Enhancing Microgrid Efficiency: A Comparative Analysis of Forecasting Techniques for Load Demand Prediction

Gwaivangmin B.I [1], Bakare G.A [2], Haruna Y.S [3] and Amoo A.L [4]
Department of Electrical and Electronics Engineering
Abubakar Tafawa Balewa University, Bauchi
bgwaivangmin@gmail.com

Received: 18-FEB-2024; Reviewed: 09-JUNE-2024; Accepted: 21-SEP-2024
https://dx.doi.org/10.4314/fuoyejet.v9i3.7

ORIGINAL RESEARCH

Abstract -This study compared three machine learning techniques - Ridge Regression, ARIMA, and Random Forest Regression - to forecast short-term electricity demand in a Nigerian university microgrid. The goal was to identify the most accurate method for predicting the university's power needs 24 hours ahead. Six years of historical load data and weather information were used to train and evaluate the models. Random Forest Regression (RFR) emerged as the clear winner, achieving a significant improvement in accuracy compared to both Ridge Regression (RG) and ARIMA. Notably, RFR offered a 45% reduction in Root Mean Squared Error (RMSE) and a 33% decrease in Mean Absolute Percentage Error (MAPE) compared to RG. These results suggest that RFR provides the most precise predictions for university electricity demand in this scenario.

Keywords: Demand, Forecasting, Load demand, Machine learning, Management, Micro grid

1. INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Microgrids are gaining popularity in remote areas, transforming power supply. These localized systems not only improve energy security and reduce outages, but also seamlessly integrate renewable sources for a more sustainable and reliable solution (Wang & Li., 2023). However, accurate short-term electricity demand prediction is crucial for efficient microgrid management (Butt et al., 2020). This enables optimized resource allocation, leading to cost savings and stable grid operation (Yu et al., 2024; Gopinathan & Annamareddi, 2015).

Traditional methods like moving averages and trend analysis are commonly used for electricity demand prediction. However, these methods struggle with the complex patterns and non-stationary data often found in microgrid load profiles (Kondaiah *et al.*, 2022).This highlights the need for more sophisticated forecasting techniques.

Machine learning is showing promise for microgrid load forecasting. Techniques like Ridge Regression and ARIMA are being explored to achieve more accurate predictions of a microgrid's electricity demand (Ji*et al.*, 2023).ARIMA, a powerful tool for time series forecasting, excels at handling diverse patterns and trends, including data with fluctuating means and variance (Box *et al.*, 2015; Ji*et al.*, 2023). Its ability to model seasonal trends makes it particularly suitable for forecasting seasonal electricity demand (Wang*et al.*, 2024). ARIMA stands out for its user-friendliness. Not only can it achieve reasonable forecasts with simpler models, but it also provides interpretable insights into the data patterns, making it easier to understand the forecast. (Hyndman & Athanasopoulos, 2021).

Ridge Regression is another prominent forecasting technique that thrives in situations with interconnected features (multicollinearity) (James *et al.*, 2021). Through a process called regularization, it avoids overfitting the data, leading to more accurate predictions on unseen data. Ridge Regression balances bias and variance, further enhancing its forecasting power. Its resilience against noise and outliers makes it a dependable tool for dealing with real-world, messy datasets (Hastie *et al.*, 2009). These characteristics make Ridge Regression a valuable tool for robust and accurate forecasting, even in challenging situations.By leveraging the strengths of machine learning techniques like ARIMA and Ridge Regression, microgrid operators can achieve more accurate short-term load forecasting. This, in turn, allows for optimized resource allocation, leading to a more efficient, cost-effective and stable microgrid operation. However, existing research also highlights

*Corresponding Author

Section B- ELECTRICAL/COMPUTER ENGINEERING &RELATEDSCIENCES
Can be cited as:
Gwaivangmin B.I, Bakare G.A, Haruna Y.Sand Amoo A.L(2024). Enhancing Microgrid Efficiency: A Comparative Analysis of Forecasting Techniques for Load Demand Prediction. FUOYE Journal of Engineering and Technology (FUOYE**JET), 9(3), 417-425.**
https://dx.doi.org/10.4314/fuoyejet.v9i3.7

limitations associated with both ARIMA and Ridge Regression. ARIMA can be computationally expensive for complex models, and its performance can be sensitive to the

selection of appropriate parameters (Wang et al., 2023). Ridge Regression, while effective in handling multicollinearity, may not be as efficient in capturing non-linear relationships within the data (James *et al.,* 2021).

This study addresses the issue of short-term electricity demand prediction in Nigerian university microgrids. It conducts a comparative analysis of three techniques: ARIMA, Ridge Regression, and Support Vector Machine (SVM) regression. By comparing these methods, the study aims to identify the most accurate approach for forecasting power needs in this specific context. By analyzing their performance using key metrics, it aims to provide valuable insights for optimizing energy management and decision-making within microgrid systems as a whole.

1.3 Literature Review

Accurate short-term prediction of electricity needs (load forecasting) is crucial for efficient microgrid management. This enables better energy use, reduces outages, and integrates renewables seamlessly, leading to a more reliable and sustainable power supply (Hatziargyriou *et al.,* 2016) The ability to accurately predict electricity demand empowers optimized resource allocation, minimized costs, and stable grid operation (Yu*et al.,* 2024). Fortunately, recent advancements in forecasting techniques provide promising tools specifically applicable to microgrid environments.

This review examines recent advancements in short-term load forecasting, highlighting the increasing adoption of machine learning and artificial intelligence techniques for improved prediction accuracy. Hybrid approaches combining optimization algorithms and neural networks have shown potential for improved forecasting accuracy. Machine learning techniques are increasingly being recognized for their effectiveness in short-term load forecasting. Supporting this approach, Salisu et al. (2019) employed a combination of Particle Swarm Optimization and an Adaptive Neuro-Fuzzy Inference System for solar radiation prediction in Nigeria. In line with this, Wang et al. (2020) proposed a combined method for short-term load forecasting, incorporating time-segmentation and an advanced neural network technique. These studies, along with others, highlight the potential

However, existing research also highlights limitations associated with both ARIMA and Ridge Regression. ARIMA can be computationally expensive for complex models, and its performance can be sensitive to the

of machine learning, particularly Artificial Neural Networks (ANNs), for enhancing the accuracy of short-term load forecasts. Further emphasizing the potential of machine learning, studies by Agboola et al. (2021) and Hammad et al. (2020) demonstrate the effectiveness of Artificial Neural Networks (ANNs) for both short-term and long-term load forecasting. Niu et al. (2021) contribute to this by highlighting the importance of incorporating feature selection and parameter optimization techniques. These methods can further refine machine learning models, ultimately leading to even more accurate short-term load forecasts. The field is undergoing a significant leap forward with the integration of deep learning and metaheuristic algorithms. Islam et al. (2022) provide compelling evidence for this trend. Their research showcases how integrating these techniques into smart grid load forecasting leads to substantial improvements in both prediction accuracy and operational efficiency. While these studies showcase advancements in general load forecasting, further research is necessary to explore their specific application to microgrids. Microgrids present unique challenges due to factors like intermittent renewable energy sources and varying load profiles. Optimizing forecasting models for microgrids may require incorporating weather data to account for the impact of weather fluctuations, adapting existing methods to handle the dynamic nature of microgrid loads, and evaluating the performance of different techniques within the microgrid context. By addressing these considerations, researchers can leverage the advancements in load forecasting to enhance the efficiency and reliability of microgrid operations.

2.0 Conceptual Framework

2.1 Variables Impacting Electricity Demand Prediction

Microgrid load forecasting is complex due to its dependence on weather. Key factors influencing demand include temperature (affecting heating/cooling needs), humidity, precipitation (impacting human behavior and renewable generation), wind (affecting traditional and renewable energy), storms (causing disruptions), solar radiation (critical for solar-integrated microgrids), and even unprecedented weather events

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

2.2 L2-Regularized Linear Regression Model

According to Rubin*et al.* 2022, wang et al. (2020) and Wahid et al. (2017)Ridge Regression addresses a common issue in forecasting models: multicollinearity, where variables are highly performs well on unseen data.Ridge Regression uses a special function to determine how well it's fitting the data. This function, called the cost function, is shown in equation 1.

$$J(\theta) = RSS(\theta) + \alpha \times \Sigma\theta^2 \quad (1)$$

Unlike Ordinary Least Squares (OLS), which minimizes the residual sum of squares (RSS), Ridge Regression introduces a regularization term. This penalty term, defined in Equation 1 as the cost function (J(θ)), discourages overly large coefficient values (θ) in the model. The hyperparameter (α) controls the strength of this regularization: Increasing α strengthens regularization.

Unlike some techniques, Ridge Regression does not discard any features. Instead, it shrinks the coefficients of less important features towards zero. This reduces their influence on the model and makes it less susceptible to overfitting from noisy data.

2.2 ARIMA Model

Several studies (Hyndman & Athanasopoulos, 2018; Liu., 2015; Omar et al., 2018; Li et al., 2022; Li, 2017) highlight ARIMA (AutoRegressive Integrated Moving Average) as a powerful tool for forecasting future values based on past trends. This method ingeniously combines three key ideas:

i. Remembering the Past (AutoRegressive): ARIMA assumes that past values of the data influence future values. It considers a set number of previous observations (like past sales figures) to predict the next value.
ii. Accounting for Trends (Integrated): Sometimes data has trends or seasonality that can skew predictions. ARIMA addresses this by differencing the data, essentially removing the trend and making it more predictable.
iii. ARIMA incorporates past forecast errors (the residuals between predicted and actual values) to achieve adaptive learning. This allows the model to refine its predictions over time.

ARIMA is defined by a three-parameter model (p, d, q) that captures autoregressive, differencing, and moving average components:

i. Looking Back (p): This determines how many past values are considered for the prediction.
ii. Removing Trends (d): This indicates how many times the data needs to be differenced to become stable.
iii. Learning from Errors (q): This specifies how many past forecast errors are included to refine the model.

The combination of these parameters is written as ARIMA(p, d, q).

While the full mathematical equation can be complex, the core idea is that ARIMA leverages historical data patterns and past mistakes to make informed predictions about the future.

The ARIMA (p, d, q) model incorporates three key components:

- Autoregressive terms (p): These consider the impact of past values on the prediction (Equation 2).
- Differencing (d): This step removes trends and seasonality by differencing the data a specific number of times (d).
- Moving average component (q): This term accounts for the influence of past errors on the prediction (Equation 2).

This combined approach allows ARIMA to capture both past trends and error patterns for effective forecasting.

$$(1-\phi 1L-\phi 2L2-\ldots-\phi pLp)\ (1-L)\ dyt=c+(1+\theta 1L+\theta 2\ L2+\ldots+\theta qLq)\varepsilon t \quad (2)$$

where:

- d: Differencing order to achieve stationarity.
- ϕ1, ϕ2, ..., ϕp: Autoregressive coefficients.
- c (optional): Constant term.
- θ1, θ2, ..., θq: Moving average coefficients.

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

- εt**:** Unpredictable difference at time t.

To understand the ARIMA model's inner workings, we can decompose the ARIMA equation into two main components.

The left side of the ARIMA equation represents the influence of past observations and trends on the predicted value. Here's a breakdown:

- **Autoregressive terms (ϕ):** These terms capture how much the current value (yt) depends on its **recent past values** (up to p lags). They essentially consider the history of the data.
- **Differencing ((1-L) d yt):** This step ensures the data is **stationary** (meaning its statistical properties don't change over time). It removes trends and seasonality by differencing the data d times. The lag operator (L) is used to refer to past values.
- **Right-hand side (moving average):** This side models the influence of past forecast errors (ε) on the current prediction. The moving average terms (θ) consider up to q past errors to refine the forecast.

Determining the optimal values for the ARIMA parameters (ϕ, θ, and c) is vital. Parameter estimation techniques such as maximum likelihood estimation or least squares optimization are employed to analyze the data and determine the optimal values for these parameters. Statistical methods are then used to select the appropriate orders for p, d, and q based on analyses of autocorrelation and partial autocorrelation plots.

2.4 Random Forest Regression Model

Several studies (Awad & Fraihat., 2023; Liu et al., 2022; Nhung & Simioni., 2021; Beaulac & Rosenthal, 2018; Kim et al., 2022; Oshiro et al., 2012) highlight that Random Forest regression doesn't follow a single, unified mathematical model in the traditional sense. This is because it's an ensemble method, which means it combines predictions from multiple models (decision trees) rather than relying on a single equation.

However, we can break down the core concepts involved:

Individual Decision Trees:

- Each tree can be represented by a mathematical model specific to decision trees. This typically involves a series of **if-else** statements based on features (x_i) and thresholds (τ_j).
- At each node (split point) in the tree, a split function is used to determine which branch (left or right) a data point follows. This split function often involves comparing a feature value (x_i) to a threshold (τ_j).

if (x_i>= τ_j) {go to right child node}
else {go to left child node} (3)

- Each leaf node in the tree assigns a final prediction value (y) based on the data points that reach that node (e.g., average of target values in that group).

Ensemble Model:

The final prediction (F(x)) of the random forest is the average (or weighted average) of the predictions from all the individual trees (T) in the forest for a new data point (x):

$$F(x) = \frac{1}{T}\sum(y^t(x)) \; for \; t = 1 \; to \; T \quad (4)$$

$y^t(x)$represents the prediction of tree t for data point x.

Randomness in Training:

- To prevent overfitting and improve generalization, randomness is introduced during tree building:
  - **Feature Subset Selection:** At each split point, a random subset of features (m_{try}) is chosen from the entire set of features (p). The tree considers only these m_{try} features for splitting.
  - **Random Splitting:** When considering a split point within a tree, only a random subset of data points from the current node is used to determine the best split.

Data Normalization.

Preprocessing data is essential for machine learning, and a key step is normalization (Kuhn & Johnson, 2013; James et al., 2021). This ensures all features in a dataset use a similar scale, often ranging from 0 to 1.

This is particularly important because some machine learning models perform better when they don't have to deal with features that have vastly different ranges. Imagine trying to compare a feature measured in feet (e.g., 10) with another measured in inches (e.g., 2). It would be difficult for the model to interpret their relative importance without normalization. Normalization addresses this issue by transforming all features to a common scale, allowing the model to focus on the underlying relationships within the data.

To achieve this common scale, machine learning models often rely on a normalization technique. This is a mathematical process that transforms the data according to the following formula (Equation 6):

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

$x' = \{x - min(x)\} / \{max(x) - min(x)\}(5)$

This formula performs min-max normalization, a technique for scaling data to a common range. Let's break down the components:

- x**:** This represents the original value in your data that you want to normalize.
- *min(x):* This refers to the minimum value within the entire dataset for the specific feature you're normalizing.
- *max(x):* This denotes the maximum value within the entire dataset for the same feature.
- **x':** This represents the normalized value of $'x'$ after applying the formula.

The formula essentially transforms the original *values (x)* to a new range between 0 and 1. Here's how it achieves this:

- The difference between x and the minimum value *(min(x))* is calculated. This effectively removes the minimum value from the original scale.
- This difference is then divided by the total range of the original data *(max(x) - min(x)).* This scales the difference to fall within the new range of 0 to 1.

2.6 Forecasting Performance Indicators

Following established practices (Klimberg et al., 2010; Liu et al., 2020; Tadayonrad & Ndiaye.,2023), we employed various statistical metrics to assess the model's accuracy onnew, unseen data**.** This evaluation was conducted after training the model on historical data (load, temperature, humidity, etc.) from the past six years, which was normalized beforehand.

These evaluators are detailed in the following section.

(i) Root-mean-square error (RMSE)

This metric captures the average magnitude of the difference between predicted (y_i) and actual (xi) values. Lower RMSE indicates better model performance.

The formula for RMSE is:

RMSE = $sqrt\{(1/n) \times Sum\ (y_i - x_i)^2\}(6)$

RMSE is a common metric used to evaluate the difference between predicted values (y_i) and actual values (x_i) in a dataset. Here's a breakdown of the formula:

- n: Represents the total number of data points you're considering.
- y_i: This refers to the predicted value for each data point.
- x_i: This denotes the actual value observed for each data point.
- $\sum(y_i - x_i^2)$: This calculates the squared difference between the predicted values (y_i) and the actual values (x_i) for all data points. Essentially, it sums up the squared errors for each prediction.
- (1/n) : This part takes the average of the squared errors.
- sqrt( ): Finally, the square root of the average squared error is calculated, providing the RMSE value.

(ii)Mean Square Error (MSE)

Mean Squared Error (MSE) is another metric closely related to RMSE. Just like RMSE, MSE calculates the average squared difference between the predicted values (y_i) and the actual values (x_i). However, unlike RMSE, MSE remains in the squared units of the original data.

The formula for MSE is:

MSE = $(1/n) \times Sum\ (y_i - x_i)^2(7)$

Assuming "n" represents the total number of instances, "y_i" is the projected value, and "x_i" is the factualvalue.

(iii) Mean Absolute Percentage Error (MAPE)

This metric expresses the average error as a percentage of the actual values. Lower MAPE indicates better model performance.

The formula for MAPE is:

MAPE = $(1/n) \times Sum\ (|\ (y_i - x_i) / x_i|\ ) \times 100\%(8)$

In the given context, "n" denotes the totalnumber of data points, "y_i" represents the anticipatedvalue, and "x_i"signifies the factual value.

In general, we strive for lower values of RMSE, MSE, and MAPE. These metrics all indicate how close a model's predictions are to the actual observed values. Lower values signify a better fit, meaning the model's predictions are, on average, closer to reality.

3.0 Materials And Methods/Methodology/Experimental Procedure

3.1 Materials

3.1.1Data And Tools For Prediction:

This study utilized three key resources for predicting load demand:

i. Historical Load Data (6 years): Obtained from Jos Electricity Distribution Company (JEDC), this data provides insights into past electricity consumption patterns.

ii. Meteorological Data: Temperature, humidity, and solar radiation data were sourced from the Nigerian Meteorological Agency (NIMET) to understand the influence of weather conditions on electricity demand.

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

iii. Machine Learning Tools: Python libraries - Ridge Regression, ARIMA, and Random Forest Regression - were implemented within Anaconda and Spyder environments to perform the actual demand prediction tasks.

3.2 Method

3.2.1Our Approach: A Step-By-Step Breakdown

The load demand forecasting process involved four key stages for each model (Ridge Regression - RG, ARIMA, and Random Forest Regression - RFR):

i. Data Collection and Pre-processing: In this initial step, we gathered the necessary data (historical load, weather information) and prepared it for modeling. This involved cleaning, and normalization.
ii. Model Development: Here, we built the specific models (RG, ARIMA, RFR) using the pre-processed data. Each model has its own unique training process to learn from the data and establish relationships between variables.
iii. Model Evaluation: After training the models, we assessed their performance using statistical metrics like RMSE, MSE, and MAPE. This evaluation helps us understand how well each model generalizes to unseen data.
iv. Model Selection**:** Based on the evaluation results, we compared the performance of RG, ARIMA, and RFR. The model with the lowest error metrics and the most accurate predictions for our specific scenario was chosen for further analysis.

3.3 Implementation Of Short-Term Load Forecasting Based On RG.

The execution of RGfor Short-term Load Forecasting was done following the step-by-step approach in 3.2.1 using the data and tools in 3.1.1.

3.4ARIMA Short-Term Load Forecasting Implementation

The Steps in 3.2.1and data and tools in 3.1.1were usedfor the simulation of ARIMA.

3.5 Random Forest Regression Short-Term Load Forecasting Implementation

The Steps in 3.2.1 and data and tools in 3.1.1were used for the simulation of Random Forest regression.

Results And Discussion

4.1 Ridge Regression Vs. ARIMA And RFR: A Performance Comparison

This study investigated the effectiveness of three machine learning models for predicting electricity demand 24 hours in advance**.** Ridge Regression, ARIMA, and Random Forest Regression (RFR) were compared using a combination of tables and graphs to assess their performance in this short-term load forecasting task.

4.2 Results For The RG Model

Ridge Regression (RG) was employed in Python to simulate short-term load demand forecasting. Figure 1 illustrate the comparison between actual and predicted load demand values generated by the RG model in line chart.

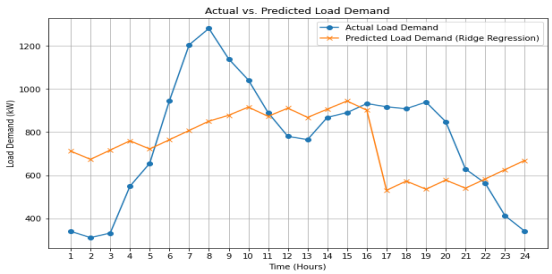

Figure 1 Ridge Regression: Predicted vs. Actual 24-Hour Load

The RG model demonstrated promising performance for 24-hour load demand prediction. It achieved a Mean Squared Error (MSE) of 65748.170, a Root Mean Squared Error (RMSE) of 256.414, and a Mean Absolute Percentage Error (MAPE) of 37.730.

4.3 Results For ARIMA Model.

We leveraged the ARIMA model on Python to simulate the load demand for the next 24 hours. Figure 2 presents this forecast as a line chart

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

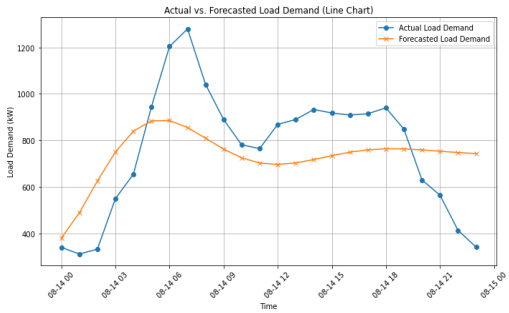

Figure 2 ARIMA Model: Predicted vs. Actual 24-Hour Load

The ARIMA model demonstrated promising results in predicting 24-hour load demand. It achieved a Mean Squared Error (MSE) of 46580.220, a Root Mean Squared Error (RMSE) of 215.820, and a Mean Absolute Percentage Error (MAPE) of 30.210. These metrics indicate that ARIMA's forecasts were reasonably close to the actual values, both in terms of magnitude (RMSE) and percentage (MAPE).

4.4 Results For RFR Model

We used the Random Forest Regression (RFR) model on Python to simulate the load demand for the next 24 hours. This forecast is visualized as a line chart in Figure 3.

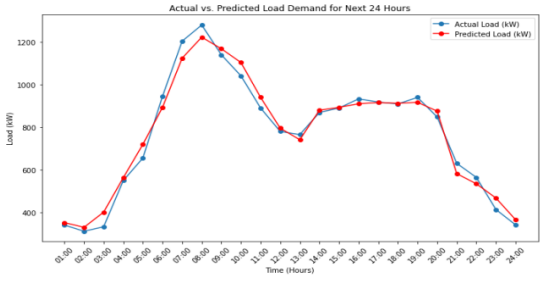

Figure 3 Random Forest Regression (RFR): Predicted vs. Actual 24-Hour Load

The study assessed Random Forest Regression's (RFR) effectiveness in predicting 24-hour load demand through simulation. Notably, the model achieved a Root Mean Squared Error (RMSE) of 140.099, a Mean Squared Error (MSE) of 21608.009, and a Mean Absolute Percentage Error (MAPE) of 25.064. These metrics indicate that RFR's forecasts were highly accurate, with errors on average much lower than both ARIMA and Ridge Regression.

4.5 Analysis Of The Results

Table 1: Comparison of MSE, RMSE, and MAPE for RG, ARIMA, and RFR

Method	RMSE	MSE	MAPE
RG	256.414	65748.170	37.730
ARIMA	215.820	46580.220	30.210
RFR	140.099	21608.009	25.064

This study investigated three machine learning models for predicting 24-hour electricity demand: Ridge Regression (RG), ARIMA, and Random Forest Regression (RFR). RFR emerged as the clear winner, boasting superior accuracy. Compared to RG, RFR delivered a significant 45% reduction in RMSE and a 33% decrease in MAPE, indicating its predictions were much closer to actual values (both in magnitude and percentage terms). Even against ARIMA, which outperformed RG, RFR displayed a notable 35% improvement in RMSE**.** These compelling results highlight RFR as the most accurate model for this specific short-term electricity demand prediction task.

5.0 Conclusion

This study explored the effectiveness of machine learning models for predicting short-term electricity demand. We compared three models: Ridge Regression (RG), ARIMA, and Random Forest Regression (RFR). Our investigation revealed that RFR significantly outperforms both RG and ARIMA in predicting 24-hour electricity demand. Notably, RFR achieved a 45% reduction in Root Mean Squared Error (RMSE) and a 33%

https://dx.doi.org/10.4314/fuoyejet.v9i3.7 engineering.fuoye.edu.ng/journal

decrease in Mean Absolute Percentage Error (MAPE) compared to Ridge Regression. Even compared to ARIMA, which performed better than RG, RFR showed a substantial 35% improvement in RMSE. These findings convincingly demonstrate that Random Forest Regression is the most accurate model for short-term electricity demand prediction in this scenario. Future research could explore how RFR can be integrated into practical applications for optimizing energy management in microgrids.

REFERENCES

Agboola, O.J., Roland, U., Big-Alabi. A., Esoasa, O (2021) Artificial Neural Network for Long-Term Industrial Load Forecast: Trans- Amadi Industrial Layout.Nigeria.18(49):212-221.

Awad, M., & Fraihat, S. (2023). Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. *Journal of Sensor and Actuator Networks, 12*(5), 67. https://doi.org/10.3390/jsan12050067

Beaulac, C., & Rosenthal, J. (2018). Predicting university students' academic success and major using random forests. *Research in Higher Education*. https://doi.org/10.1007/s11162-019-09546-y

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). John Wiley & Sons, Inc.

Butt, F. M., Hussain, L., Mahmood, A., & Lone, K. J. (2020). Artificial intelligence-based accurately load forecasting system to forecast short and medium-term load demands. *Mathematical Biosciences and Engineering*, 18(1), 400-425. https://doi.org/10.3934/mbe.2021022

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media, Inc.

Gopinathan, S., & Annamareddi, S. (2015). Short-term load forecasting using wavelet transform combined with Holt–Winters and weighted nearest neighbor models. International Journal of Electrical Power & Energy Systems, 64, 340–346. https://doi.org/10.1016/j.ijepes.2014.07.043

Hammad, M. A., Jereb, B., Rosi, B., & Dragan, D. (2020). Methods and models for electric load forecasting: A comprehensive review. *Sustainability*, 12(4), 1-28.DOI: https://doi.org/10.2478/jlst-2020-0004

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer Series in Statistics.

Hatziargyriou, N. D., Kleftakis, V., Papadimitriou, C. N., & Messinis, G. (2016). Microgrids in distribution. In *Smart grid handbook*. https://doi.org/10.1002/9781118755471.sgd067

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3

Islam, B. U., Rasheed, M., & Ahmed, S. F. (2022). Review of short-term load forecasting for smart grids using deep neural networks and metaheuristic methods. *Mathematical Problems in Engineering, 2022*, 4049685. https://doi.org/10.1155/2022/4049685

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning with applications in R*. Springer.

Ji, X., Huang, H., Chen, D., Yin, K., Zuo, Y., Chen, Z., & Bai, R. (2023). A hybrid residential short-term load forecasting method using attention mechanism and deep learning. *Buildings, 13*(1), 72. https://doi.org/10.3390/buildings13010072

Kim, J., Yoon, Y., Park, H. J., & Kim, Y. H. (2022). Comparative study of classification algorithms for various DNA microarray data. *Genes*, *13*(3), 494. https://doi.org/10.3390/genes13030494

Kondaiah, V. Y., Balasubramanian, S., Padmanaban, S., & Khan, B. (2022). A review on short-term load forecasting models for micro-grid application. *The Journal of Engineering, 2022*(4). https://doi.org/10.1049/tje2.12151

Klimberg, R. K., Sillup, G. P., Boyle, K. J., & Tavva, V. (2010). Forecasting performance measures - What are their practical meaning? In *Advances in Business and Management Forecasting* (Vol. 7). https://doi.org/10.1108/S1477-4070(2010)0000007012

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 10, No. 1). Springer.

Li, B.-J., Yang, J.-X., Luo, Q.-Y., Wang, W.-C., Zhang, T.-H., Zhong, L., & Sun, G.-L. (2022). A hybrid model of ensemble empirical mode decomposition and sparrow search algorithm-based long short-term memory neural networks for monthly runoff forecasting. *Frontiers in Environmental Science, 10*. https://doi.org/10.3389/fenvs.2022.909682

Li, X., Pan, B., Law, R., & Huang, X. (2017). Forecasting tourism demand with composite search index. Tourism Management, 59, 57–66. DOI: 10.1016/j.tourman.2016.07.005

Liu, D.-R. (2015). A hybrid neural network model for sales forecasting based on ARIMA and search popularity of article titles. *Computational Intelligence and Neuroscience, 2016*(4), 1-9. https://doi.org/10.1155/2016/9656453

Liu, J., Zhou, Z., Kong, S., & Ma, Z. (2022). Application of random forest based on semi-automatic parameter adjustment for optimization of anti-breast cancer drugs. Frontiers in Oncology, 12, 956705. https://doi.org/10.3389/fonc.2022.956705

Omar, H., Hoang, H. V., & Murat, M., Malinowska, I., Gos, M., & Krzyszczak, J. (2018). Forecasting daily meteorological time series using ARIMA and regression models. *International Agrophysics, 32*(2), 253-264. https://doi.org/10.1515/intag-2017-0007

Nhung, D., & Simioni, M. (2021, August). A comparison of random forest and logistic regression model in credit scoring of rural households. In *The 23rd Malaysian Finance Association International Conference 2021 (MFAIC2021)*, Universiti Sains Malaysia.

Niu, W.-J., Feng, Z.-K., Li, S.-S., Wu, H.-J., & Wang, J.-Y. (2021). Short-term electricity load time series prediction by

machine learning model via feature selection and parameter optimization using hybrid cooperation search algorithm. Environmental Research Letters, 16(5), 055032. https://doi.org/10.1088/1748-9326/abeeb1

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How many trees in a random forest? In *IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition*. https://api.semanticscholar.org/CorpusID:12211379

Rubin, J., Mariani, L., Smith, A., & Zee, J. (2022). Ridge regression for functional form identification of continuous predictors of clinical outcomes in glomerular disease. *Glomerular Disease*, *3*(1), 47–55. https://doi.org/10.1159/000528847

Salisu, S., Mustafa, M. W., Mustapha, M., & Mohammed, O. O. (2019). Solar radiation forecasting in Nigeria based on hybrid PSO-ANFIS and WT-ANFIS approach. *International Journal of Electrical and Computer Engineering (IJECE)*, *9*(5), 3916-3926. https://doi.org/10.11591/ijece.v9i5.pp3916-3926

Tadayonrad, Y., & Ndiaye, A. B. (2023). A new key performance indicator model for demand forecasting in inventory management considering supply chain reliability and seasonality. *Supply Chain Analytics, 3*, 100026. https://doi.org/10.1016/j.sca.2023.100026

Wahid, A., Khan, D. M., & Hussain, I. (2017). Robust adaptive lasso method for parameter estimation and variable selection in high-dimensional sparse models. *PLOS ONE, 12*(8), e0183518. https://doi.org/10.1371/journal.pone.0183518

Wang, F., Mukherjee, S., Richardson, S., & Hill, S. M. (2020). High-dimensional regression in practice: An empirical study of finite-sample prediction, variable selection, and ranking. *Statistics and Computing, 30*(3), 697-719. https://doi.org/10.1007/s11222-019-09914-9

Wang, R., Chen, S., & Lu, J. (2020). Electric short-term load forecast integrated method based on time-segment and improved MDSC-BP. Systems Science & Control Engineering, 9(sup1), 80–86. https://doi.org/10.1080/21642583.2020.1843088

Wang, X., Kang, Y., Hyndman, R. J., & Li, F. (2023). Distributed ARIMA models for ultra-long time series. *International Journal of Forecasting, 39*(3), 1163-1184. https://doi.org/10.1016/j.ijforecast.2022.05.001

Wang, W., Ma, B., Guo, X., Chen, Y., & Xu, Y. (2024). A hybrid ARIMA-LSTM model for short-term vehicle speed prediction. Energies, 17(15), 3736. https://doi.org/10.3390/en17153736

Wong, S. Y., & Li, C. (2023). Techno-economic analysis of optimal hybrid renewable energy systems A case study for a campus microgrid. Energy Reports, 9, 134-138. https://doi.org/10.1016/j.egyr.2023.09.153

Yu, Z., Zheng, W., Zeng, K., Zhao, R., Zhang, Y., & Zeng, M. (2024). Energy optimization management of microgrid using improved soft actor-critic algorithm. International Journal of Renewable Energy Development,13(2), 329-339. https://doi.org/10.61435/ijred.2024.59988