AI-based Energy Consumptions Predictions in Smart Grids

Raissa Raffi Darmawan
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 15143
raissa.darmawan@binus.ac.id

Jordan Pascal Wijaya
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 15143
jordan.wijaya001@binus.ac.id

Nadia
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
nadia002@binus.ac.id

Franz Adeta Junior
Computer Science Department,
School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
franz.junior@binus.ac.id

Abstract— ***Smart Grid is an electrical network technology that can regulate electricity supply and maintain its operations. Smart Grid can help Indonesia overcome the excess energy supply that occurred as a result of the creation of mega-electricity projects aimed at meeting Indonesia's electricity needs. However, due to COVID-19, electricity demand in Indonesia has decreased, resulting in an excess supply of electrical energy. Therefore, the aim of this research is to optimize existing electrical energy. To be able to eliminate wasted energy and eliminate losses that occur. The methods we use are artificial neural networks, convolutional neural networks, recurrent neural networks, and long short-term memory. The samples we got from the weather dataset from Indonesia in 2018–2020 from Badan Meteorologi, Klimatologi, and Geofisika (BMKG) Indonesia. We use this dataset to be able to consider the dependency between energy use and weather data. Comparison of (ANN), (CNN), (RNN), and (LSTM). The experimental results of the data set we obtained were that the CNN model outperformed any model. Through experiments and research that the method (CNN) reduces overfitting better than other methods. This approach can be proposed to help predict peak demand periods and automatically control smart equipment to reduce consumption and reduce energy consumption costs for users.***

Keywords— Smart Grid, Forecasting, Deep Learning

I. INTRODUCTION

Smart Grid (SG) is a way for technology to prevent the increasing electricity demand that people currently have. The Smart Grid (SG) allows integration of energy distribution and digital communication technology in a two-way flow of electricity and data, meaning companies are enabled to optimize the generation, transmission and distribution of electricity. As for the consumers, they also benefit from the data that helps them better understand the energy that they use and the energy that they produce and store them through renewable energy such as solar panels and EV batteries. In the current state of Indonesia, electricity demand keeps increasing and thus the project of 35,000-megawatt (MW) program is deployed. It turns out the project is soon to be the start of an oversupply problem that has been occurring in Indonesia for several years now. This happens due several reasons including the economic issue that slowed down during the Covid-19 pandemic. As per 2023, oversupply electricity in Indonesia on average is above 40%. As for electricity usage per region in Indonesia has not reached 100%, this includes regions such as West Kalimantan, East Nusa Tenggara, and Eastern Indonesia [1].

With the increasing electricity demand in Indonesia, it needs a way for us to balance the energy supply and the energy demand more dynamically. One way is to start constructing a Smart Grid System with AI powered that could aid those needs. The AI contributes in predicting the energy generation so that there is this minimal gap between the current energy generation and the predicted energy generation. In order to build the Smart Grid (SG), reliable and accurate predictions are required to be planned and made prior to its installation [2].

To overcome the problem of excess energy supply in Indonesia, implementing a Smart Grid system with artificial intelligence (AI) is a potential solution. With the help of AI technology, such as Deep Learning (DL) and Natural Language Processing (NLP) methods, accurate predictions regarding electrical energy generation can be made. One suitable approach is the use of a Deep Learning model called Long Short-Term Memory (LSTM), which is able to overcome challenges in time series data analysis. LSTM allows systems to learn patterns and trends from historical data, enabling more reliable predictions of future energy needs. By utilizing LSTM in a Smart Grid system, it can be ensured that energy production can be adjusted dynamically to actual demand, reducing the risk of excess energy supply and reducing economically detrimental energy waste. Thus, the use of the LSTM method in Smart Grid development is a progressive step towards developing efficient and sustainable energy infrastructure in Indonesia [3].

II. LITERATURE REVIEW

In this section, we discuss a brief explanation of the literature that we collected which consists of the types of methods that are used, the results, the limitations, and the

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©2024 IEEE

advantages of each method based on the papers that we have collected.

Paper from [4] refers to a neural network, which mostly consists of a set of nodes or units with a set of directed edges adjacent to them. An Artificial Neural Network (ANN) refers to any representation of neural networks such as, feedforward, convolutional, recurrent, radial bias function, etc. We can demonstrate a simple architecture of multi-layered feedforward neural networks as a "Processing Elements" (PE) because of their ability to process data. Each processing element (PE) consists of weighted inputs, a hidden layer as the transfer function, and one output. The way it works is that some of the neurons in a given structure creates an interface with real environments to accept the inputs. After that, the other neurons will deliver the network outputs to the real world. Other neurons that remain are hidden from view. The output could be things such as characters that the network believes it has scanned or an image that it believes is being displayed. Many different types of Artificial Neural Networks (ANN) architecture are developed through different connections and computations such as convolutional neural networks (CNN) and recurrent neural networks (RNN). The advantages of having ANN as a method of forecasting or prediction in an AI Smart Grid system is that it is capable of producing surprising accurate predictions and is able to absorb large numbers of data. The paper from [4] however, shows that a simple ANN showed worse performance than other models such as the Deep Learning model simply because of the size of the datasets that are very large (big data). But, for datasets that contain not very large data, a simple ANN would be able to perform well compared to other models such as the RNN, LSTM, BiLSTM, and GRU model [4].

A method from paper [5] suggests the method Fuzzy Logic which is recognized as a significant component in Artificial Intelligence (AI) research for the Smart Grid. In recent years, there has been a notable increase in the application of AI solutions to enhance the efficiency and sustainability of energy networks. Fuzzy Logic offers a promising approach to address the complex challenges of the Smart Grid, such as optimizing renewable energy usage and supporting real-time demand response. However, while Fuzzy Logic shows promise in addressing these challenges, each paper presents its own set of strengths and weaknesses in its implementation for the Smart Grid. Some papers demonstrate the effectiveness of Fuzzy Logic in optimizing renewable energy usage and facilitating real time demand response, thereby improving grid stability and efficiency. On the other hand, certain limitations such as computational complexity and difficulty in parameter tuning have been identified in other papers. Therefore, while Fuzzy Logic holds potential, further research is needed to address these drawbacks and refine its application in the context of the Smart Grids [5].

The following paper from [6] describes the application of Deep Learning for Supply Forecasting LSTM model and its architecture using a proposed Supply Forecasting Framework that can be explained by each step, that includes Step 1: Goal, Step 2: Data. Step 3: Model, Step 4: Forecasting, and Step 5: Performance Evaluation. From step 1, the model provides an energy supply forecasting measured as Global Horizontal Irradiance (GHI) for PV panels with an extensive range of forecasting horizons data with a range from minute-ahead (15 min to 1 hour) to hours ahead (1 to 24 hours) to days ahead (1 to 7 days). A pyranometer then was installed on the PV panels to obtain the GHI data and stored in a server as raw data. Depicted from step 2, The measurement from that frequency is 10 seconds. In step 3, the raw data is pre-process and put in the Deep Learning models and then post-process it in order to output the GHI forecasts or predictions with horizons ranging 15 minutes to 7 days ahead. Then in step 4, the model then displays the predicted mean irradiance during the next daylight hours of the given horizon. Lastly, in step 5, the model measures the accuracy of the proposed Deep Learning Model using a time-series cross-validation and performance metrics. The paper provides a comprehensive explanation on how to use the proposed Deep Learning Model that can help in forecasting AI-based Smart Grid (SG) system, only one thing that lacks is a that the lack of inputs to the proposed forecasting model that can be added more to provide a more diverse result such as ground measurements air temperature or humidity, satellite imagery, or other measurements [6].

From paper [7], Load Forecasting Techniques (LFTs) and their applications in Smart Grids (SGs). Various methods, including traditional approaches, clustering-based techniques, artificial intelligence (AI)-based models, time series-based methods, and meta-heuristic algorithms, have been explored in recent studies. These techniques exhibit diverse strengths, with AI-based methods, particularly machine learning (ML) and neural network (NN) models, demonstrating superior performance in terms of accuracy and efficiency. ML and NN models have shown promising results in handling complex and non-linear relationships in data, enhancing forecasting precision in SGs. However, despite their advantages, LFTs may suffer from limitations such as computational intensity, difficulty in interpretation, and potential lack of generalizability to other applications. These drawbacks underscore the need for further research to address these challenges and optimize LFTs for SGs applications. Overall, the evolving landscape of LFTs offers promising opportunities for improving the reliability, stability, and efficiency of SGs through more accurate and efficient load forecasting.

The LSTM model mentioned in paper [8] was used to predict an energy generation to utilize it for a recommender system for the AI-empowered Smart Grid using datasets of residential houses which have high demand in energy consumptions. From what the research has analyzed, the proposed approach was able to obtain a better result of predictions compared to the research baseline approaches. As a result, the proposed recommender system can strive to close the gap between energy demand and response [8].

Based on the literatures we have read, we have concluded that the CNN model is a method that is applicable for smart grid in cases where the data is not much which will have a better performance. However, in cases where there is too much data, the LSTM model seems to be an appropriate method for that task.

III. METHOD

In this part, we will discuss a simple model architecture based on the literature review we have done which is the CNN model. We will be using methods that are applicable in an AI-powered Smart Grid (SG) by using weather data forecasting or prediction with deep learning methods such as ANN, CNN, RNN, and LSTM.

Convolutional neural network (CNN) is a class of deep neural networks which are typically used in analyzing a visual of an imagery or image-recognition task. There are typically three layers in the CNN which consist of a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer performs a dot product between two matrices where one is the set of learnable parameters otherwise known as a kernel, the other matrix is the restricted portion of the receptive field. The pooling layer is a replacement of an output of the network at certain locations by using a summary statistic of nearby outputs. It helps reduce the spatial size of the representation which decreases the amount of computation and weights. Lastly, a fully connected layer has full connectivity in its neurons with all neurons in the preceding and succeeding layer, this layer helps to map the visualization between the input and the output of the model as can be seen from **Fig. 1**.

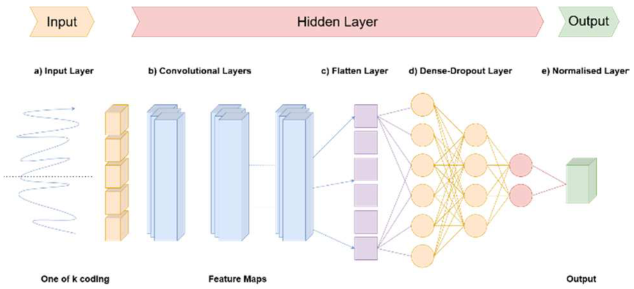

Fig. 1. 1-D CNN Architecture.
Source: [9]

IV. EXPERIMENT

A. Data Description

In this paper, we are using real datasets of weather data from a daily climate dataset that is set in Indonesia. Dataset was obtained from the website Kaggle.com with a couple of adjustments to the datasets so that it is applicable to our model. For example, from TABLE II, we filtered the daily climate dataset to only range from the start of 2018 to the end of 2020 as there was an excessive amount of data ranging from 2010-2020 that could potentially slow down the simple model that we are creating.

a). Data Sources

In these datasets, we used variables that vary. For the weather data, there are 10 variables that are used as features or input for our model and one target model that will be obtained as the output data. The input data consists of climate related variables that cover almost all regions in Indonesia, as can be seen from TABLE I (Min Temperature (°C), Max Temperature (°C), Avg Temperature (°C), Avg Humidity (%), Rainfall (mm), Duration of Sunshine (hour), Max Wind Speed (m/s), Wind Direction at Maximum Speed (°), Avg Wind Speed (m/s), Most wind direction (°)). Then we set the Average Temperature (°C) as the target variable.

b). Data Preprocessing & Cleaning

The weather datasets were loaded as a csv file that were filtered to only range from 2018 to 2020 and set in Indonesia. Datasets that have columns which contain a NaN value will be filled with the mean of each column so that it can still be used for our model. Selecting an excessive amount of features will slow down the model, therefore we select data that are relevant to our case, as mentioned in the Data Sources above. From those selected data, then we handle it by selecting only columns that are a number and not a non-numeric. Data were then normalized using MinMaxScaler from the sklearn library in python. We defined a target variable as our output data which is the Average Temperature (°C) variable as our target variable. And to finish the preprocessing, we need to split the data into two parts, the training sets and the testing sets so that we can use it in our model.

TABLE I. WEATHER DATA VARIABLES

Variable	**Abbreviation**	**Measurement**
Min Temperature	Tn	°C
Max Temperature	Tx	°C
Average Temperature	Tavg	°C
Average Humidity	RH_avg	%
Rainfall	RR	mm
Duration of Sunshine	ss	hour
Max Wind Speed	ff_x	m/s
Wind Direction at Maximum Speed	ddd_x	(degrees °)
Average Wind Speed	ff_avg	m/s
Most Wind Direction	ddd_car	(degrees °)

TABLE II. ALGORITHMS AND DATASETS USED SUMMARY

Method	Data	Period
ANN	Daily	01/01/2018 to 31/12/2020
CNN	Daily	01/01/2018 to 31/12/2020
RNN	Daily	01/01/2018 to 31/12/2020
LSTM	Daily	01/01/2018 to 31/12/2020

B. *Model Training and Evaluation*

a) *Training Procedure*

Several series steps were taken to train four predictive models. First, the data was split into two parts, training and testing sets using 'train_test_split' from scikit-learn. Specifically, the data was split into portions which in this study case, 80% of the data were used for the training and 20% were used for testing. For each model, we used hyperparameters to support our architectures that can be described as follows:

- The ANN model used a number of epochs that is set to 100 with a batch size of 32. The network then included hidden layers with specified numbers, for example a dense layer with 50 layers. ReLU is the activation function used for the hidden layer as well as a linear function for the output layer using an optimizer of Adam to update the model parameters. A lost function used was Mean Squared Error.
- The CNN model used a number of epochs that is set to 50 with a batch size of 32. This network includes layers such as convolutional layers which are Filters, which is set to 64, Kernel Size, which is set to 2, and an Activation Function that uses ReLU. A pooling layer, which is a MaxPooling layer that uses a pool size of 2. A dense layer that included 50 neurons and an activation function of ReLU and an output layer with a linear activation function. As for the optimizer, it uses Adam and a loss function of Mean Squared Error.
- The RNN model used a number of epochs that is set to 50 with a batch size of 32. The layers include RNN layer, which are units or a number of neurons in the RNN layer which is set to 50, and input shape which shape the input data as timesteps and features. A dense layer is included too which is a single layer with 1 neuron for the output. An optimizer used in this model was Adam with a loss function of Mean Squared Error.
- The LSTM model used a number of epochs that is set to 50 with a batch size of 32. The layers included are a LSTM layer which are units of neurons which are set to 50 with an input shape that shapes the input data as timesteps and features. A dense layer is a single layer with 1 neuron for the output. It uses Adam as the optimizer with a loss function Mean Squared Error.

b) *Evaluation Metrics*

A commonly used performance prediction accuracy is the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are used in this study, we also use a statistical measurement which is the R^2, we can define the evaluation metrics as below:

$$RMSE = \frac{1}{N}\sum_{i=1}^{N} \left|\left(y_{predicted} - y_{actual}\right)\right| \quad (1)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_{actual} - y_{predicted}\right)^2}{\sum_{i=1}^{N} \left(y_{actual} - \underline{y}\right)^2} \quad (2)$$

In these formulas, the N represents the number of observations, the y_{actual} is the actual value and $y_{predicted}$ is the predicted value. The RMSE is the squared difference between the predicted and actual values [10]. The RMSE provides a suitable option for regression tasks where the target variable is continuous which makes it applicable to our dataset, the lower RMSE values indicate a better fit model. The MAE helps find the best fitting model by reducing the overall prediction error. The R^2 is a statistical measure that represents the proportion of the variance for a dependent variable which is explained by variables in a regression model, this makes it intuitive as the R^2 ranges from 0 to 1 where higher value means a better model performance.

C. *Implementation Details*

The data was split and normalized as well as calculated as stated in the evaluation metrics such as RMSE and R2 Score using the Scikit-learn library for python. The data went through different steps such as cleaning, preprocessing, and handling structured data in DataFrames using the Pandas library. To scale the features to range between 0 and 1, the model uses the MinMaxScaler which is crucial for ensuring the neural networks to train effectively. The models were built and trained using a deep learning tool which is TensorFlow. Keras was also used to define, compile, and train the ANN, CNN, RNN, and LSTM models. NumPy was used for numerical operations such as performing calculations for the MSE. Also, a plotting library Matplotlib was used to generate visualizations of plots of training history and actual vs. predicted values. Models were trained on a standard CPU and sped up the training process by a GPU to optimize the parallel processing which is beneficial for matrix operations involved in neural networks.

V. RESULTS & DISCUSSION

A. *Model Performance*

Based on TABLE III, we can conclude that the CNN model has the best performance as stated in the literature review with CNN marginally outperforming the other models in terms of RMSE and R-squared score. The predicted RMSEs values are in °C as we are using the average temperature as the target variable. Those RMSEs values using ANN, CNN, RNN and LSTM models have a value of 0.006551 °C, 0.006493 °C, 0.006902 °C, 0.006754 °C. So on average, we can see that the prediction made by the CNN model has relatively the lower RMSE and highest R-Squared score compared to the other models, note that these are the results of a simple model using weather data and may differ depending on the amount of datasets being used. The models that is being used in [10] used weather environmental variables (i.e., pressure (hPa), temperature (°C), relative humidity (%), wind velocity (m/s), rainfall duration (min), rainfall amount (mm)), and type of (occupancy related) data as their input data.

TABLE III. MODEL PERFORMANCE SUMMARY

Model	RMSE (°C)	R^2 Score
ANN	0.006551	0.781959
CNN	0.006493	0.785797
RNN	0.006902	0.757944
LSTM	0.006754	0.768243

From the ANN model, we can observe that the scatter plot of the actual vs predicted values for the ANN model has a positive correlation indicating that the model predictions as a whole follows the trend of the actual values. Interestingly enough, a dashed line can be seen from the scatter plot particularly around certain values. This suggests that there are some deviations that could indicate some predictions errors that occur during the training process.

From the ANN model, the training and validation loss graph can be represented by the X-axis and the Y-axis where the X-axis represents the number of epochs or iterations during the training process, and where the Y-axis represents the loss value as the colors suggest. A quick drop within the first few epochs from the losses between the training and the validation suggests that the model is learning quickly from the given weather data. A plateau can be noticed after the initial drop as both losses continue to drop but at much slower rate and eventually plateauing, suggesting that the model is stabilized.

Similar to the ANN model, based on **Fig. 2** and **Fig. 3**, the CNN model indicates a solid performance with a training and validation loss that are relatively stable and low which suggests that the model has learned the data well and not overfitting. As for the scatter plot, it has a strong correlation between the predicted and actual values with a couple of errors. Compared to the previous model which is the ANN model, the CNN model has more variability in terms of validation losses and has a slightly different pattern of deviation from the scatter plot but overall, both models show a similar trend.

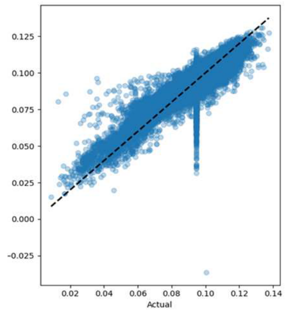

Fig. 2. Predicted vs. Actual values plot using CNN.

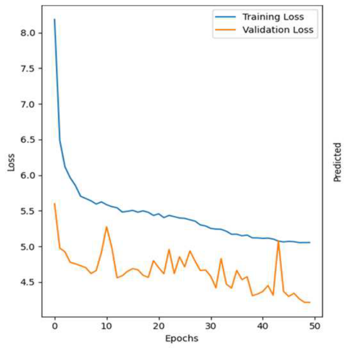

Fig. 3. Training & Validation loss using CNN.

A clear positive relationship also can be seen between the predicted and actual values for the RNN model. Around the line when y = x, the majority of points are clustered which is ideal for this scenario meaning the predicted values are equal to the actual values. Although there are some noticeable deviations from the ideal line as can be seen a vertical line of points at approximately x = 0.10, y = 0.08.

Based on the RNN model, both the validation and training losses decrease and converge over time. Which could indicate that the model is not underfitting drastically. A gap can also be seen between the training and validation loss, which is relatively small. This could also indicate that the model generalizes well to new data. The fluctuations in the graph can be caused due to variability in the validation data. In the final epochs, the validation loss shows a slight increase while the training loss is decreasing. This could mean that there is a slight overfitting in the model. Furthermore, an overfitting like this shouldn't be much of a concern because the increase is minor and has minimal significant issues.

A general pattern can be seen from the LSTM model, which is that the predicted values closely follow the actual values making the model to have accurate predictions. The overlap in the graph suggests that the model effectively captures each pattern in the data. A few spikes can also be seen where the predicted values significantly deviate from the actual values. These deviations could mean that there are some periods that the model is struggling to predict accurately. However, the deviations seem to be a rare instance and overall, the performance is still strong. Overall, the graph is consistent in terms of learning the data as the predicted values are consistently following the trend of the actual values.

An initial drop can be seen in an instant from the LSTM model, this indicates that the model is learning rapidly and reducing error on both the training and validation datasets. As the epochs progress, both the training and validation loss is continuing to decrease steadily, with some fluctuations which is normal and shows that the model is generalizing well. At the end of it, the validation loss remains close to the training loss with both being low and close to each other. Suggests that the model has reached a good fit and has no significant indication of an overfitting.

B. Error Analysis

We can identify some common errors made by the models based on the graph and the potential reasons behind it. The ANN model showed loss decrease and then plateau in the training and validation graph which indicates a stable model, however, there is a significant gap between the training and validation loss which suggests an overfitting. Similarly, the CNN model follows a similar trend with the ANN model which has decreasing loss values over epochs and an overfitting can also be seen in the graph. In terms of the scatter plots, all of the models relatively show a strong positive correlation but there are some outliers and deviations which could indicate some systematic errors made by the models, for example the vertical lines that can be seen in the ANN and CNN models suggests that there is some systematic errors at a specific actual values. The deviations that occur means that there are areas where models are less accurate. A few fluctuations in the RNN model in both training and validation losses can be seen which indicate a slight overfitting because the validation loss is generally higher than the training loss. The RNN model struggles with some generalization as the training and validation loss continue to decrease and converge. As for the scatter plot, the RNN model has noticeable deviations with some large spikes which suggests periods where the prediction accuracy made by the model is poor. The LSTM model has minimal errors which can be seen by how minor the overfitting is which indicates the model is effectively learning. But some anomalies such as occasional spikes where predictions deviate significantly can also be seen suggesting that the model is struggling with certain patterns or outliers.

VI. CONCLUSION

These models are created as a means to be predictive models to forecast or predict energy consumption that can be a good starting point on integrating it into a smart grid, especially in Indonesia. These models can be integrated to a Smart Grid (SG) with real-time prediction for future periods which is crucial for optimizing energy distribution and generation in a smart grid. A real-world application example of it based on these models is the Demand Response Management which could be effective for adjusting the demand for power instead of adjusting the supply to achieve grid stability. The models use past years daily weather data that is set in Indonesia with using features that the datasets provided and using the average temperature as the target variable. Based on the results, the ANN, CNN, RNN and LSTM model has a RMSE values of (0.006551, 0.006493, 0.006902, 0.006754) and with R-squared scores of (0.781959, 0.785797, 0.757944, 0.768243). Meaning that the CNN model is able to perform the best with the lowest RMSE value and highest R-squared score. Note that these models may differ in terms of results based on the amount of data given and other aspects. While the CNN model gives us the better result, the temporal relationships captured by the LSTM model seems more suitable for forecasting tasks. However, the relatively positive outcomes that these models give does not mean it's the most perfect one. Adjustments for future work could potentially help to improve the accuracy of the model by having better data preprocessing that could potentially change the results.

REFERENCES

[1] A. P. Redaputri, "The Condition of Excess Electricity Supply in Indonesia," *Journal of Earth Energy Engineering*, vol. 12, no. 2, Oct. 2023, doi: 10.25299/jeee.2023.12856.

[2] Adebayo Olusegun Aderibigbe, Emmanuel Chigozie Ani, Peter Efosa Ohenhen, Nzubechukwu Chukwudum Ohalete, and Donald Obinna Daraojimba, "ENHANCING ENERGY EFFICIENCY WITH AI: A REVIEW OF MACHINE LEARNING MODELS IN ELECTRICITY DEMAND FORECASTING," *Engineering Science & Technology Journal*, vol. 4, no. 6, pp. 341–356, Dec. 2023, doi: 10.51594/estj.v4i6.636.

[3] S. Salehimehr, B. Taheri, and M. Sedighizadeh, "Short-term load forecasting in smart grids using artificial intelligence methods: A survey," *The Journal of Engineering*, vol. 2022, no. 12, pp. 1133–1142, Dec. 2022, doi: 10.1049/tje2.12183.

[4] A. Alrasheedi and A. Almalaq, "Hybrid Deep Learning Applied on Saudi Smart Grids for Short-Term Load Forecasting," *Mathematics*, vol. 10, no. 15, Aug. 2022, doi: 10.3390/math10152666.

[5] N. Bassiliades and G. Chalkiadakis, "Artificial Intelligence Techniques for the Smart Grid," *Advances in Building Energy Research*, vol. 12, no. 1. Taylor and Francis Ltd., pp. 1–2, Jan. 02, 2018. doi: 10.1080/17512549.2017.1410362.

[6] P. Bouquet, I. Jackson, M. Nick, and A. Kaboli, "AI-based forecasting for optimised solar energy management and smart grid efficiency," *Int J Prod Res*, vol. 62, no. 13, pp. 4623–4644, Jul. 2024, doi: 10.1080/00207543.2023.2269565.

[7] H. Habbak, M. Mahmoud, K. Metwally, M. M. Fouda, and M. I. Ibrahem, "Load Forecasting Techniques and Their Applications in Smart Grids," *Energies*, vol. 16, no. 3. MDPI, Feb. 01, 2023. doi: 10.3390/en16031480.

[8] R. K. Patel, A. Kumari, S. Tanwar, W. C. Hong, and R. Sharma, "AI-Empowered Recommender System for Renewable Energy Harvesting in Smart Grid System," *IEEE Access*, vol. 10, pp. 24316–24326, 2022, doi: 10.1109/ACCESS.2022.3152528.

[9] R. Sánchez-Reolid, F. López de la Rosa, M. T. López, and A. Fernández-Caballero, "One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity," *Biomed Signal Process Control*, vol. 71, Jan. 2022, doi: 10.1016/j.bspc.2021.103203.

[10] M. Faiq *et al.*, "Prediction of energy consumption in campus buildings using long short-term memory," *Alexandria Engineering Journal*, vol. 67, pp. 65–76, Mar. 2023, doi: 10.1016/j.aej.2022.12.015.