to predict future information. This is applicable in weather forecasting, stock price prediction, forecasting in power systems, robotics, et cetera. These data are collected and fed into a machine learning model to produce an output. The Models are complex algorithms with brain-like structures called artificial neural networks.

II. THEORETICAL ANALYSIS

A. Forecasting

Estimating future occurrences and situations based on historical data is meant by the term "forecast" (Samuel *et al.*, 2017). Today, the operation and management of electrical energy systems and energy management increasingly depend on energy forecasts. It is the process of estimating the electrical loads required at a given location using historical electrical load demand data (Samuel *et al*., 2017). These forecasts can be just minutes in the future for operational needs or up to five to

*Corresponding author: henrino2003@yahoo.com

doi: http://dx.doi.org/10.4314/njtd.v18i4.1424

ten years in the future for planning (Melodi *et al.*, 2017). Although forecasting and prediction are frequently conflated, a forecast primarily relies on historical data records. A forecast is thought to be more precise and may include a variety of potential outcomes.

B. *Electrical Energy Forecasting Classification*

According to the period for which the forecasts are made, there are four categories of energy forecasting. Table 1 classifies forecasting and its corresponding application.

From Table 1, the lead time for STLF is from half an hour to a few hours. This is the main focus of this work as we would be carrying out our forecast on an hourly basis. Energy forecasting is crucial because underestimating consumption could result in the degeneration and potential collapse of the existing infrastructure due to underinvestment, which could result in load shedding or consumer disconnection. Overestimating it will result in economic waste because huge investments are required to develop excessive power infrastructure (Okakwu *et al.,* 2019).

Table 1. Energy forecasting classification summary

Forecast	Lead Time	Applications
Very short-term (VSTLF)	Few minutes to half an hour	Real-time control. Real-time security evaluation
Short-term (STLF)	Half an hour to a few hours	Allocation of spinning reserve, unit commitment, maintenance schedule
Medium (MTLF)	Few days to a few weeks	Seasonal planning (dry season, rainy season)
Long-term (LTLF)	Few months to a few years	To plan the growth of the generation capacity.

Electrical generation must follow electrical load demand for the best running of the power system (Elgarhy *et al.,* 2017). In order to use their electrical infrastructure effectively, safely, and affordably, the production, transmission, and distribution utilities need a way to forecast the electrical energy. Transmission utilities use electric consumption forecasting methods to improve power flow on the transmission network and lessen congestion and overloads. Since their distribution systems are primarily radial and have predictable maximum load needs, distribution utilities may not be particularly interested in short-term electric load projections.

C. *Machine Learning and Deep Learning*

Deep learning (DL) is a subset of machine learning in which the network is composed of deeper inner hidden layers. By imitating the connectivity grid in the human brain, it aims to make machines like computers think and comprehend like humans do. Machine learning (ML) algorithms are fundamental AI algorithms that identify patterns in unstructured data and use those patterns to make subjective decisions (Almalaq & Edwards, 2017). ML algorithms can be classified into supervised learning and unsupervised learning. A dataset with features is applied to a supervised learning algorithm, and each of the features has a label associated with it. However, in order to derive meaningful attributes from the dataset's structure, an unsupervised learning technique is applied to a dataset with several features (Almalaq & Edwards, 2017). In the learning algorithms, the dataset is always divided into training and testing sub-datasets. The training dataset aids in the model's learning from the raw dataset, whereas the test dataset helps in the validation of the model's output (Almalaq & Edwards, 2017). Deep learning is an approach which learns features and tasks directly from data (Brownlee, 2018). Images, text, and sound all qualify as data, and any of these sorts of data can be fed into deep learning models to solve problems. (Adewuyi *et al.*, 2019) referred Deep learning to a class of machine learning approaches that utilizes multiple layers of information-processing stages in hierarchical designs for feature learning or pattern categorization. Deep learning uses neural networks that can spontaneously learn any complex input-to-output mappings. Additionally, it supports numerous inputs and outputs. It's interesting to note that several of these aspects show tremendous potential for forecasting electrical load or energy, especially for issues with complex and nonlinear dependencies, multivalent inputs, and multi-step forecasting (Brownlee, 2018). The automated feature learning characteristic of convolutional neural networks and the inherently supportive nature of sequential data in recurrent neural networks are two qualities and other neural network capabilities that hold considerable promise (Brownlee, 2018).

D. *Recurrent Neural Networks*

Recurrent neural networks are a type of neural network where the input from one layer of neurons is used as the output from the following layer. Unlike traditional neural networks, memory is connected to each neuron in an RNN and remembers past data (Dhruv & Naskar, 2020). From Figure 1, the input x^t is passed through the RNN cell which produces an output of h^t. Looking closely, the information travels back within the cell. The RNN cell has multiple copies of itself in the same network, and when the cell is unrolled as seen in Figure 1, each cell passes off data to the next, following a sequential structure. Recurrent neural networks are frequently used for situations involving sequential or time-series data (Zhang *et al.*, 2018).

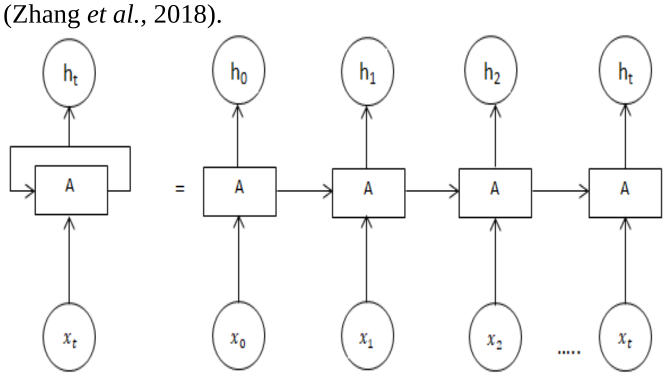

Figure 1. An unrolled Recurrent Neural Network (Virmani & Gite, 2017)

RNNs experience vanishing gradient and expanding gradient issues despite being employed for sequential analysis. The vanishing gradients issue is due to the repeated update of the hidden states in the feedback loop, where the length of the memory persists using the estimations of the previous hidden states and the gradients. After a while, the gradients diminish due to multiplication and differentiation operations (Virmani & Gite, 2017). A kind of recurrent neural network called LSTM is used for sequential prediction and can be used in solving the vanishing gradient issue in RNN amongst other proposed techniques (Islam *et al.*, 2019). However, this study only give emphasis to the LSTM Model.

E. Long Short-term Memory (LSTM)

A special kind of RNN commonly utilized for prediction, incorporating sequential data, is the long short-term memory (LSTM) Model. LSTMs are used to solve issues including character recognition, speech recognition, stock market prediction, forecasting electrical load or energy and other numerous applications. Due to their widespread use in sequential analysis, LSTMs can be trained to anticipate energy consumption in real-time for the upcoming hour or month using the past data.

The LSTM model is a powerful recurrent neural system which can aid in overcoming the exploding/disappearing gradient problems that frequently occur while learning long-term dependencies, even when the minimal time lags are very long (Van Houdt *et al*., 2020).

A cell, an input gate, an output gate, and a forget gate make up a single hidden layer of an LSTM unit, also referred to as a Vanilla LSTM. The LSTM network did not originally include this forget gate, but (Gers et al., 2000) suggested it so that the network could reset its state. The three gates control the flow of information related to the cell, and the cell remembers values across arbitrary time intervals.

The LSTM architecture, or memory blocks, is composed of a number of recurrently connected sub-networks. The memory block's purpose is to keep its state constant throughout time and control information flow using nonlinear gating units (Van Houdt *et al*., 2020). The gates, the input signal parameters, the output parameters, the nonlinear activation functions, and the vector operators are all included in the proposed architecture of the basic LSTM block, which is seen in Figure 2. The output of the block is recurrently connected back to the block input and all of the gates. The gate activation function is always sigmoid, while the input and output activation function are usually tanh. The sigmoid function outputs 0 to 1. It can be used to forget or remember the information. 0 to completely forget and 1, to let the information through. tanh on the other hand, goes from -1 to 1 and it is a great function to add weight to individual values. However, we made use of it in our LSTM model to overcome the vanishing gradient problem.

Let's assume that the network has P blocks and N inputs in order to better understand how the LSTM model functions. The following describes this recurrent neural system's forward pass.

Cell State: From Figure 2, the cell state consists of the path between the cell state $C^{(t-1)}$ and the next cell state C^t. Information flows through the path. The other gates let information through the cell state.

In other words, this step computes the cell value, which combines the block input $z^{(t)}$, the input gate $i^{(t)}$ and the forget gate $f(t)$ values, with the previous cell value. This can be done as depicted below:

$$C^t = (z^{(t)} * i^{(t)}) + (C^{(t-1)} * f(t)) \quad (1)$$

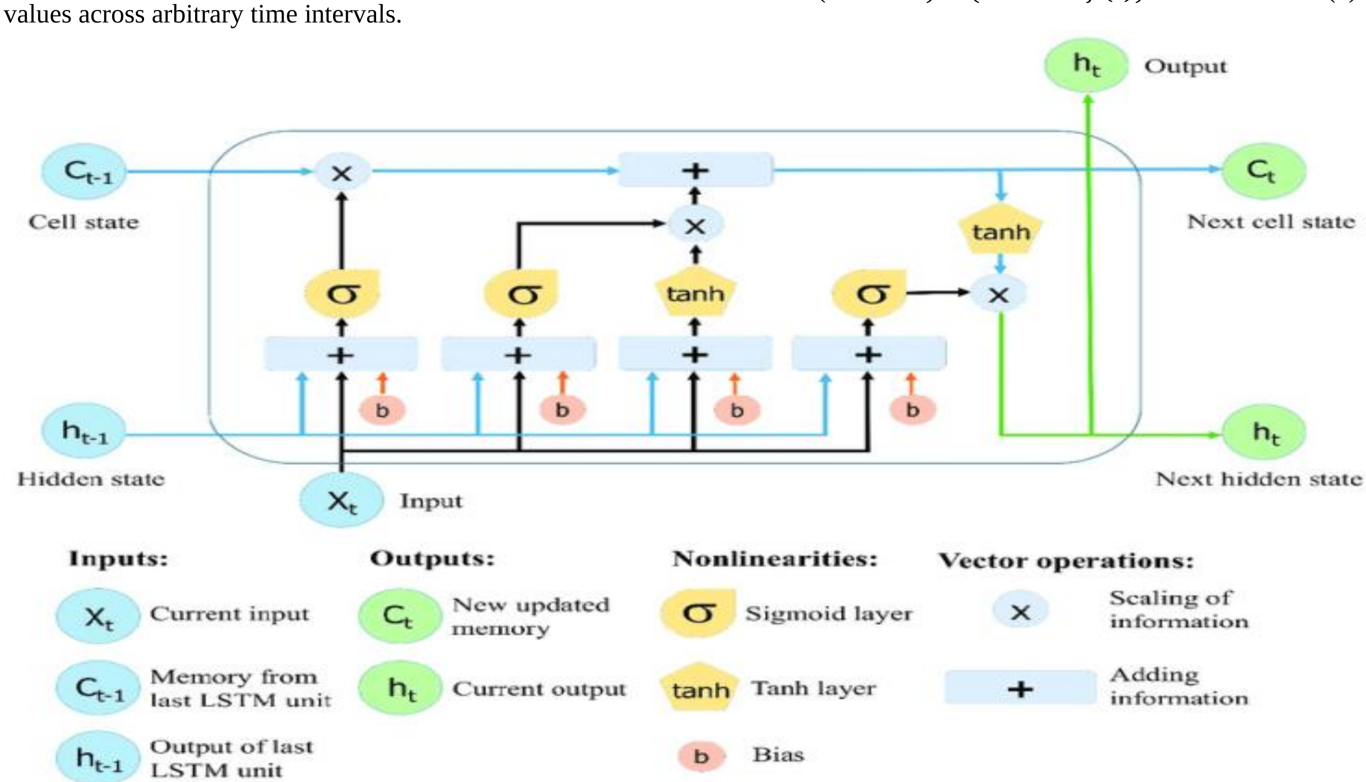

Figure 2 Architecture of the LSTM Model (Le *et al.*, 2019).

Block input: This step is devoted to updating the block input component, which combines the current input $x^{(t)}$ and the output of that LSTM unit $h^{(t-1)}$ in the last iteration. This can be done as depicted below:

$$z^{(t)} = g\left(W_z x^{(t)} + R_z h^{(t-1)} + b_z\right) \quad (2)$$

where W_z and R_z are the weights associated with $x^{(t)}$ and $h^{(t-1)}$, respectively, while b_z stands for the bias weight vector

Input gate: From Figure 2, the input gate comprises of the second sigmoid and tanh function at the middle of the diagram. The input gate is the new information that will be stored in the current cell state. This implies that we update the input gate that combines the current input $x^{(t)}$, the output of that LSTM unit $h^{(t-1)}$ and the cell value $C^{(t-1)}$. in the last iteration. The following Eqns. shows this procedure:

$$i^{(t)} = \sigma\left(W_i x^{(t)} + R_i h^{(t-1)} + b_i\right) \quad (3)$$

$$\breve{C}^{(t)} = tanh\left(W_c x^{(t)} + R_c h^{(t-1)} + b_c\right) \quad (4)$$

where W_i , R_i, W_c $R_c$are the weights associated with $x^{(t)}$ and $h^{(t-1)}$, while b_i and b_c represents the bias vectors associated with this component. The sigmoid layer decides which values are updated, while the $tanh$ layer gives weights to the values to be added to the state.

Forget gate: The forget gate helps the LSTM unit in determining which information should be removed or remembered from its previous cell states $C^{(t-1)}$. Therefore, the activation values $f(t)$, of the forget gates at time step t are calculated based on the current input $x^{(t)}$, the output or history information from the previous step $h^{(t-1)}$ and the state $C^{(t-1)}$ of the memory cells at the previous time step $(t-1)$, and the bias terms b_f of the forget gates. This can be done as follows:

$$f(t) = \sigma(W_f x^{(t)} + R_f h^{(t-1)} + b_f) \quad (5)$$

Output gate: From figure 2, the output gate consists of the sigmoid function, vector operator and tanh function at the far end within the LSTM unit. This part calculates the output gate, which combines the current input $x^{(t)}$, the output of that LSTM unit $h^{(t-1)}$ and the cell value $C^{(t-1)}$ in the last iteration. This can be done as depicted below:

$$o^{(t)} = \sigma\left(W_o x^{(t)} + R_o h^{(t-1)} + b_o\right) \quad (6)$$

$$h^{(t)} = o^{(t)} * tanh(C^t)$$

From the above equation, the sigmoid layer decides which part of cell state is selected for output. The Tanh layer on the other hand gives weight to the values/vector (-1 to 1)

III. REVIEW OF RELATED WORKS

Over the years, numerous researchers have devised various techniques for forecasting utility energy consumption. However, scientific breakthroughs have made it possible to develop more accurate and precise forecasting techniques. Some of these techniques fall within the category of traditional forecasting approaches, and some have inaccurate load estimation problems (Adewuyi *et al.*, 2020). An accurate short-term forecasting strategy is therefore essential.

A review of studies (published between 1991 and 1999) that discussed the use of neural networks for short-term load forecasting is presented in (Hippert *et al.*, 2001). Support Vector Machine (SVM) is another machine learning method that is employed for load forecasting in (Pai *et al.*, 2005) and (Zhu *et al.*, 2007). A hybrid method based on empirical mode decomposition (EMD) and support vector machines (SVMs) was provided from the work of Zhu *et al.* (2007). The EMD and SVM-based hybrid method performed the best overall, with a MAPE of 2.168%, whereas the traditional SVM approach had a higher MAPE of 2.526%. In (Kandil *et al.*, 2006) and (Hayati & Shirvany, 2007), load forecasting was conducted using an artificial neural network (ANN), a typical neural network with one input layer, one hidden layer, and one output layer.

The deep belief network (DBN) approach was presented by Qiu *et al.* (2014) in a study for short load forecasting. In the study, the authors used ensemble methods and emerge Support Vector Regression (SVR) to enhance the performance of DBN with load forecasting. In order to test their suggested methodology, the authors reviewed three datasets of electrical load demand and three regression datasets. According to the findings, SVR, Feedforward NN, DBN, and ensemble NN were all surpassed by the ensemble deep learning that combined DBN and SVR. The proposed technique predicted South Australia's load demand with an RMSE of 30.598; nevertheless, the RMSE for SVR is 44.674 and the RMSE for Feedforward NN was 38.8585.

In a study by Wang *et al.* (2016), the Autoencoder Deep Learning technique was employed for the STLF electricity costing. The authors proposed Stacked Denoising Autoencoders for short-term forecasting. Online forecasting and one-day-ahead forecasting, were the focus of the forecasting methodologies. In particular for day ahead forecasting, the Stacked Denoising Autoencoder model demonstrated successful forecasting results. The outcomes are contrasted with state of the art forecasting techniques like least absolute shrinkage and selection operator (LASSO), multivariate adaptive regression splines (MARS), Support Vector Machine (SVM), and conventional neural network (NN). Gensler *et al.* (2016) looked into the use of Autoencoders and long short-term memory (LSTM) for forecasting renewable energy power plants in their study. The Autoencoder and LSTM were combined in the study for forecasting, and the outcomes were evaluated against cutting-edge techniques including Artificial Neural Network, LSTM, and DBN. With an average RMSE of 0.0713, an average MAE of 0.0366, and an average absolute deviation of 0.2765, the LSTM Deep Neural Network (DNN) achieved the best result.

The Long Short-Term Memory algorithm (LSTM) is one of the popular deep learning techniques used in short-term load forecasting. Marino *et al.* (2016) employed two methods for determining a building's energy load: the standard LSTM and the LSTM-based sequence-to-sequence (S2S) approach. The standard LSTM performed well in this study's one hour resolution but poorly in the one-minute resolution. When compared to results from the literature, both dataset resolutions for the second technique performed admirably. A different study that looked at the use of appliances for individual load forecasting used LSTM for residential load (Kong *et al.*, 2017). According to the study's findings, their algorithm surpasses the state of art approaches for estimating residential loads.

Shi *et al*. (2017) study on household load forecasting included a novel technique called pooling deep recurrent neural network (PDRNN). The performance of forecasting, according to the researchers is improved by the neural network having more layers. In order to maximize the variety and volume of the data, the study had used a pool of inputs for a group of customers. 920 smart metered users in Ireland provided the study's data. The study's findings were contrasted with those of other load forecasting methods, including ARIMA, SVR, and DRNN. The method outperformed ARIMA, SVR, and DRNN in terms of RMSE by 19.5%, 13.1%, and 6.5%, respectively.

Agrawal *et al.* (2018) explained in their study on long-term forecasting of electricity demand based on LSTM deep architecture that the standard methodology for the LSTM is mainly restricted to electricity demand data with the granularity of a month or a year, leading to very low accuracy in their load prediction. As a result, they developed a technique for LTLF with hourly resolution. LSTM variant of the Recurrent Neural Network (RNN) cells served as the model's central component. Kim *et al.* (2018) also suggested a method for predicting short-term electricity consumption using a Long-Short-Term-Memory (LSTM) network that uses a sequence of prior consumption patterns to estimate power consumed one month in the future. Their proposed strategy performs well, with a prediction accuracy of roughly 82.5%. Long Short-Term Memory (LSTM) Model was used by De *et al.* (2018) to build a system for forecasting photovoltaic power. The neural network model was implemented using Python and Keras. The created model was trained with 100 Epochs, and subjected to simulation experiments. Their result showed that the division with two input variables yielded an RMSE value of 2.2013. They also demonstrated that by increasing the number of input variables in the LSTM Model for training doesn't appreciate its overall performance.

Siami-Namini *et al.* (2018) also investigated the potential of recently developed deep learning-based time series forecasting algorithms like "Long Short-Term Memory (LSTM)" over more established ones. The results of the conducted and reported empirical research demonstrated that deep learning-based algorithms like the LSTM outperform traditional-based algorithms like the ARIMA model. More specifically, the average error rate reduction achieved by LSTM was between 84 and 87 per cent lower than ARIMA, demonstrating LSTM's superiority over ARIMA. Additionally, they discovered when developing their model that the number of training sessions, or "epoch" as it is known in deep learning, had no bearing on how well the trained forecast model performed and instead exhibited truly random behaviour.

Two methods were described in the study by Nugaliyadde *et al.* (2019), one utilizing a Recurrent Neural Network (RNN) model and the other using a Long Short Term Memory (LSTM) network on a publicly accessible London smart meter dataset. The proposed models were evaluated to forecast power usage for a single house and a block of houses for a specific period to evaluate the applicability of the RNN and the LSTM network. Short-term, mid-term, and long-term projections were made, considering each day, trimester, and 13 months. The average Root Mean Square Error for the RNN and the LSTM network was 0.1

.Recurrent Neural Network (RNN) and Long Shot Term Memory (LSTM) models were proposed by Yuniarti *et al.* (2019) to anticipate electricity load using various hidden layers made up of three layers. The Random Forest (RF) and Support Vector Machine (SVM) models were then used against the LSTM Model. According to their experimental findings, the LSTM model's Root Mean Square Error (RSME) was the lowest compared to SVM and RF.

As time went on, Adewuyi *et al.* (2020) created specific DL techniques (LSTM, CNN, and MLP) for short-term load forecasting issues using tropical institutional data obtained from a Transmission Company of Nigeria (TCN) and one-year weather data gathered from the Nigerian Meteorological Agency (NiMet) for an institutional customer. They evaluated the predictive accuracy of the various methodologies, and their findings indicated that, when comparing the three, the LSTM model performed on average in terms of the training, validation, and testing metrics. In a different study, two forecasting models were developed to forecast short-term electrical load using long short-term memory neural network (LSTM NN) by Hossain & Mahmood, (2020). In contrast to the other model, which projected multi-step intraday rolling horizons, the first model predicted a single step ahead load.

A more sophisticated approach to STLF is using machine learning techniques that can learn from data (Adewuyi *et al.,* 2020). However, each of these strategies has a unique set of issues, particularly when fitting a model to realize its innate abilities on a job while dealing with a large amount of applicable data. Deep learning approaches or procedures do, however, offer necessary assistance. Some machine learning algorithms and models, such as deep learning techniques, are capable of learning characteristics and tasks directly from application data. Deep learning is found in the machine learning computing knowledge domain and has been the subject of numerous research advancements in several fields of study. However, the methods have recently been used because of their potential for researching power system issues, particularly the forecasting of electrical energy consumption.

Research studies adopting the deep learning method have increased over the past decades (Adewuyi *et al.,* 2020). A vast majority of works in this domain focused on computing applications. However, the last few years also witnessed the application of the deep neural network approach on power system datasets, especially for load forecasting purposes.

In consideration for innovative model expansion, Du *et al*. (2020) proposed an Attention-BiLSTM (Attention-based Bidirectional Long Short-Term Memory, Attention-BiLSTM) network to perform accurate short-term power load forecasting. This model was based on BiLSTM recurrent neural network, which has high robustness in time series data modelling, and the Attention mechanism, which can highlight the key features playing critical roles in load forecasting in input data. The suggested model outperforms existing models in prediction accuracy and algorithm robustness, according to verification studies with actual data in a particular area.

Sheng & Jia, (2020) also developed a novel work which involved a hybrid model of SARIMAX-LSTM in which the

SARIMAX model fits and predicts the data, obtains the relevant residual and prediction results, and then uses the LSTM network to predict the prediction error of the SARIMAX model, and modifies the prediction results of the SARIMAX model. Their model was compared with the SARIMAX, LSTM, and SARIMAX-BP models. Their experiments showed that the model can be well adapted to short-term load forecasting and has the best forecasting effect.

In recent years other researchers have embarked on evaluating deep learning models. Peng *et al.* (2022) long short-term memory-based model was used to forecast energy consumption to improve prediction accuracy while emphasizing key affecting factors. The results from one comparison case and two expanded applications demonstrate that the suggested model outperformed existing common models and basic long short-term memory in terms of prediction accuracy. The suggested model's mean absolute percentage errors for three real-life examples were 4.01 per cent, 5.37 per cent, and 1.60 per cent, respectively. Similarly, the study by Shachee *et al.* (2022) employed LSTM-RNN architecture to forecast household electrical energy consumption for two months from the specified starting date. The model was developed using the necessary features, and its performance was assessed by contrasting real and anticipated values. As a result, the study gave residential homes a forecast of their energy usage, which aided in energy saving at the appropriate moment. The outcome of the proposed predictive model was examined by utilizing data about domestic electric consumption for the consumption forecast. The results show that the revolutionary deep learning approaches produce significantly higher precision than the statistical and engineering prediction models. Compared to conventional models, the proposed LSTM model achieved a very compatible RMSE of 0.6.

It has been observed from previous works that deep learning models pose a more formidable and robust approach to forecasting electrical energy-related works as observed from their results. Deep learning algorithms are now favoured to be more precise and deliver more accurate results than the conventional approaches over the years. Hence, this study has to do with univariant time series analysis for modelling energy forecasting problems using deep architectures based on the proposed model. i.e., Long short-term memory (LSTM).

IV. MATERIALS AND METHODS

A. *Choice of Methodology*

The Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology was considered for this research project as all the six life cycle stages adhere to this research work's objectives. The CRISP-DM is divided into six iterative phases that are loosely associated with one another as illustrated in Figure 3. The phases Business Understanding, Data Understanding, Data Preparation, Data Modeling, Evaluation, and Deployment, which direct the project needs and aims, form the basis of the entire process. The six stages of the life cycle consist of:

i. Business Understanding: Understanding the project's objectives.
ii. Data Understanding: Involves the source of data collection, identifying the data problem, et cetera.
iii. Data Preparation: On the basis of the raw data gathered in the Data Understanding phase, the Data Preparation phase will coordinate the necessary tasks to produce a final dataset. Tasks involving data preparation could be carried out repeatedly. Data preparation is also known as data preprocessing in machine learning analysis. Data Preprocessing helps convert raw, unclean data into a usable format compatible with the machine learning model. Data Preprocessing involves data cleaning, integration, transformation, and reduction. The Data Transformation could be based on Numericalization or Normalization.
iv. Modelling: Optimizing the values of the model by using several modelling techniques.
v. Evaluation: The model is evaluated to see if the project objectives are met.
vi. Deployment: Implementation of the model.

CRISP-DM methodology lets one quickly understand a project as all the stages/phases involved are organized, structured, and well documented.

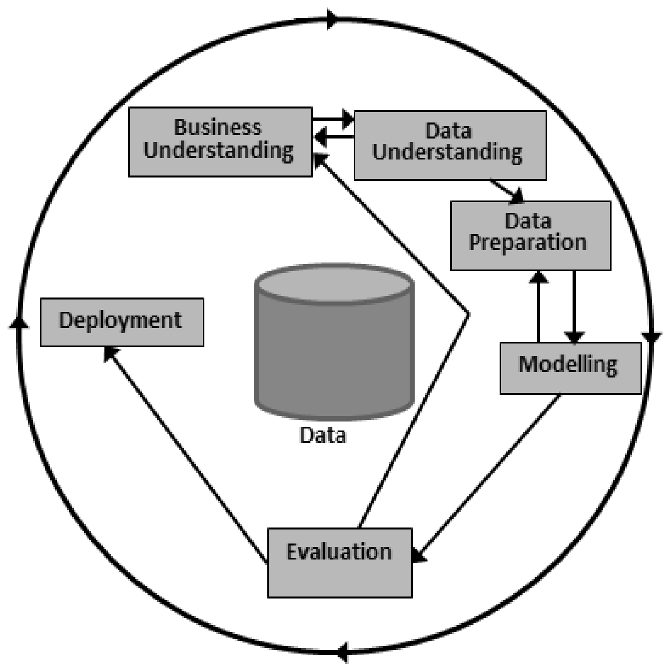

Figure 3. Cross Industry Standard for Data Mining (Berwind *et al.*, 2016)

B. *Model and Architecture of the Proposed System*

The proposed architecture is shown in Figure 2. The architecture structure has been discussed briefly in the previous section.

As mentioned earlier, LSTM can be utilised in a wide variety of situations. However, our focus is on how it can be used for forecasting energy consumption based on our historical data.

In other to run the model on our personal computers, some software libraries have been developed. The Model is implemented in this study using Jupyter Notebook, a software application. This is carried out using software libraries such as python 3.7.7, Pandas, Seaborn, Numpy, Keras, et cetera to meet the research objectives. These libraries are imported into Jupyter Notebook to carry out specific functions involving mathematical computations, data transformation, validation, visualization, modelling et cetera.

''Numpy'' is python's math Library for matrix operations. ''Pandas'' was used to import the dataset from the local file system into the application software for analysis. Different modules and layers from keras, a python library for developing and evaluating deep learning models were used in developing the LSTM model. The Seaborn library was used for data visualization functions.

Table 2 gives a representation of how the model is implemented using Keras. We make use of sequential method which is widely used in most neural network. The sequential model is a plain stack of layers where each layer has exactly one input and output tensor. We add LSTM layers in sequential model via the add () method. As mentioned earlier, the various libraries have been developed already. All we do is to call the LSTM Layer. We also define the units of LSTM in a single layer, i.e. how many units is wanted in the layer of the LSTM. The return sequence enables us in restacking our LTSM units. This model comprises of a stack of four LSTMs having dropouts of 20%, and a dense output layer with unit equals to 1. Figure 9 represents the summary of the model. Once the model is generated, we configure the model with optimizer and losses using regressor.compile(). We train the model using regressor.fit().

A known set of steps that must be performed in order to solve any problem is known as an algorithm. The study's algorithm is shown in Table 3.

Table 2. The LSTM Model (Keras)

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
regressor = Sequential()
# Adding the first LSTM layer and some Dropout regularization
regressor.add(LSTM(units = 50, return_sequences = True,
input_shape = (X_Train.shape[1], 1)))
regressor.add(Dropout(0.2))
# Adding a second LSTM layer and some Dropout regularization
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
 # Adding a third LSTM layer and some Dropout regularization
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
# Adding a fourth LSTM layer and some Dropout regularization
regressor.add(LSTM(units = 50))
 regressor.add(Dropout(0.2))
# Adding the output layer
regressor.add(Dense(units = 1))
 # Compiling the RNN
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
 regressor.summary()
regressor.fit(X_Train, Y_Train, epochs = 100, batch_size = 32)
```

C. Research Data

The secondary data used in training and testing the model proposed in this research are the daily half-hourly load reading sheet obtained from Transmission Company of Nigeria (TCN), Benin regional 132/33kV transmission station control center from August to December 2021. A few months' data could do for short-term electric forecasting. The entire dataset consisted of 3672 data points and two columns representing the date and energy consumption. The maximum and minimum energy consumption were 349 MW and 35.7MW respectively.

D. Flowchart of the model

The flowchart representing the model system is shown in Figure 4.

Table 3. Algorithm of study

INPUT: Dataset **UTPUT: Forecast**

```
1. Acquire the dataset
2. Import all necessary libraries
3. Assess the quality and structure of the
   dataset   (Preprocessing)
4. Clean the dataset
      1. Check if there are Outliers
            Are there outliers?
         If (Outliers == True)
               print ("there are outliers, fix using a
               desired method")
         else
               print ("check for missing values")
      2. Check if there are Missing values
            Are there missing values?
         If (Missing values == True)
               print ("there are missing values, fix using
               the desired method")
         else
               print ("Proceed to datatype conversion")
      3. Check if there are columns that need to
         be converted from one data type to another
            Are there any columns requiring conversion?
         If (Converted Columns == True)
     print ("convert column to the
     desired datatype")
         else
     print ("move to step 5")
5. Transform the Data
   Scale the data into values between 1 and 0 using
   MinMaxScaler6. Check if the data needs some
   analysis (Descriptive, Inferential, et cetera)
7. Divide the dataset into training and testing datasets
  (70% - 30%)
8. Create a forecast model using LSTM with the Processed
   Training Dataset
9. With the developed model, use the testing dataset to test
   the model
10. Visualize the result using Matplotlib and Seaborn
11. Interpretation and validation of the forecast
12. Forecast Visualization.
13. Deploy your project using Streamlit (if necessary)
14. Rebuild the project (If necessary)
15. The End
```

V. RESULTS AND DISCUSSION

A. Analytical results

Table 4 represents the extracted features from the dataset for the first 10 hours of August 1st 2021.

Figure 5 shows the fluctuation in the load consumption for the date from August 1st to December 31st 2021. We can see that the data is a bit cyclic and projects no trend or seasonal pattern. The plot shows that the peak energy consumption was around November, and the minimum energy consumption was around August.

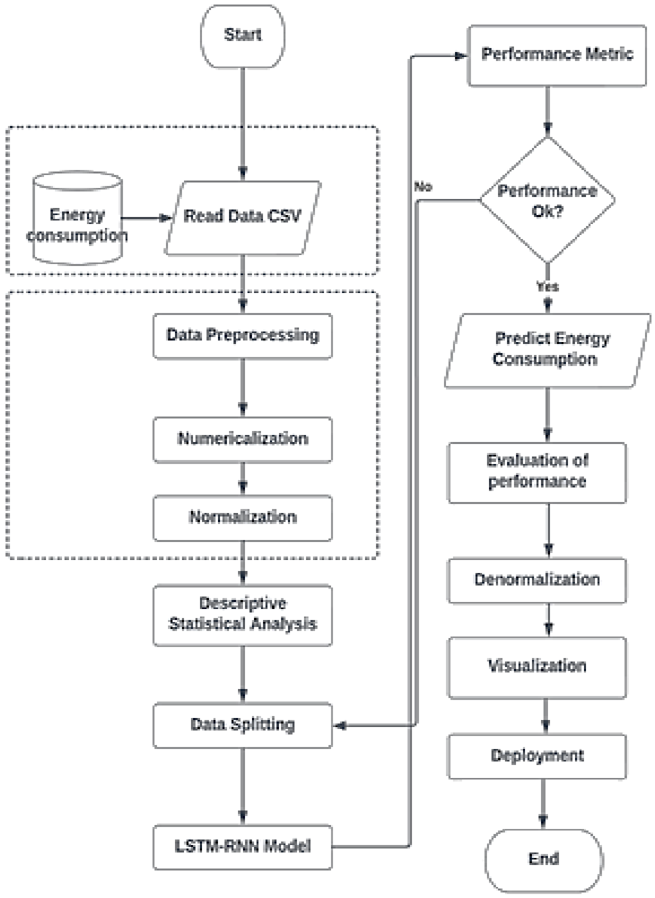

Figure 4. Flowchart representing the model operations

Table 4. Feature extraction

Energy consumption	Month	Date	Time	Day
227.2	8	01/08/2021	01:00:00	Sunday
220.8	8	01/08/2021	02:00:00	Sunday
204.8	8	01/08/2021	03:00:00	Sunday
216.2	8	01/08/2021	04:00:00	Sunday
197.6	8	01/08/2021	05:00:00	Sunday
195.9	8	01/08/2021	06:00:00	Sunday
197.0	8	01/08/2021	07:00:00	Sunday
197.6	8	01/08/2021	08:00:00	Sunday
214.9	8	01/08/2021	09:00:00	Sunday
211.5	8	01/08/2021	10:00:00	Sunday

Figure 8 represents a plot of the energy consumption against time. From the plot, consumption usually increases as time passes. The consumption picks up usually between 7 am and 8 am, and it also shows a significant increase during the evening hours. There is usually a maximum energy usage between 6 pm and 10 pm, from where it gradually decreases.

Table 5 represents the summary of the model for the training discussed in section III. The model consists of multiple hidden LSTM layers, otherwise known as "stacked LSTM".

B. *Comparison of Actual and predicted results*

Figure 9 shows the actual/expected results as compared with the predicted energy consumption for 100 hours at 100 epochs (i.e., Four days). The predicted energy values were compared with the testing dataset which is the expected result, to evaluate the model performance. While Figure 10 is a graphical plot demonstrating the pattern between the actual/expected values and forecasted values for the first 24 time instant, i.e., 24 hours. Table 6 represents the tabulated values of the predicted energy consumption for 24 time-steps.

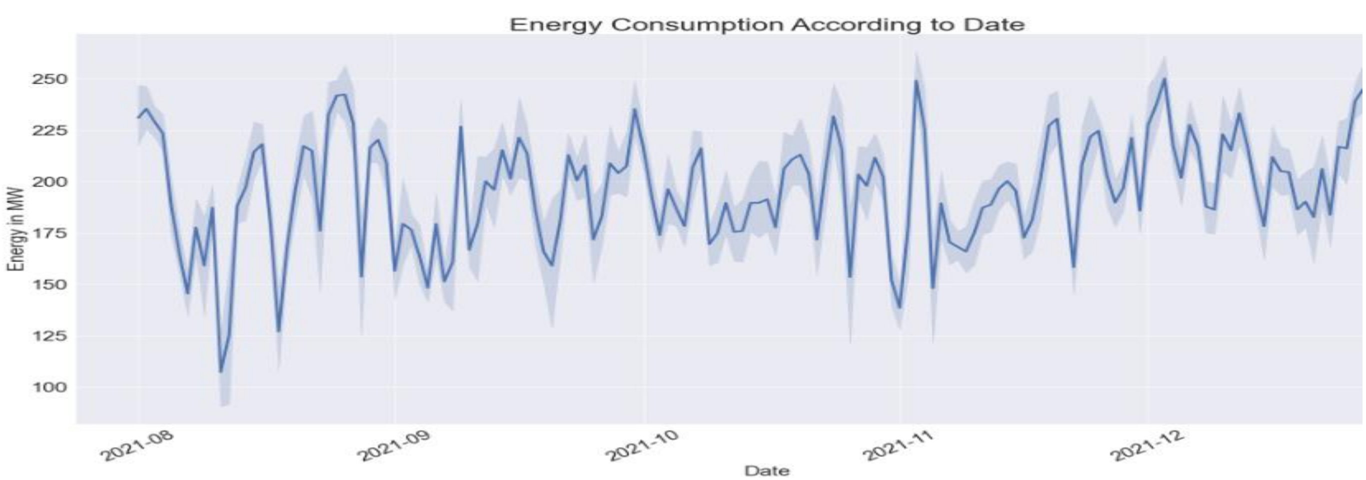

Figure 5. Energy consumption according to date

Figure 6 shows a daily plot of the energy consumed on August 1st for 24 time-step signifying one day. From the plot, it is observed that the energy consumption moderately increased during the early hours of the day, from which it got to the peak point at the 20th hour (i.e., 20:00). It is also seen on the weekly subplot, where consumption decreases as it approaches the weekend.

C. *Model Evaluation*

Considering the number of the training sequence, the MAPE and RMSE are computed as shown in Table 6. Also, Figures 11 and 12 represent the graphical plotting for two cases.

It is observed from Figure 6a and Figure 8 that the energy consumption is usually at its peak during the early hours of the evenings, which is a bit strange considering that it is known that consumption is usually at its peak during the day since

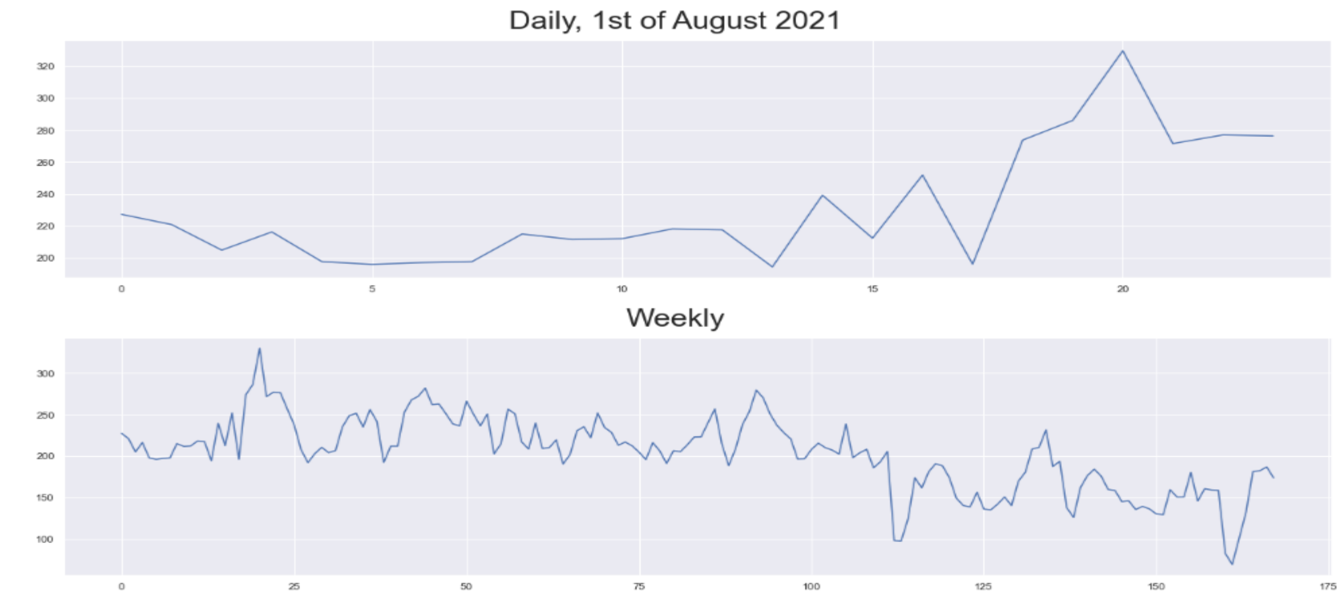

Figure 6. Combined consumption plot in August

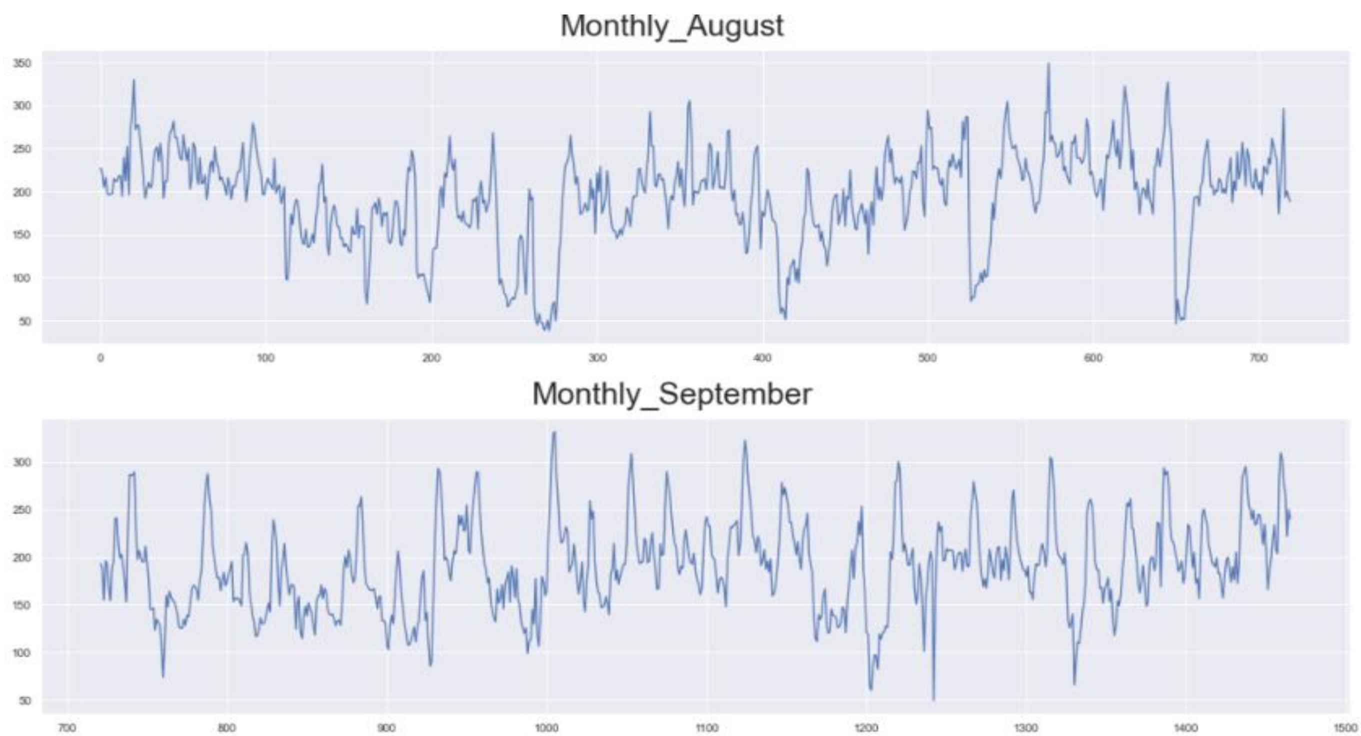

Figure 7. Energy consumption in August and September

people are active and power is not utilized as much and depreciates at night when people are resting. From the report, it is evident that consumption sometimes may not follow the usual manner, and this may result from load shedding as more energy may likely be transmitted to the public during evening hours compared to the day. This unsolved theory needs to be reviewed critically.

Additionally, Table 7 shows the resulting performance of the model when Epoch is set to 50, 100, 200, 300 and 1000. The total training time for all trials is 2 hours 52 minutes. The results accept the idea of Siami-Namini *et al*. (2018) that the number of training sessions, or "epoch" as it is known in deep learning, exhibits truly random behaviour when varied. However, the continuous iterations on the training data at various Epochs gave different results from which the best model is chosen, i.e., Epoch 100. Figure 10 describes a graphical relationship between the actual and predicted load at Epoch 100. As seen in plotting, the deep learning model

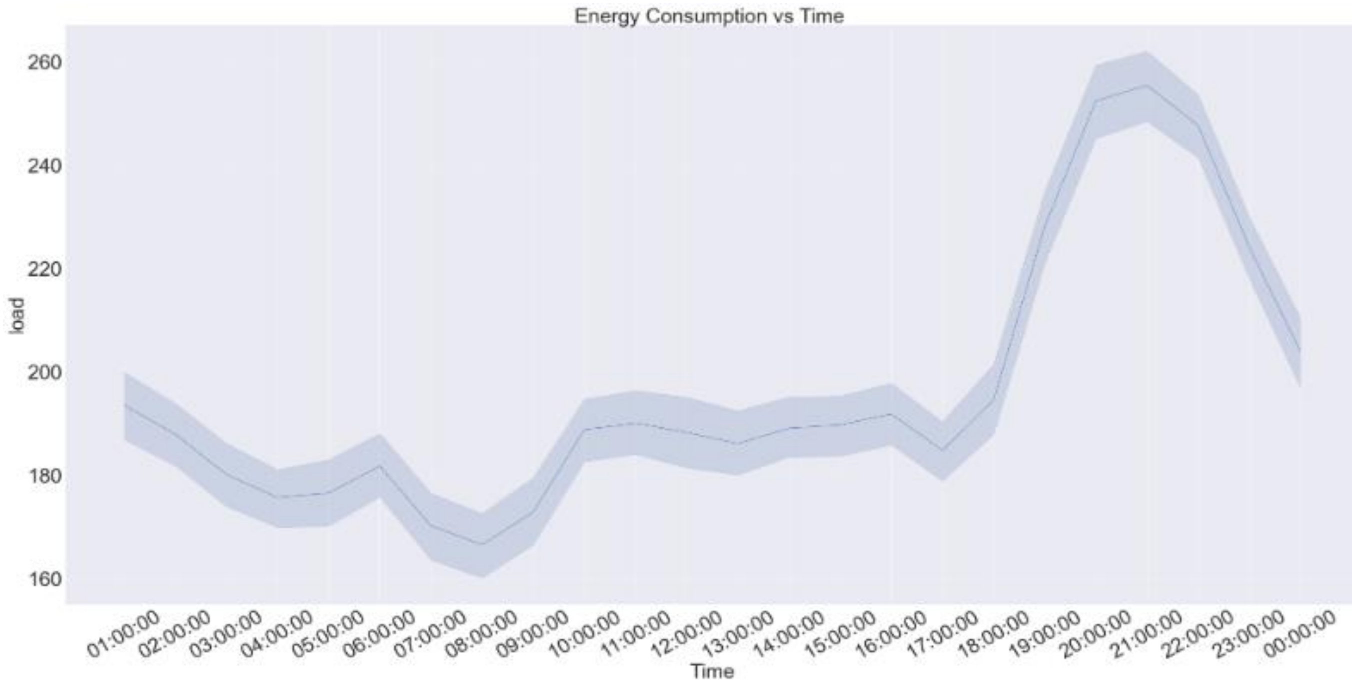

Figure 8. Energy consumption versus time

Table 5. Summary of Keras LSTM Model

Model: "sequential"

Layer (type)	Output Shape	Param#
lstm (LSTM)	(None, 60, 50)	10400
dropout (Dropout)	(None, 60, 50)	0
lstm_1 (LSTM)	(None, 60, 50)	20200
dropout_1 (Dropout)	(None, 60, 50)	0
lstm_2 (LSTM)	(None, 60, 50)	20200
dropout_2 (Dropout)	(None, 60, 50)	0
lstm_3 (LSTM)	(None, 50)	20200
dropout_3 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0

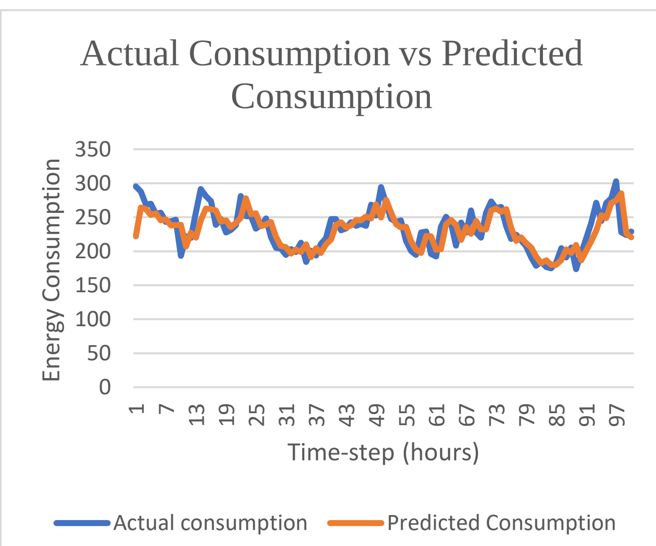

Figure 9. Actual Consumption vs Predicted consumption

Table 6. Actual consumption and Predicted consumption in MW

Time-steps (hours)	Actual consumption (MW)	Predicted Consumption (MW)
1	295.2	221.9940338
2	287.7	264.4190369
3	268.4	261.8044128
4	270.1	253.6004486
5	254.7	255.5489502
6	256.6	245.5325470
7	243.1	246.7816162
8	244.2	237.9253235
9	246.3	238.4672699
10	193.1	238.8977051
11	221.7	206.8859863
12	219.5	227.0987549
13	257.5	219.9811096
14	291.7	245.3375092
15	281.2	262.6774902
16	273.8	262.0673218
17	238.7	260.1496277
18	247.5	243.3737183
19	227.6	245.3516541
20	231.1	235.3358612
21	238.3	241.6934509
22	281.4	249.3227997
23	251.4	277.6965332
24	251.7	254.1683502

Table 7. LSTM performance at different Epoch

LSTM Modelling	MAPE	RMSE
Epoch 50	0.017	40.11
Epoch 100	0.010	19.79
Epoch 200	0.023	52.93
Epoch 300	0.022	52.28
Epoch 1000	0.019	44.90

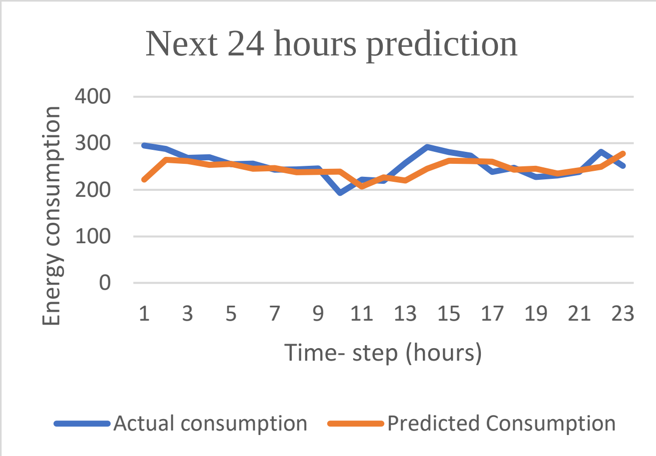

Figure 10. First 24 time-step consumption results

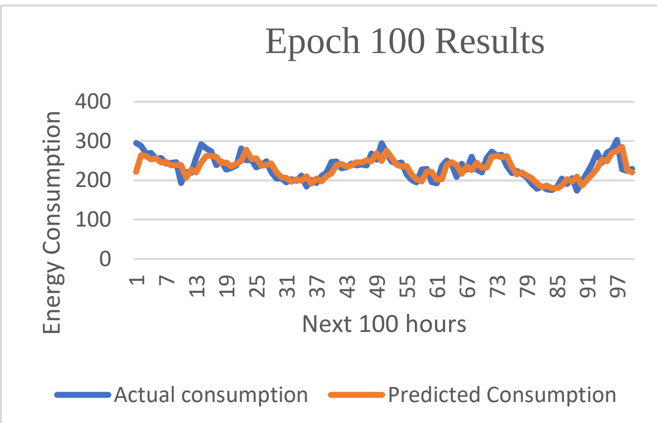

Figure 11. Epoch 100 Result

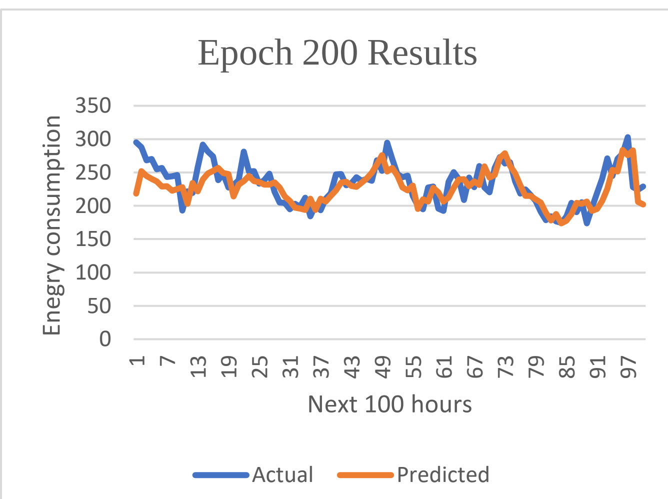

Figure 12. Epoch 200 Result

predicts the future load consumption, and from the plot, it is evident that the LSTM model learned every point, curve, and pattern during the training. The forecasting reliabilities are evaluated by computing the Mean Absolute Percentage Error and Root Mean Square error between the actual and predicted values at various Epochs. As seen in Table 6, Epoch 100 approach resulted in a MAPE of 0.010 and RMSE of 19.759, representing a high degree of accuracy. The results suggest that the LSTM model can perform good prediction with the least error, and finally, this recurrent neural network could be an important tool for short-term load forecasting.

VI. CONCLUSION

In order to create a predictive model to forecast energy consumption based on historical data, this project used the energy consumption data from TCN, Benin City. The STLF energy consumption result demonstrates the LSTM Model's strong performance and excellent prediction accuracy. The MAPE and RMSE between the actual and predicted values were computed to assess the system's forecasting accuracy. After employing multiple layers of LSTM for modelling, a MAPE value of 0.010 per cent and RMSE of 19.79 was attained, demonstrating the high accuracy level. The findings imply that the developed deep learning model makes accurate predictions with minimal error and that this recurrent neural network may be a crucial tool for anticipating short-term load.

AUTHOR CONTRIBUTIONS

E. O. Edoka: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **V. K. Abanihi:** Conceptualization, Supervision, **H. E. Amhenrior:** Data curation; Funding acquisition; Writing - review & editing; Methodology; Project administration. **E. M. J. Evbogbai:** Methodology and supervision. **L. O. Bello:** Supervision; Validation; Visualization. **V. Oisamoje:** Supervision; Validation; Visualization; Investigation; Project administration.

ACKNOWLEDGEMENT(S)

The authors wish to acknowledge and thank the Transmission Company of Nigeria, Sapele Road, Benin City, Edo State for the release of the data used for this work.

REFERENCES

Ade-Ikuesan, O. O.; Oyedeji, A. O. and Osifeko, M. O. (2019). Linear regression long-term energy demand forecast modelling in Ogun State, Nigeria. Journal of Applied Sciences and Environmental Management, 23(4), 753–757.

Adewuyi, S. A.; Aina, S., and Oluwaranti, A. I. (2020). A deep learning model for electricity demand forecasting based on tropical data. Applied Computer Science, 16(1).

Agrawal, R. K.; Muchahary, F. and Tripathi, M. M. (2018). Long-term load forecasting with hourly predictions based on long-short-term-memory networks. 2018 IEEE Texas Power and Energy Conference (TPEC), 1–6.

Almalaq, A. and Edwards, G. (2017, December). A review of deep learning methods applied on load forecasting. In 2017 16th IEEE international conference on machine learning and applications (ICMLA) (pp. 511-516). IEEE.

Berwind, K.; M. Bornschlegl; M. Hemmje and M. Kaufmann. (2016). Towards a cross industry standard process to support big data applications in virtual research environments. CERC2016, 82.

da Silva, D. G.; M. T. B. Geller; M. S. dos Santos Moura; and A. A. de Moura Meneses. (2022). Performance evaluation of LSTM neural networks for consumption prediction. E-Prime-Advances in Electrical Engineering, Electronics and Energy, 2, 100030.

De, V.; T. T. Teo; W. L. Woo and T. Logenthiran (2018). Photovoltaic power forecasting using LSTM on a limited dataset. 2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), 710–715.

Dhruv, P. and Naskar, S. (2020). Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. Machine Learning and Information Processing: Proceedings of ICMLIP 2019, 367-381.

Du, J.; Y. Cheng; Q. Zhou; J. Zhang; X. Zhang and G. Li. (2020). Power load forecasting using BiLSTM-attention. IOP Conference Series: Earth and Environmental Science, 440(3), 032115.

Elgarhy, S. M.; M. M. Othman; A. Taha and H. M. Hasanien. (2017). Short-term load forecasting using ANN technique. 2017 Nineteenth International Middle East Power Systems Conference (MEPCON), 1385–1394.

Gensler, A.; J. Henze; B. Sick and N. Raabe. (2016). Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In 2016 IEEE international conference on systems, man, and cybernetics (SMC) (pp. 002858-002865). IEEE.

Gers, F. A.; Schmidhuber, J. and Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

Hossain, M. S. and Mahmood, H. (2020). Short-term load forecasting using an LSTM neural network. 2020 IEEE Power and Energy Conference at Illinois (PECI), 1–6.

Hayati, M. and Shirvany, Y. (2007). Artificial neural network approach for short term load forecasting for Illam region. World Academy of Science, Engineering and Technology, 28, 280-284.

Hippert, H. S.; Pedreira, C. E. and Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems*, *16*(1), 44-55.

Islam, M. S.; S. S. S. Mousumi; S. Abujar and S. A. Hossain. (2019). Sequence-to-sequence Bangla sentence generation with LSTM recurrent neural networks. Procedia Computer Science, 152, 51-58.

Kalimoldayev, M.; Drozdenko, A.; Koplyk, I.; Marinich, T.; A. Abdildayeva and T. Zhukabayeva. (2020). Analysis of modern approaches for the prediction of electric energy consumption. Open Engineering, 10(1), 350–361.

Kandil, N.; R. Wamkeue; M. Saad and S. Georges. (2006). An efficient approach for short term load forecasting using artificial neural networks. International Journal of Electrical Power & Energy Systems, 28(8), 525-530.

Kim, N.; M. Kim and J. K. Choi. (2018). LSTM-based short-term electricity consumption forecast with daily load profile sequences. 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 136–137.

Kong, W.; Z. Y. Dong; D. J. Hill; F. Luo and Y. Xu. (2017). Short-term residential load forecasting based on resident behaviour learning. IEEE Transactions on power systems, 33(1), 1087-1088.

Le, X.-H.; H. V. Ho; G. Lee and S. Jung. (2019). Application of long-short-term(Kim et al., 2018)m memory (LSTM) neural network for flood forecasting. Water, 11(7), 1387.

Marino, D. L.; K. Amarasinghe and M. Manic (2016). Building energy load forecasting using deep neural networks. In IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society (pp. 7046-7051). IEEE.

Melodi, A. O.; S. T. Adeniyi and R. H. Oluwaniyi. (2017). Long-term load forecasting for Nigeria's electric power grid using ann and fuzzy logic models. 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON), 962–968.

Nugaliyadde, A.; U. Somaratne and K. W. Wong. (2019). Predicting electricity consumption using deep recurrent neural networks. ArXiv Preprint ArXiv:1909.08182.

Okakwu, I. K.; E. S. Oluwasogo; A. E. Ibhaze and A. L. Imoize. (2019). A comparative study of time series analysis for forecasting energy demand in Nigeria. Nigerian Journal of Technology, 38(2), 465–469.

Pai, P. F. and Hong, W. C. (2005). Support vector machines with simulated annealing algorithms in electricity load forecasting. Energy Conversion and Management, 46(17), 2669-2688.

Peng, L.; Wang, L.; D. Xia and Q. Gao. (2022). Effective energy consumption forecasting using empirical wavelet transform and long short-term memory. Energy, 238, 121756.

Qiu, X.; L. Zhang; Y. Ren; P. N. Suganthan and G. Amaratunga (2014). Ensemble deep learning for regression and time series forecasting. In 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL) (pp. 1-6). IEEE.

Samuel, I. A.; A. Emmanuel; I. A. Odigwe and F. C. Felly-Njoku. (2017). A comparative study of regression analysis and artificial neural network methods for medium-term load forecasting. Indian Journal of Science and Technology, 10(10), 1–7.

Shachee, S. B.; H. N. Latha and N. Hegde Veena. (2022). Electrical Energy Consumption Prediction Using LSTM-RNN. In Evolutionary Computing and Mobile Sustainable Networks (pp. 365–384). Springer.

Sheng, F. and Jia, L. (2020). Short-term load forecasting based on SARIMAX-LSTM. 2020 5th International Conference on Power and Renewable Energy (ICPRE), 90–94.

Shi, H.; M. Xu and R. Li (2017). Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Transactions on Smart Grid, 9(5), 5271-5280.

Siami-Namini, S.; N. Tavakoli and A. S. Namin. (2018). A comparison of ARIMA and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1394–1401.

Van Houdt, G.; C. Mosquera and G. Nápoles. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53, 5929-5955.

Virmani, S. and Gite, S. (2017). Performance of convolutional neural network and recurrent neural network for anticipation of driver's conduct. In 2017 8th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-8). IEEE.

Wang, L.; Z. Zhang and J. Chen. (2016). Short-term electricity price forecasting with stacked denoising autoencoders. IEEE Transactions on Power Systems, 32(4), 2673-2681.

Yuniarti, E.; N. Nurmaini; B. Y. Suprapto and M. N. Rachmatullah. (2019). Short Term Electrical Energy Consumption Forecasting using RNN-LSTM. 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), 287–292.

Zhang, B.; J. L. Wu and P. C. Chang. (2018). A multiple time series-based recurrent neural network for short-term load forecasting. Soft Computing, 22, 4099-4112.

Zhu, Z.; Y. Sun and H. Li. (2007). Hybrid of EMD and SVMs for short-term load forecasting. In 2007 IEEE International Conference on Control and Automation (pp. 1044-1047). IEEE.