Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Power Forecasting

Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Power Forecasting

Maha S. Alsabban
Electrical and Computer Engineering
Effat University
Jeddah, KSA
maalsabban@effat.edu.sa

Nema Salem
Electrical and Computer Engineering
Effat University
Jeddah, KSA
nsalem@effatuniversity.edu.sa

Hebatullah M. Malik
Electrical and Computer Engineering
Effat University
Jeddah, KSA
hemalik@effat.edu.sa

2021 13th IEEE PES Asia Pacific Power & Energy Engineering Conference (APPEEC) | 978-1-6654-4878-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/APPEEC50844.2021.9687681

***Abstract*—The geographical position of the Kingdom of Saudi Arabia has significant potentials for utilizing renewable energy resources, which aligns with the country's vision for 2030. This paper proposes a solution to achieve energy sustainability by forecasting future load demands through adopting three different scenarios. We used the outsourced Individual Household Electric Power Consumption Dataset, University of California- Irvine repository, for testing our proposed system. We utilized the Long Short-term Memory-Recurrent Neural Network (LSTM-RNN) algorithm to estimate the whole house power consumption for different horizons: every 15 minutes, daily, weekly, and monthly. Next, we evaluated the performance of the system by Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and R^2 score metrics. Then, we applied the Mean Absolute Percentage Error (MAPE) to find its accuracy. The results showed that the monthly forecasting interpretation scenario was the best performing model. That scenario used (n-1) months for training and the last month for testing. The scores for that model were 0.034 (MAE), 0.001 (MSE), 0.034 (RMSE), and 97.16% (accuracy). The constructed model successfully achieved its goals of predicting the active power of the household and now can be accommodated on energy applications not only in Saudi Arabia but also in any other country.**

***Index Terms*—LSTM-RNN, load forecasting, power, deep learning, artificial intelligence**

I. Introduction

The Kingdom of Saudi Arabia is building its energy vision following global concerns. The Saudi 2030 vision aims at lowering its fossil fuel generation by substituting it with clean energy sources [1]. However, the goal seems far out of reach as the energy demand in the country is expected to increase at a higher rate than the projected renewable availability. This challenge allows researchers to suggest alternative paths to meet the sustainability objectives the country had set. One of the solutions is load forecasting.

Forecasting is the process of making estimations/predictions of future based on previous historical data [2]. There are several advantages and benefits of making load predictions. It allows better planning for future consumption by the utility company, ensures continuous supply of power because of pre-determination of needed resources, gives a good reference for utilities' companies to maximize the use of generation methods [3]. This paper proposes an LSTM-RNN based forecasting model and illustrates the obtained results.

II. Literature Review

Customer Side Demand Response (CSDR) has drawn lots of attention in the last years from the industry and academic fields perspectives [4]. Therefore, there is a need for accurate load forecasting for (CSDR). The following paragraph discusses several existing studies covering methodologies, performance metrics, and datasets' types. In [5], the authors proposed an LSTM-RNN model simulated in MATLAB to forecast the future load using a normalized set of data. The LSTM works in figuring out the prediction of the next-time step as electric loads relies on the factor of time. The model results achieved RMSE less than 1. In [6], a Deep-Learning method (DL), called Temporal Convolutional Networks (TCN), was used to accomplish a Short-term Residential Load Forecast (STRLF). The model does not only keep long load memory, but also it has the ability of parallel processing of load information. The researchers acquired data from the Almanac of Minutely Power dataset-version 2 (AMPds2) smart meter [7]. The results showed that (TCN) forecasting system achieves a good performance of tracking the fluctuation residential loads with MAPE of 6.66% and MAE of 0.36 in TCN(B1C1), and RMSE of 1.24 in TCN(B2C2) when compared to all available benchmarks' forecasting models.The authors of [4] presented an STRLF framework based on Iterative ResBlocks in a Deep Neural Network (IRBDNN) method conducted on Reference Energy Disaggregation Dataset (REDD) [8]. The IRBDNN method outperformed already existing methods (e.g., Auto-Regressive Moving Average (ARMA), Elaboration Likelihood Model (ELM), etc.) with reduced measurements metrics of RMSE, MAE, and MAPE by (3.89% - 20%), (2.18% - 22.58%), and (0.69% - 32.78%), respectively. Moreover, the more iterations performed, the better the performance achieved. In [9], the authors forecasted five minutes onward residential active power consumption of an aggregated set of data utilizing an Appliance Usage Patterns (AUP) based method. The researchers utilized the REDD dataset [8]. The prediction of the total power demands was associated with confidence levels (α) of 50%, 70%, 90%, and 95%. They measured the accuracy of the power-demand in terms of each confidence level (α), for each consecutive three hours a day. The achieved results were in the range (70.4% - 81.2%). The

978-1-6654-4878-9/21/$31.00 ©2021 IEEE

authors of [10] proposed a hybrid model based on Convolutional Neural Network (CNN) and time-series image encoding techniques to predict power usage at each single household level. The study utilized the Boston housing dataset [11] and active power consumption of single residential household's historical data [12] to measure the performance and compare it with existing techniques (e.g., Support Vector Machine (SVM) [13], Artificial Neural Network (ANN) [14], CNN [15]). The selected image encoding technique was Recurrence Plots (RP) (among Gramian Angular Summation Field (GASF), Gramian Angular Difference Field (GADF), Modulation Transfer Function (MTF)), that achieved the best MAE, RMSE, and MAPE of 1.12, 3.92, and 14.15, respectively. The proposed model showed good performance in comparison with the previously mentioned models with MAE of 0.59%, RMSE of 0.79%, and MAPE of 12.54%. In [16], the authors presented a Support Vector Regression (SVR) based electricity consumption for anonymous individual household's prediction model on an hourly and daily basis. The study used a dataset from the Canadian utility company [17]. It visualized and featured it by using the Exploratory Data Analysis (EDA). The study states that despite the feasibility of residential energy consumption for individual household forecasting, the model performance relies on the variability behavior of that household (stochastic nature). Therefore, daily forecasting performed better with MAPE of 12.78 in comparison with 23.31 on an hourly basis. The authors of [18] proposed an individual load-forecasting method based on integrating the Empirical Mode Decomposition (EMD) and the neural network regression techniques. The EMD divides the load profile (sequential data/time series data) into multiple IMFs and residuals while obtaining the voluntary patterns. Thus, these elements are used along with the LSTM to achieve proper forecasting. After that, the study got the final value of the forecasting by applying the restoration process. Several models had been developed and tested with and without the EMD approach. The results showed better performance of the models (SVR, Gradient Boosting Regression (GBR), ANN, Deep Neural Network (DNN), and Stacked Long Short-Term Memory (SLSTM)) based on EMD than the ones without it by the measurement metrics of MAPE and RMSE. However, the training time is different in each of the previously mentioned models. The MAPE and RMSE of EMD-SVR are 31.1 and 20.6, EMD-GBR are 31.29 and 18.15, EMD-ANN are 29.71 and 15.99, EMD-DNN are 26.91 and 16.15, and EMD-SLSTM are 25.73 and 14.54, respectively. Whereas the MAPE and RMSE without EMD are 64.58 and 30.63 (SVR), 45.57 and 29.15 (GBR), 46.91 and 29.73 (ANN), 49.39 and 29.44 (DNN), 43.64 and 29.65 (SLSTM). If the load profile is highly nonstationary, the EMD-SLSTM model will ease the process of forecasting. Moreover, the EMD-SLSTM method has the best performance in comparison with all other EMD based methods. In [19], the authors developed an Adaptive Neuro Fuzzy Inference System (ANFIS) based model for residential load forecasting. The model considered the temperature and an extra variable presented based on occupancy and special/weekdays for the power consumption forecasting. The study acquired dataset from the light, gas, and water division of Memphis for a single apartment. The performance results from comparison to an ANN-based model in which ANFIS based model outperformed. The RMSE (kWh) and MAPE of the ANFIS model are 4.14 and 9.20%, respectively, while 5.96 and 86.89% of the ANN-based model. The authors of [20] proposed a novel Residential Electrical Load Anomaly Detection (RELAD) model that includes a One-Step-Ahead Load-Predictor (OSA-LP) and a Rule-Engine based Anomaly Detector (RE-AD). The input to the system is real-time load data besides the historical load data of the previous 24 hours. The output is composed of the prediction outcome of the next step and the real-time data detection result. The OSA-LP minimizes the over/underfitting issue related to the real-time forecast and the RE-AD can easily specify any anomalies such as leakage, steal of electricity. The study compared this model to other two methods with the same dataset (LSTM-AD and Auto-Regressive Integrated Moving Average (ARIMA)-AD). In regards to the prediction process, it outperformed both of them with R^2 of 0.6967 and Root Mean Square Deviation (RMSD) of 217.3001. In addition, the study compared the model to the other three methods (SVM-AD, ARIMA-AD, and LSTM-AD). In regards to anomaly detection, RELAD showed the best anomaly detection performance with an F_1 of 0.9953.

III. Introduction of the Utilized Dataset

A multivariate dataset called individual household electric power consumption was provided by the University of California, Irvine (UCI). The university acquired the dataset from a household located in Sceaux, France, for 47 months starting from December 2006 until November 2010 with a sampling rate of one minute. The dataset has nine features shown in Fig. 1, however, these features are altered for the purpose of the study.

Five pre-processing techniques are applied on UCI to achieve better results in the following order. First, replacement of null values to avoid inaccurate results from an incomplete dataset. Second, appending an additional feature for enhancing the forecasting results and billing purposes. Third, data reduction to optimize computation time. Fourth, data normalization to reduce prediction and analysis complexities. Finally, conversion of data from time-series to supervised-learning to facilitate training of machine-learning algorithms. After this step, the maintained dataset features are Global_active_power, Voltage, Sub_metering_1, Sub_metering_2, Sub_metering_3, and Sub_metering_4 with a final shape of $(2,075,259 \times 6)$.

A. Exploratory Data Analysis

After obtaining the modified dataset, we observed the general trends of each feature for the entire capture period.

Firstly, the Global_reactive_power readings are very low when compared to the Global_active_power. This might be accounted for two facts. (1) the appliances' nature. (2) the residential sector consumption pattern. Secondly, voltage is erratic in the range from 230 Volts to 250 Volts due to the fact that voltage from the mains can have values 230+10%

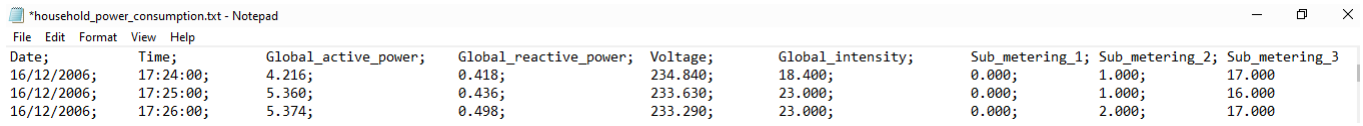

Fig. 1. UCI data features

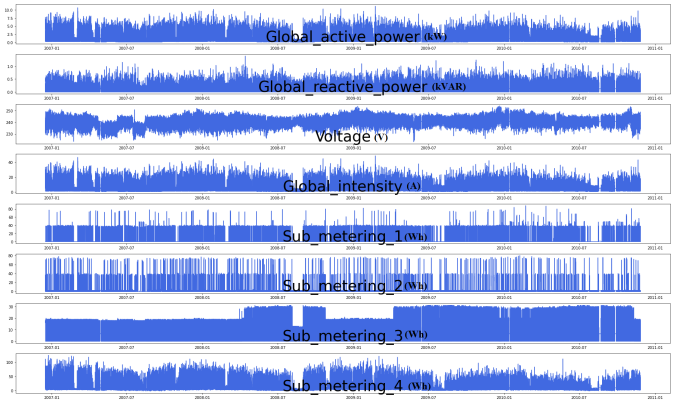

Fig. 2. The dataset features per minute resolution

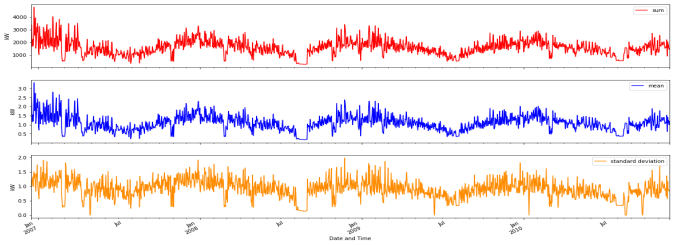

Fig. 3. Global_active_power resampled over day

and 230-6% [21]. Moreover, maximum values of the four individual Sub-meterings 1, 2, 3, and 4 are 88 Wh, 80 Wh, 31 Wh, and 124.8 Wh, respectively. In addition, Sub_metering_3 indicates the possibility of control availability. An interesting observe is that Sub_metering_4 tends to decrease as Sub_metering_3 increases. All of this is a reflection of the consumption regularity of corresponding appliances connected to each Sub-meter.

As for the statistical properties of the power-related features, Figures 3, and 4 show (from top to bottom) the summation, mean, and standard deviation of Global_active_power, and Global_reactive_power consumed per day, respectively. The three statistical properties follow the same trend in both features. Firstly, Global_active_power ranges between 250.30 kW and 4773.39 kW and the average consumption of means across all days is 1.093 kW. Secondly, Global_reactive_power has a range from 34.92 kVAR to 417.83 kVAR and the average consumption of means across all days is 0.124 kVAR. The standard deviation shows how spread the data-points are from the mean.

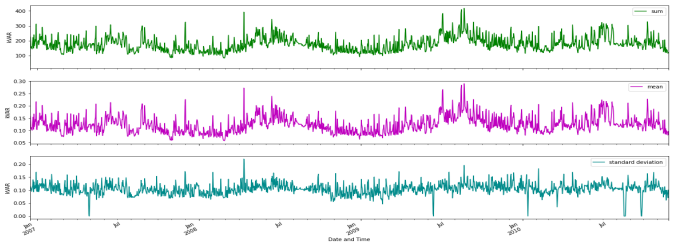

Fig. 4. Global_reactive_power resampled over day

IV. Proposed Methodology

The proposed system is programmed, tested, and ran in Spyder software using Python programming language. the UCI dataset is fed to the forecasting algorithm (LSTM-RNN) to forecast the power consumption of a house for different periods. Fig. 5 illustrates the system's block diagram.

In our implementation, we used the LSTM algorithm to forecast the Global_active_power at a time-step given the previous measurements of all other features. We utilized a single LSTM cell in the hidden layer in which the output is a multiplication of two activation functions (Sigmoid (σ) and $tanh$ functions), as in Fig. 6. The cell's components can be explained through equations 1 to 6, where **W** is a given weight matrix. The equations are formed using the help of three gates, namely the forget, input, and output gate [22].

In our application, forget gate and the output gate are resembled in the logistic sigmoid function (f_t) and (o_t), respectively. The input gate is made up of two layers: a logistic sigmoid (σ) and a $tanh$ layer. Finally, the symbol ($\oplus$) represents a concatenation process, (*) represents an element-wise multiplication, and ($\cdot$) represents a dot product [23]. This LSTM-RNN architecture performs better in time-series sequential problems and learning long-term dependencies in a large dataset. Table I shows the model architecture parameters.

$$f_t = \sigma\left(W_f \cdot [h_{t-1} \oplus x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1} \oplus x_t] + b_i\right) \quad (2)$$

$$b_t = tanh\left(W_C \cdot [h_{t-1} \oplus x_t] + b_C\right) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * b_t \quad (4)$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1} \oplus x_t] + b_o\right) \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

Where b_f, b_i, b_C, and b_o are biases that are initialized to zeros.

The LSTM model is in a three-dimensional form consisting of the number of samples (train/test), the number of time-steps,

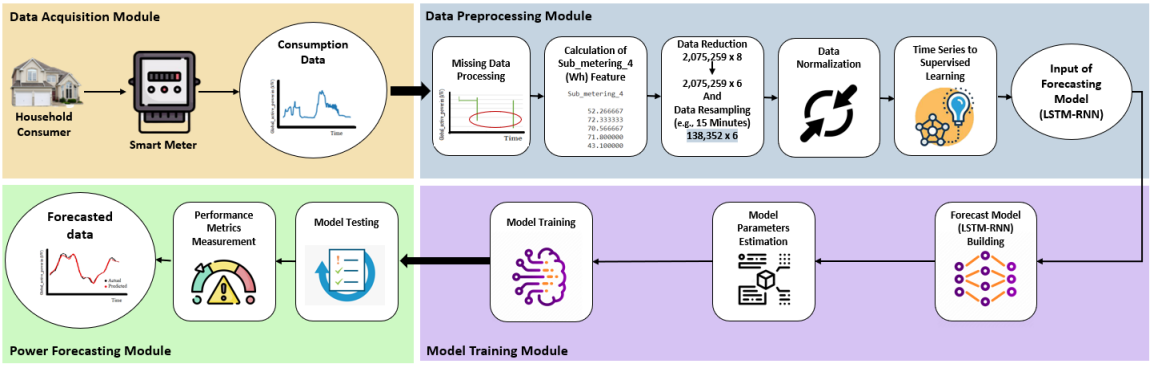

Fig. 5. Block diagram of the LSTM-RNN power forecasting system

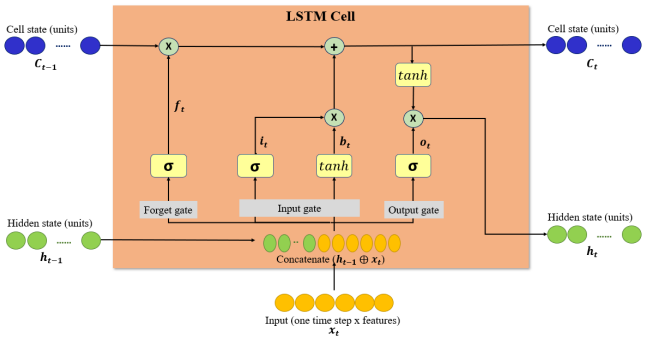

Fig. 6. The structure of a LSTM cell

and the number of features. However, the LSTM input layer is a two-dimensional shape comprising several time-steps and many features. It assumes one or more samples (a sample is one row of data) [25]. The hidden layer contains one hidden state that is composed of the number of LSTM units. Since this hidden state is not visible outside this layer and only fed back to the LSTM cell, it can be assigned several weighing values that give better results and performance based on trial and error. In our study, the units of the hidden state differ in each scenario which can capture the structure of the features of an input sequence and helps to enhance the model performance [26]. As the number of units increases, the model will be perfectly trained and would have memorized the training set. However, the computation time will become longer, and the problem of over-fitting arises [27]. In addition, the LSTM unit number should not be too small to avoid under-fitting (the inability to model the training data nor generalizing the model to new data [28]). The dropout layer is responsible for the process of randomly setting the input units to zero with a specific rate (optimal value of 20% [24]) at every single time step in the training phase of the model [29]. The output layer is a Dense layer that takes all intermediate vectors ($\mathbf{h_t}$) of the LSTM hidden layer as an input to the neuron to comply with the extracted data and reach a final output, as given in Equation 7 (the initial bias is a zero) [30]. Noting that the Global_active_power is the output of our proposed system.

$$Output = LinearActivationFn((h_t \cdot Weight) + bias) \quad (7)$$

where:

$$LinearActivationFunction : a(z) = z \quad (8)$$

The loss function measures the accuracy of the LSTM algorithm in modeling the dataset [31]. The adapted loss function while compiling the model is MSE (Equation 9) since it is a regression problem with one neuron at the output layer that has a linear activation function [32].

$$MSE(h) = \frac{1}{N}\sum_{1}^{N}(\hat{y}_t(t+h) - y_t(t+h))^2 \quad (9)$$

Where $(t+h)$ is the lead time instant in the forecast horizon, and N is the number of forecasting time-steps.

The epoch is a hyper-parameter (that is set to determine the network structure before training and controls the model behavior throughout the training process [33]) that defines the number of iterations (complete passes) of the LSTM algorithm on the training dataset [34]. After each epoch, the LSTM algorithm updates the model-internal parameters to reduce the error as much as possible. We had to choose an appropriate number of Epochs to train the model appropriately, avoiding the training over-fitting. An epoch is composed of some batches. The batch size is a hyper-parameter that determines the number of time steps. The model processes these time

TABLE I
LSTM-RNN MODEL ARCHITECTURE PARAMETERS

Parameter	Description
Input Layer Shape	One time step with six features
Hidden Layer Shape	A LSTM cell composed of units depending on the scenario in our project
Dropout Layer	Rate of 20% [24]
Output Layer Shape	One neuron/unit for Global_active_power forecasting
Optimizer	AdaM
Loss Function	MSE
Number of Epochs	20
Batch Size	It depends on the scenario in our project
Model Accuracy	It is calculated based on MAPE

steps before updating the model parameters to enhance its performance [34]. The training dataset samples should be divided evenly into batches for better results (Equation 10), this is where batch size should be chosen carefully.

$$No.\ of\ Batches = \frac{Training\ Samples}{Batch\ Size} \tag{10}$$

We found the model accuracy with the help of the MAPE metric (Equation 11). MAPE is a statistical measure used to evaluate the forecasting model error percentage [35]. The lower the MAPE, the better the model performance. For each scenario interpreted, the model accuracy is calculated by Equation 12.

$$MAPE(h) = \frac{1}{N}\sum_{1}^{N} |\frac{y_t(t+h) - \hat{y_t}(t+h)}{y_t(t+h)}| \times 100 \tag{11}$$

where $(t+h)$ is the lead time instant in the forecast horizon, and N is the number of forecasting time-steps.

$$Model\ Accuracy = 100 - MAPE \tag{12}$$

A. Model Training

Model training is the process of feeding input data and its corresponding actual output values to an algorithm for the model to learn and identify the structure of features that best represents the dataset [36]. However, the iterative process is called model fitting.

We have to divide the dataset into training and testing samples. In addition, reshaping the input data into a three-dimensional shape in compliance with the LSTM layer requirements. The study serves three different scenarios that we will discuss below.

- Scenario one: data resampled over 15 minutes (70% for Training and 30% for Testing).
- Scenario two: data resampled over days, weeks, and months (70% for Training and 30% for Testing).
- Scenario three: data resampled over days, weeks, and months (Training of all Previous time steps to Forecast Last One).

1) Scenario One: Data Resampled over 15 Minutes : First, the data was resampled over 15 minutes to match the conventional sampling rate of smart meters in the market while optimizing computation time [37]. We calculated the mean of the resampled data. Then, we reduced the data size from $(2,075,259 \times 6)$ to $(138,352 \times 6)$ by taking the average of every 15 minutes and assigning it to a single time step. Based on trial and error, we chose the number of LSTM units to be 90. Second, we split the dataset into a 70% train-data that is three years with a dimension of (105120, 1, 6) and a 30% testing-data that is one year with a dimension of (33231, 1, 6). Noting that the sample numbering starts from zero. Third, the batch size of this scenario equals 120 to distribute the training samples evenly among batches.

2) Scenario Two: Data Resampled over Days, Weeks, and Months : First, the dataset was resampled over a day, a week, and a month and saved in different files of (.*txt) format. After this step, we reduced the data size from $(2,075,259 \times 6)$ to (1442×6) for days, (207×6) for weeks, and (48×6) for months, respectively. Based on trial and error, we chose the number of LSTM units to be 100. Second, we split the dataset into 70% training data and 30% testing data. The train-date with dimensions of (1008, 1, 6) for days, (144, 1, 6) for weeks, and (32, 1, 6) for a month. The testing data with dimensions of (433, 1, 6) for days, (62, 1, 6) for weeks, and (15, 1, 6) for a month. Third, we set the batch size in a way that distributes the train samples evenly. The batch size of this scenario equals 42 for daily forecasting, 36 for weekly forecasting, and 8 for monthly forecasting.

3) Scenario Three: Data Resampled over Days, Weeks, and Months : In this scenario, we used the dataset that previously resampled over a day, week, and month. We divided the datasets into training data of all time steps that are the dimension of (1440, 1, 6) for days, (205, 1, 6) for weeks, and (46, 1, 6) for a month and testing data of last time step that is a dimension of (1, 1, 6) for days, weeks, and months. We proposed this scenario since it is more similar to what happens in real-life at the forecasting workstations, as operators (forecasters) use all the previous data instances to reach better forecasting for the next day, week, or month. Then, the actual value can be appended to the dataset and used for future prediction. Based on trial and error, we chose the number of LSTM units to be 128. Lastly, we set the batch sizes to 120 for daily forecasting, 41 for weekly forecasting, and 2 for monthly forecasting.

B. Model Testing

Model testing is the process of evaluating the trained model, through its performance metrics, by the testing data [36]. Each testing instant has a known value of output $(y_t(t+h))$, the model predicts $(\hat{y_t}(t+h))$, and then it compares both values. Going back to the time-series area, the prediction error for one node is the difference between both the predicted data value and the actual measured data value at that node for each lead time instant $t+h$ in the forecast horizon. The prediction error discussed can be best described by Equation 13.

$$e_t(t+h) = y_t(t+h) - \hat{y_t}(t+h) \tag{13}$$

The performance metrics give an insight into the accuracy of the developed model. Equations 14, 9, 15, and 16 describe the used metrics: MAE, MSE, RMSE, and R^2 score, respectively. The MAE measures the average absolute error of the complete set of predictions through the whole forecasting horizon. The MSE measures the average of the squares of the errors. The RMSE measures the accuracy of the predicted results [38], [39]. Lastly, the R^2 score ranges between [0, 1], and it expresses the changes in the target feature that is forecasted using the other independent features of the model [40]. The

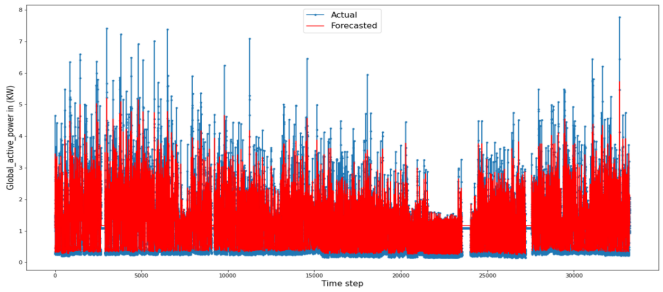

Fig. 7. Actual versus forecasted data of scenario one

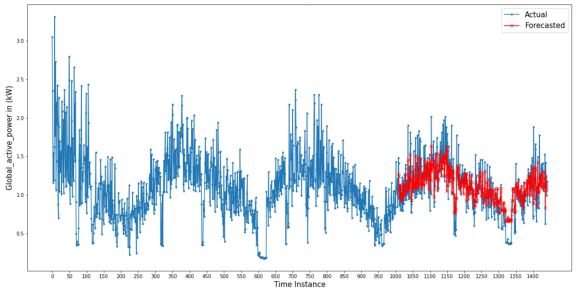

Fig. 8. Daily forecasting actual vs. forecasted data of scenario two

TABLE II
PERFORMANCE EVALUATION OF SCENARIO ONE

Performance Metric	Resolution
MAE	0.287
MSE	0.220
RMSE	0.469
R^2 Score	0.709
Model Accuracy	61.66%

lower MAE, MSE, and RMSE, the better the performance. However, the lower the R^2 score, the worse the performance.

$$MAE(h) = \frac{1}{N}\sum_{1}^{N}|e_t(t+h)| \tag{14}$$

$$RMSE(h) = \sqrt{\frac{1}{N}\sum_{1}^{N}e_t^2(t+h)} \tag{15}$$

$$R^2\ score(h) = 1 - \frac{\sum_1^N(\hat{y}_t - \bar{y})^2}{\sum_1^N(y_t - \bar{y})^2} \tag{16}$$

Where $(t+h)$ is the lead time instant in the forecast horizon, N is the number of forecasting time-steps, and $\bar{y}$ is the mean of feature y (Global_active_power).

V. RESULTS

The results of forecasted data versus actual data points are illustrated with its corresponding performance metrics.

1) Scenario One: Data Resampled over 15 Minutes: Fig. 7 shows the forecasted data points against the actual ones due to a large number of samples in the training phase. In addition, Table II shows the achieved performance metrics and the developed model accuracy.

2) Scenario Two: Data Resampled over Days, Weeks, and Months: Figures 8, 9, and 10 illustrate the whole range of samples (including training and testing phases) as well as the forecasted data points. Moreover, Table III shows the corresponding performance metrics and the developed model accuracy.

TABLE III
FORECASTING PERFORMANCE EVALUATION OF SCENARIO TWO FOR (A) DAILY, (B) WEEKLY, AND (C) MONTHLY BASIS

Performance Metric	Resolution		
	A	*B*	*C*
MAE	0.192	0.151	0.141
MSE	0.064	0.035	0.024
RMSE	0.253	0.187	0.154
R^2 Score	0.402	0.457	0.462
Model Accuracy	79.32%	84.15%	85.89%

3) Scenario Three: Data Resampled over Days, Weeks, and Months: Figures 11, 12, and 13 illustrate the whole range of samples (including training and testing phases) as well as the forecasted data points. In addition, Table IV shows the corresponding performance metrics and the developed model accuracy. As for the R^2 score in this scenario, it returns (NaN) because this metric is not suitable for single data points but

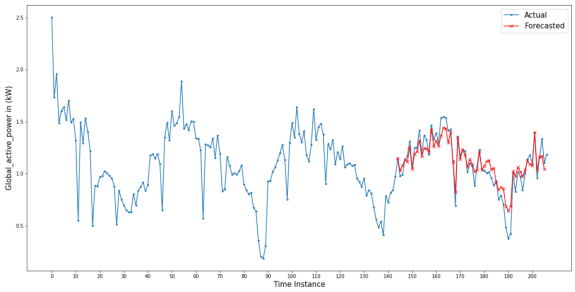

Fig. 9. Weekly forecasting actual vs. forecasted data of scenario two

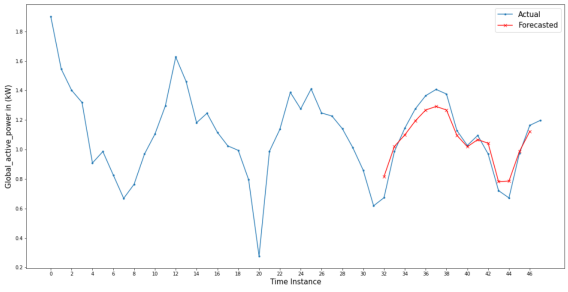

Fig. 10. Monthly forecasting actual vs. forecasted data of scenario two

TABLE IV
FORECASTING PERFORMANCE EVALUATION OF SCENARIO THREE FOR (A) DAILY, (B) WEEKLY, AND (C) MONTHLY BASIS

Performance Metric	Resolution		
	A	*B*	*C*
MAE	0.121	0.140	0.034
MSE	0.015	0.020	0.001
RMSE	0.121	0.140	0.034
R^2 Score	NaN	NaN	NaN
Model Accuracy	88.22%	88.96%	97.16%

at least two samples [41].

VI. DISCUSSION

The single-layer LSTM model inputted N number of time steps that has six features and forecasted for M number of time-steps with each step representing a certain period (15 minutes, a day, a week, or a month). The results show that the

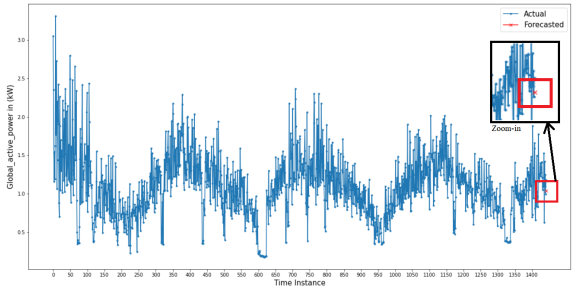

Fig. 11. Daily forecasting actual vs. forecasted data of scenario three

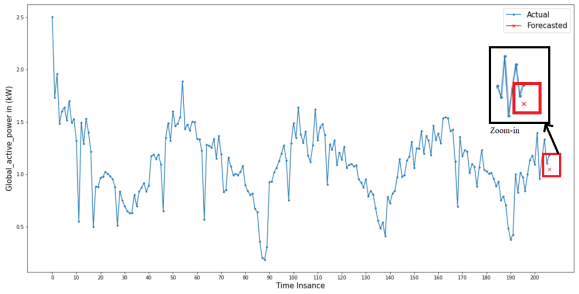

Fig. 12. Weekly forecasting actual vs. forecasted data of scenario three

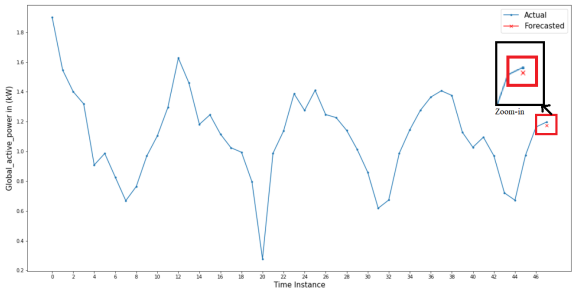

Fig. 13. Monthly forecasting actual vs. forecasted data of scenario three

TABLE V
PERFORMANCE EVALUATION PF POWER FORECASTING SYSTEM (SCENARIO ONE) WITH AND WITHOUT SUB_METERING_4 FEATURE

Scenario One	With Sub_metering_4	Without Sub_metering_4
MAE	0.287	0.305
MSE	0.220	0.225
RMSE	0.469	0.474
R^2 Score	0.709	0.703
Model Accuracy	61.66%	55.7%

forecasted points followed the same trend as the actual ones, with a minor error as measured by the performance metrics.

In addition, for the second and third scenarios, the prediction results vary depending on the hyperparameters set to the model. Mainly, we chose the values of the parameter based on trial and error.

The results of forecasting were better after the inclusion of the Sub_metering_4 feature in the dataset. Table V shows the outcome of scenario one, with and without the (Sub_metering_4). The execution time of the developed subsystem in the three scenarios ranges between 7.57 seconds and 100.55 seconds. External factors such as the processing unit of our personal computers, the excellence of coding writing style, and the number of code lines affect the execution speed. The different models achieved an excellent performance ranging from 61.66% to 97.16%. The best model performance was attained by the monthly forecasting one in scenario three with a model accuracy of 97.16%. As stated earlier, we can utilize this scenario in real-life situations where a utility company forecasts the future monthly consumption by including the demand of a district or a city rather than a single household.

VII. CONCLUSION

The main goal was to develop a system engineered towards analysing and forecasting house consumption data. We chose this system because of its future potentials of being integrated into smart cities' research and applications. We developed and tested the system on the UCI dataset after exploratory analysis and preprocessing were conducted. Three different forecasting scenarios were developed with distinct sampling rates to validate the performance under different conditions. For each scenario, parameters were tweaked using trial and error until the best result was achieved. The entirety of the system was coded using the Python programming language. The forecasting results showed an overall excellent performance.

VIII. FUTURE WORK

For future work, there are several aspects to be considered in the design. We can apply an anomaly detection algorithm on the UCI dataset as an additional pre-processing step that will further enhance the results of the forecasting models. Also, the adaptation of extra layers in the model stack could be applied in a way that increases the model accuracy regardless of the imposed complexity and rise of execution time.

REFERENCES

[1] *A renewable energy market*, publication Title: Saudi Vision 2030 Type: gov. [Online]. Available: https://vision2030.gov.sa/en/node/87

[2] A. Tuovila, "Forecasting Definition," Sep. 2020. [Online]. Available: https://www.investopedia.com/terms/f/forecasting.asp

[3] "Electric Load Forecasting: Advantages and Challenges," Oct. 2016. [Online]. Available: https://engineering.electrical-equipment.org/electrical-distribution/electric-load-forecasting-advantages-challenges.html

[4] Y. Hong, Y. Zhou, Q. Li, W. Xu, and X. Zheng, "A Deep Learning Method for Short-Term Residential Load Forecasting in Smart Grid," *IEEE Access*, vol. 8, pp. 55 785–55 797, 2020.

[5] M. Yuvaraju and M. Divya, "Residential Load Forecasting by Recurrent Neural Network on LSTM Model," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. Madurai, India: IEEE, 2020, pp. 395–400.

[6] Q. Peng and Z.-W. Liu, "Short-Term Residential Load Forecasting Based on Smart Meter Data Using Temporal Convolutional Networks," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 5423–5428.

[7] S. Makonin, "AMPds2: The Almanac of Minutely Power dataset (Version 2)," 2020. [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/FIE0S4

[8] J. Z. Kolter and M. J. Johnson, *REDD: A public data set for energy disaggregation research*, 2011. [Online]. Available: http://redd.csail.mit.edu/

[9] S. Welikala, C. Dinesh, M. P. B. Ekanayake, R. I. Godaliyadda, and J. Ekanayake, "Incorporating Appliance Usage Patterns for Non-Intrusive Load Monitoring and Load Forecasting," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 448–461, 2019.

[10] A. Estebsari and R. Rajabi, "Single Residential Load Forecasting Using Deep Learning and Image Encoding Techniques," *Electronics*, vol. 9, no. 1, p. 68, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/1/68

[11] D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, no. 1, pp. 81–102, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0095069678900062

[12] G. Hebrail and A. Berard, *Individual Household Electric Power Consumption Dataset*, publication Title: UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption

[13] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector Machines: a study on EUNITE competition 2001," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[14] D. Park, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.

[15] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1483–1488.

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

[17] Government of Canada, *The Daily — Households and the Environment Survey: Energy use, 2013*, 2016, publication Title: Statistics Canada. [Online]. Available: https://www150.statcan.gc.ca/n1/daily-quotidien/160318/dq160318d-eng.htm

[18] H. Jun, C. Haoyuan, X. Zhenjian, J. Wei, Z. Jia, D. Jian, C. Chao, and W. Na, "A Novel Short-term Residential Load Forecasting Model Combining Machine Learning Method with Empirical Mode Decomposition," in *2020 Asia Energy and Electrical Engineering Symposium (AEEES)*, 2020, pp. 816–820.

[19] S. M. M. Alam and M. H. Ali, "A New Subtractive Clustering Based ANFIS System for Residential Load Forecasting," in *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2020, pp. 1–5.

[20] X. Wang and S.-H. Ahn, "Real-time prediction and anomaly detection of electrical load in a residential community," *Applied Energy*, vol. 259, p. 114145, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030626191931832X

[21] M. Jones, "Chapter 5 - Faultfinding to Fettling," in *Building Valve Amplifiers*, ser. second edition, M. Jones, Ed. Oxford: Newnes, Jan. 2014, pp. 381–429. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080966380000059

[22] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[23] C. Olah, *Understanding LSTM Networks*, 2015, publication Title: Colah's Blog. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] J. Brownlee, "How to Reshape Input Data for Long Short-Term Memory Networks in Keras," Aug. 2017. [Online]. Available: https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/

[26] A. M. Sefidian, *How to Reshape Input Data for Long Short-Term Memory Networks in Keras*, 2019, publication Title: Amir Masoud Sefidian. [Online]. Available: http://www.sefidian.com/2019/08/19/how-to-reshape-input-data-for-long-short-term-memory-networks-in-keras/

[27] Subrat, *Neural Network - What is num_units in Tensorflow BasicLSTMCell?*, 2016, publication Title: Stack Overflow. [Online]. Available: https://stackoverflow.com/questions/37901047/what-is-num-units-in-tensorflow-basiclstmcell

[28] J. Brownlee, *Overfitting and Underfitting With Machine Learning Algorithms*, 2016, publication Title: Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

[29] Keras Team, *Keras Documentation: Dropout layer*. [Online]. Available: https://keras.io/api/layers/regularization_layers/dropout/

[30] S. P. Singh, *Fully Connected Layer: The Brute Force Layer of a Machine Learning Model*, 2019, publication Title: OpenGenus IQ: Learn Computer Science. [Online]. Available: https://iq.opengenus.org/fully-connected-layer/

[31] Algorithmia, *Introduction to Loss Functions*, 2018, publication Title: Algorithmia Blog. [Online]. Available: https://algorithmia.com/blog/introduction-to-loss-functions

[32] J. Brownlee, "Loss and Loss Functions for Training Deep Learning Neural Networks," Jan. 2019. [Online]. Available: https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/

[33] J. Leonel, *Hyperparameters in Machine/Deep Learning*, 2019, publication Title: Medium. [Online]. Available: https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981

[34] J. Brownlee, *Difference Between a Batch and an Epoch in a Neural Network*, 2018, publication Title: Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

[35] E. Stellwagen, "Forecasting 101: A Guide to Forecast Error Measurement Statistics and How to Use Them," Aug. 2011. [Online]. Available: https://www.forecastpro.com/Trends/forecasting101August2011.html

[36] H. Madsen, P. Pinson, G. Kariniotakis, H. A. Nielsen, and T. S. Nielsen, "Standardizing the Performance Evaluation of Short-Term Wind Power Prediction Models," *Wind Engineering*, vol. 29, no. 6, pp. 475–489, 2005. [Online]. Available: http://journals.sagepub.com/doi/10.1260/030952405776234599

[37] P. Vadda and S. M. Seelam, "Smart Metering for Smart Electricity Consumption," Master's thesis, Blekinge Institute of Technology, Sweden, 2013.

[38] H. Shaker, H. Zareipour, and D. Wood, "On error measures in wind forecasting evaluations," 2013, pp. 1–6.

[39] Y. Wang, Y. Liu, J. Yan, and S. Han, "Research on Applicability of Ultra-short Term Wind Power Forecasting Models," pp. 6.1.3–6.1.3, 2014. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0869

[40] *Coefficient of Determination - R squared value in Python - AskPython*. [Online]. Available: https://www.askpython.com/python/coefficient-of-determination

[41] Scikit, *sklearn.metrics.r2_score — scikit-learn 0.24.1 Documentation*, publication Title: Scikit Learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html