Computer Engineering and Applications Vol. 14, No. 1, February 2025

Enhanced Short-Term Residential Load Forecasting Using K-means Clustering and Iterative Residual LSTM Networks

Abdullahi Sulaiman[1*], Ayodele Isqeel Abdullateef[1], Abdulkabir Olatunji Issa[1], Abdulrasheed Olayinka Issa[2]

1Department of Electrical and Electronics, Faculty of Engineering and Technology, University of Ilorin, Kwara, Nigeria.
2Electrical and Computer Engineering, Mississippi State University, Mississippi, US
**absulaiman120@gmail.com*

ABSTRACT

Accurate short-term load forecasting (STLF) is essential for optimizing energy management systems, ensuring operational efficiency, and balancing supply and demand in power grids. This study introduces a hybrid model, K-RNLSTM, which integrates K-means clustering with iterative Residual Long Short-Term Memory (LSTM) networks to improve prediction accuracy. The K-means clustering algorithm categorizes similar load patterns, allowing the model to handle seasonal and hourly variations more effectively. Iterative ResBlocks are incorporated within the LSTM framework to capture complex non-linear dependencies and improve the learning process without suffering from degradation. The model was evaluated using real-world residential electricity consumption data across four seasons: winter, spring, summer, and autumn. The K-RNLSTM model consistently outperformed traditional methods such as Extreme Learning Machines (ELM), Seasonal-Trend Loess (STL), Gated Recurrent Units (GRU), and standard LSTM in terms of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The results demonstrated that K-RNLSTM achieved an average RMSE of 0.71, MAE of 0.43, and MAPE of 1.31%, surpassing benchmark models across all seasonal variations. Furthermore, the integration of ResBlocks significantly improved the model's ability to minimize large forecasting errors, particularly during peak demand periods. This research demonstrates the effectiveness of combining clustering techniques with deep learning models for short-term load forecasting, offering a robust solution for power system operators to optimize energy distribution and reduce operational costs.

Keywords: Load Forecasting, Energy management, LSTM, K-mean, ResBlocks, STLF

1. INTRODUCTION

Efficient electricity load forecasting is paramount for the strategic planning and operational efficiency of power systems. [1]The forecasting horizon, spanning from long-term projections influencing system expansion planning to mid-term and short-term forecasts crucial for real-time operations and electricity market functions in time series, underlines its multifaceted importance. This study zooms in on the intricate task of short-term load forecasting (STLF), specifically targeting the estimation of residential load for the upcoming hour up to a week in advance[2].

Time series forecasting has become increasingly popular among researchers in recent decades, finding applications in diverse fields such as business, economics, engineering, medicine, social sciences, and politics [3].

Various machine learning methods, including autoregressive integrated moving average (ARIMA) [4], Gated Recurrent Unit (GRU), support vector machine (SVM)[4][5], random forest (RF)[6], and Extreme Learning Machine (ELM)[7] methodologies, have been proposed for STLF. In this study, the integration of residual neural network into the predominant use of recurrent neural network (RNN) based models like the Long Short-Term Memory (LSTM) model. This model is known for capturing long-term dependencies in time series data and has been further enhanced in this study by incorporating an Iterative ResBlock proposed in [8]. This synergistic combination aims to leverage both LSTM's temporal pattern recognition capabilities and the iterative refinement provided by the ResBlock in time series thereby improving the overall accuracy of short-term load forecasts.

While prevalent STLF approaches predominantly address the aggregated demand at the system or household level, this study delves into the granular realm of seasonal load forecasting[9].

Advancements in metering technologies have paved the way for detailed load monitoring and forecasting at the consumer level, offering insights into residential user behavior and the untapped potential for demand response strategies [10].

The primary goal of this research is to craft an efficient algorithm for seasonal aggregated load STLF, leveraging the rich data. Harnessing the capabilities of LSTM combined with an Iterative ResBlock, the proposed model integrates past forecast errors. This adaptive learning mechanism enhances the overall prediction performance by leveraging the forecast errors from similar patterns in past data sequences.

This study's contribution spans three dimensions, [1] the introduction of a seasonal aggregated STLF with a k-mean algorithm tailored for residential loads, [2] the utilization of the baseline data to enhance forecasting precision, and the incorporation of a model that dynamically adapts based on past forecast errors, [3] combining LSTM with Iterative ResBlock for improved predictive accuracy. Rigorous testing using real-world load datasets establishes the proposed algorithm's superiority, as evidenced by comparisons with established load forecasting benchmark methods.

1.1 MOTIVATION

The analysis of electricity consumption patterns is crucial for optimizing energy usage and managing costs effectively[1]. In this case study, we focus on the electricity consumption data from Zone 1, which is one of the three designated zones in the dataset. The objective is to employ time series techniques to forecast electricity load accurately[11], allowing customers to proactively manage their usage and avoid exceeding the contracted value of maximum demand.

Figure 1 illustrates the seasonal consumption trends for the year 2017 at Zone 1, depicting variations in electricity usage during winter, spring, summer, and autumn. Notably, the mid-summer period exhibits a peak consumption that surpasses the

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

contracted peak demand, indicated by the dotted red line. This deviation from the contracted peak demand necessitates a closer examination and the implementation of forecasting methods to enhance load prediction accuracy.

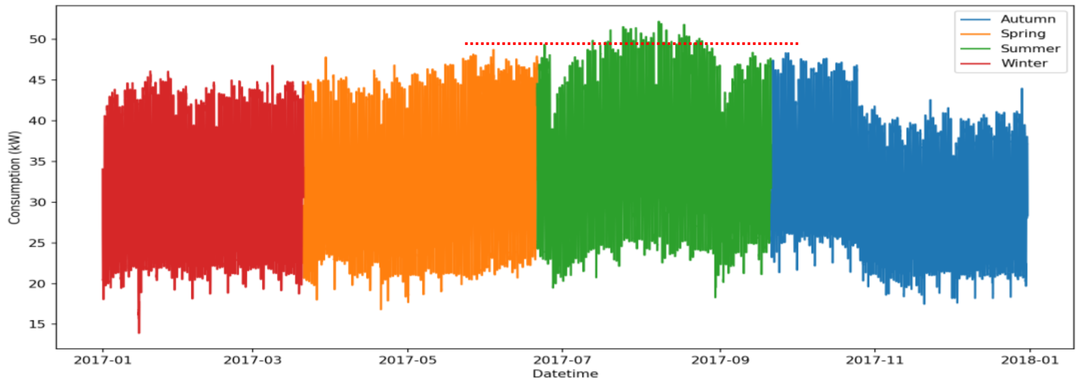

FIGURE 1. Zone 1 seasonal consumption trends

To address this challenge, time series analysis techniques, such as advanced machine learning models like LSTM networks, can be employed. These techniques leverage historical consumption data to identify patterns and trends, enabling the creation of reliable forecasts[12].

The selection of an appropriate time series model depends on the characteristics of the consumption data, such as seasonality, trend, and any potential external factors influencing electricity usage. Additionally, incorporating weather data, economic indicators, or other relevant variables could enhance the forecasting accuracy[13]

The three distinct consumption zones (Zone 1, 2, and 3) provide an opportunity for a granular analysis. By focusing on Zone 1, we can tailor the forecasting models to the specific characteristics of this region, potentially uncovering zone specific patterns and behaviors. This finer granularity allows for more targeted strategies in load management and cost optimization.

The integration of time series forecasting techniques offers a robust approach to predict electricity consumption[14] in Zone 1 accurately. By leveraging historical data and considering the unique characteristics of the region, these models enable customers to make informed decisions, avoiding excessive usage during peak periods and optimizing their electricity costs by the contracted peak demand.

2. METHODOLOGY

2.1 LONG SHORT-TERM MEMORY (LSTM) ARCHITECTURE

Long Short-Term Memory (LSTM) network architecture as shown in Figure 2, is a specialized form of recurrent neural networks (RNNs) designed to mitigate the vanishing gradient problem inherent in traditional RNNs[15]. These networks excel in tasks involving sequential data, making them well-suited for applications like time series forecasting, natural language processing, and speech recognition. The LSTM architecture is distinguished by its use of three primary gates, Forget Gate ( ), Input

Gate ( ), and Output Gate ( ) to regulate the information flow within the network [16] , enabling it to capture and retain long-range dependencies in sequential data.

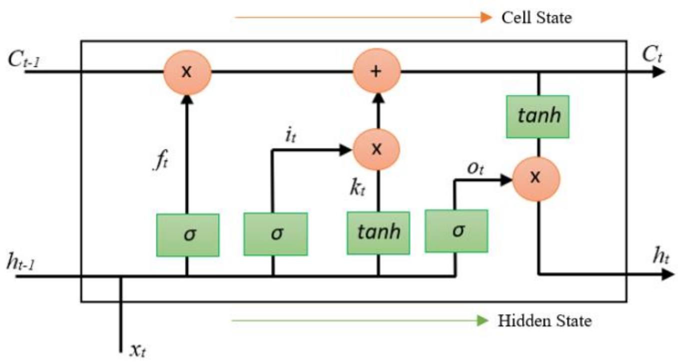

FIGURE 1. LSTM Architecture

The equations governing the LSTM architecture are intricately linked to define the information flow within the network:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \qquad (1)$$

The forget gate f_t decides what information from the previous cell state should be retained or discarded.

W_f is the weight matrix, $[h_{t-1}, x_t]$ is the concatenation of the previous hidden state and the current input, this concatenation combines the historical context stored in the hidden state with the current input, creating a comprehensive representation of the context for the current time step. h_{t-1} represents the hidden state from the previous time step. This hidden state encapsulates information that the network has learned and retained from the past sequence. In the context of sequential data, acts as a memory that helps the network maintain context and capture dependencies across different time steps. x_t denotes the input at the current time step. This input could be any relevant information for the task at hand. In the case of time series forecasting or sequence prediction, it could be the current data point in the sequence.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (2)$$

The input gate updates the cell state by determining which values to update.

$$k_t = \tanh(W_k \cdot [h_{t-1}, x_t] + b_k) \qquad (3)$$

The candidate cell state k_t represents the new values that can be added to the cell state. It is worth noting that the f_t produces a value between 0 and 1 for every entry C_{t-1}, with higher values signifying the significance of retaining the corresponding

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

information in the cell state, and lower values indicating the opposite. Simultaneously, the input gate (i_t) dictates the inclusion of new information in by choosing from the cell candidate k_t [17]. The cell state is updated based on information from the forget gate, input gate, and candidate cell state. Consequently, the cell state maintains a record of past information in the time series and undergoes the following update:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot k_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

The output gate determines the content of the next hidden state while the hidden state is updated based on the cell state and the output gate.

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

Where b_f , and represent bias terms associated with the Forget Gate, Input Gate, Candidate Cell State, and Output Gate, respectively, σ is the sigmoid activation function and tanh is the hyperbolic tangent activation function.

This work trains LSTM networks on historical data to learn temporal dependencies and forecast future power consumption values. The input (x_t) represents the current power consumption, and the output (h_t) could signify the predicted consumption for the next time step. The training process involves adjusting the weights and biases to minimize prediction errors, enhancing the model's ability to capture intricate patterns in power consumption.

Here's a suitable model for incorporating K-Means clustering into your paper on short-term load forecasting for residential electricity consumption:

2.2 K-MEANS CLUSTERING MODEL FOR ELECTRICITY LOAD DATA

The study on the K-Means clustering model for electricity load data aims to enhance short-term load forecasting by categorizing electricity consumption patterns into distinct clusters. This approach ensures improved accuracy and reliability of predictions by capturing seasonal and hourly variations in electricity consumption. Similar research has been conducted with various approaches and methodologies. For instance, [18] compares K-Means and hierarchical clustering for power consumption analysis, indicating that hierarchical clustering is more credible for reflecting both heating and refrigeration demands. Another study by [19] uses the K-Means algorithm to predict electricity demands in smart cities, achieving significant accuracy improvements by integrating various data sources like meteorological and demographic information. Additionally, the adaptive K-Means algorithm is employed to enhance load prediction accuracy in large datasets by incorporating PCA dimension reduction and advanced clustering methods [20]. These studies collectively highlight the versatility and effectiveness of K-Means clustering in analyzing and predicting electricity load patterns, each contributing unique insights into optimizing power consumption forecasting models.

This study employs K-Means clustering to categorize electricity load data into distinct patterns, facilitating improved short-term load forecasting. By clustering similar load behaviors, our approach enhances the accuracy and reliability of predictions, catering to diverse consumption profiles.

2.3 STAGES OF THE K-MEANS CLUSTERING METHOD

2.3.1 CATEGORIZING LOAD DATA

We categorize the load data into four subsets, each corresponding to a distinct season (Winter, Spring, Summer, and Autumn) throughout the year. Further granularity is achieved by dividing the load data for each season into 24 groups, corresponding to the 24 hours of the day. This approach ensures that the model captures both seasonal and hourly variations in electricity consumption.

2.3.2 DETERMINING NUMBER OF CLUSTERS (K)

The optimal number of clusters (K) is determined using the Elbow method. This involves plotting the within-cluster sum of squares against the number of clusters and identifying the point where the decrease in the sum of squares becomes less significant. For our electricity load data, we identified three clusters within each seasonal-hourly subset, representing low, medium, and high consumption patterns.

2.3.3 INITIALIZING CLUSTER CENTROIDS

Cluster centroids are initially assigned randomly within each seasonal-hourly subset. This random initialization ensures that the clustering process starts with diverse centroid positions, preventing the algorithm from converging prematurely to suboptimal solutions.

2.3.4 CALCULATING DISTANCES TO CENTROIDS

The Euclidean distance metric equation 7 is used to calculate the distance between each data point and the centroids. This distance measurement helps in assigning each data point to the closest centroid, ensuring accurate grouping based on load patterns.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ij} - c_{jk})^2} \tag{7}$$

Where d_{ij} is the distance of data point *i* to the center of cluster *j*, x_{ij} represents the *k-th* attribute value of data point i and c_{jk} denotes the *k-th* attribute value of centroid *j*.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

2.3.5 ASSIGNING DATA POINTS TO CLUSTERS

Each data point is assigned to the cluster with the nearest centroid, based on the calculated distances. This step groups similar consumption patterns together, forming clusters that represent different electricity usage profiles within each seasonal-hourly subset.

2.3.6 UPDATING CLUSTER CENTROIDS

After initial clustering, the centroids are recalculated by equation (8) averaging the data points within each cluster. This new centroid position is computed as follows:

$$c_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i \tag{8}$$

Where c_j is the new centroid for cluster *j*, n_j is the number of data points in cluster *j*, and x_i represents the vector of attribute values for data point *i* in cluster j.

2.3.7 ITERATION AND CONVERGENCE

The clustering process is iterated, recalculating centroids and reassigning data points until the centroids stabilize and do not change significantly. This iterative refinement ensures that the clusters accurately represent the underlying patterns in the data.

2.4. DATA PREPROCESSING

Typically, the data obtained may not be directly suitable for input into a forecasting model [21]. Data preprocessing for LSTM models involves addressing imperfections and irregularities in the raw data to create a cleaner and more reliable dataset. One crucial aspect is handling missing values, which can adversely affect the LSTM model's performance. In time-series, data that often involves sequential observations, therefore it is crucial to deal with duplicate timestamps. Duplicate timestamps can introduce redundancy and inaccuracies, potentially leading to biased model training and evaluation. When preprocessing data for LSTM models, it is essential to address these duplicates systematically. The data is first sorted based on timestamps and then checking for and eliminating any repeated timestamps while retaining the associated information. This ensures that each timestamp in the dataset corresponds to a unique set of observations, preventing any distortion in the temporal relationships the LSTM model aims to capture.

Outliers, another facet of data cleaning, are data points significantly deviating from the overall pattern and can distort the model's understanding. Identifying outliers through statistical methods and by removing them helps ensure that the LSTM model is trained on a more robust and representative dataset. By meticulously addressing missing values and outliers, data cleaning contributes to the creation of a refined dataset, setting the foundation for effective LSTM model training and subsequent analyses.

2.5 FEATURE EXTRACTION

With the pre-processed power consumption data at our disposal, the next step in developing an LSTM forecasting model involves setting up the input features and target variables. In time-series forecasting, such as power consumption prediction, the temporal nature of the data is crucial. The target variable at a specific time, denoted as P_t, represents the power consumption that we aim to predict. Simultaneously, the input feature vector g_t is crafted by aggregating power consumption data from multiple preceding timestamps. Specifically, the feature vector is defined in equation (9).

$$g_t = [P_{t-\tau}, ..., P_{t-2}, P_{t-1}] \tag{9}$$

where τ signifies a chosen time interval. This formulation allows the LSTM model to leverage historical patterns in power consumption to make accurate predictions. The feature vector x_t serves as the contextual input to the LSTM model, encapsulating information from the past τ timestamps. This strategic selection of input features enables the model to discern and understand temporal dependencies in the data, a crucial aspect for effective forecasting. By incorporating past power consumption values into the input vector, the LSTM model gains the ability to capture and learn from sequential patterns, making it adept at predicting future power consumption trends. This temporal context provided by the feature vector enhances the model's capacity to generalize and make informed predictions based on historical load patterns.

2.6 DATASETS

In this section, we present an overview of the Kaggle dataset along with the preprocessing procedures undertaken. The Kaggle dataset, publicly available, documents power consumption data within a residential area from January 2017 to January 2018. Our experiments utilize data recorded at a frequency of every 10 seconds for each entry. The preprocessing of the Kaggle dataset encompasses essential steps, including data cleaning, integration, and transformation. The data cleaning adheres to the methodology detailed in Section 2.2. Following the cleaning process, we categorize the load data into four subsets, each corresponding to a distinct season (Winter, Spring, Summer, and Autumn) throughout the year. Further granularity is achieved by dividing the load data for each season into 24 groups, corresponding to the 24 hours of the day.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

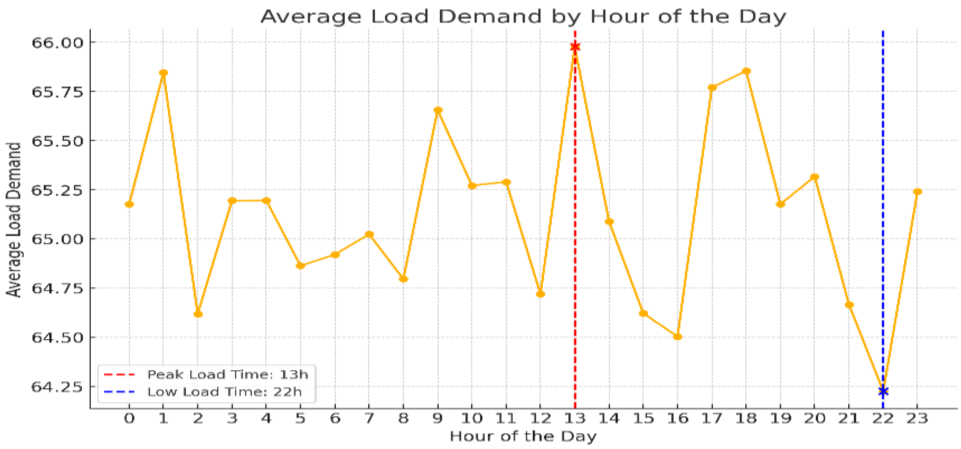

FIGURE 2. Average Load Demand by Hour of the Day

Figure 3 shows the hourly load demand of the day, the peak load demand typically occurs around 1pm, indicating higher energy consumption during early afternoon hours. Conversely, the lowest load demand tends to occur around 10 pm, reflecting reduced energy usage as people wind down for the night. Understanding these patterns can help in optimizing energy distribution and forecast future load demands effectively. Equation (10) represent these temporal and spatial dynamics.

$$L_{s,h,t} = \beta_0 + \beta_1 \cdot E_{s,h,t} + \beta_2 \cdot C_{s,h,t} + \epsilon_{s,h,t} \tag{10}$$

where $L_{s,h,t}$ represents the load at a specific time t for a given season s and hour h, denotes external factors influencing the load at the same time, season, and hour, $C_{s,h,t}$ represents contextual variables capturing the spatial correlations and nuances for the given time, season, and hour, β_0, β_1, β_2 are coefficients that quantify the impact of external factors and contextual variables on the load and $\epsilon_{s,h,t}$ is the error term, accounting for unobserved factors and measurement noise.

3. PROPOSED MODEL (K-RNLSTM)

In this section, we present the K-RNLSTM model, a key component of our forecasting framework. The K-RNLSTM model is an advanced approach for short-term load forecasting that combines the strengths of K-means clustering and iterative ResBlocks, as proposed in [8], within an LSTM framework. This hybrid model aims to capture both spatial and temporal correlations in electricity consumption data, enhancing the prediction accuracy of power consumption patterns. The following sections provide a detailed discussion of the model structure, learning capabilities, iterative procedures, and advantages over traditional methods.

The K-RNLSTM model consists of two main components:

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

1. **K-means Clustering**: This step partitions the data into clusters based on similar electricity consumption patterns. Each cluster represents a specific group of data points that share common features. The clustering is performed for each hour of the day across different seasons to capture seasonal and hourly variations in power consumption. The k-mean groups similar load patterns together, reducing the complexity and variability within each cluster. By doing so, the model can learn more effectively from data that exhibits similar characteristics.

2. **RNLSTM Model**: This model integrates iterative Residual Blocks (ResBlocks) within an LSTM framework. The RNLSTM model is designed to capture the complex nonlinear relationships between input features (e.g., historical electricity consumption, weather data) and the output values (future electricity consumption).

The K-RNLSTM model is designed to learn both deep and shallow features from the input data, leveraging the following capabilities:

1. **Deep Learning with ResBlocks:** The iterative ResBlocks allow the model to learn deep features without the typical degradation issues associated with deep networks. The skip connections in ResBlocks ensure that the model can handle different input and output dimensions, maintaining the benefits of deep learning while preventing performance degradation.
2. **Temporal Learning with LSTM**: The LSTM component of the RNLSTM model captures temporal dependencies in the data, making it well-suited for time series forecasting tasks. LSTM networks are known for their ability to learn long-term dependencies, which is crucial for accurate load forecasting.

The primary goal of the LSTM model is to learn the nonlinear relationships between input features and output values. Although increasing model depth typically enhances learning capacity, deep learning models often face performance degradation due to the intrinsic characteristics of the data or optimization difficulties. Due to this, we propose a k-mean clustering and LSTM-based model with iterative ResBlocks specifically for Short-Term Load Forecasting (STLF). ResBlocks, similar to the building blocks in ResNet used for image classification, feature distinct characteristics. Unlike ResNet blocks, where input and output dimensions are typically the same, in ResBlocks, the skip connection can handle different dimensions.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

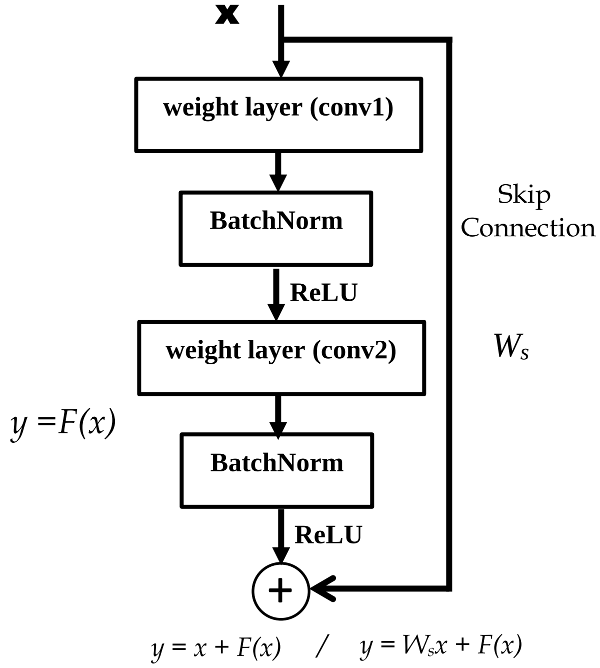

FIGURE 3. Basic ResBlocks Structure

Each ResBlock includes stacked layers and a skip connection that performs an identical mapping when dimensions match or acts as a linear projection when dimensions differ. Figure 4 shows the structure of a ResBlock, which consists of two stacked layers and a skip connection. When the input and the output of the skip connections have the same dimension, the skip connection is a typical identical mapping. Therefore, the output of the corresponding ResBlock is $y = x + F(x)$. When the input and output of the skip connection are in different dimensions, the skip connection performs as a linear projection to match the changes of dimensions. The output of the corresponding ResBlock is $y = W_s + F(x)$, where W is a linear projection. When the stacked layers and the ResBlocks contain the same number of hidden layers, the skip connection ensures that the learning ability of the ResBlock is no worse than the learning ability of the stacked layers.

The RNLSTM model integrates stacked layers and iterative ResBlocks. As illustrated in Figure 5, the structure leverages insights from the iterative ResBlock architecture. The number of iterations is denoted as 't'. When 't = 0,' the ResBlock module is non-iterative, making the RNLSTM equivalent to a standard LSTM model. As 't' increases, ResBlocks are added iteratively, each containing stacked layers, subsequent ResBlocks, and a skip connection.

For each ResBlock in the RNLSTM model, the input is directly connected to the output through a skip connection. This ensures that the learning capability of the current ResBlock with embedded ResBlocks is not inferior to a structure without embedded ResBlocks, thereby enhancing the model's ability to leverage spatio-temporal correlations effectively.

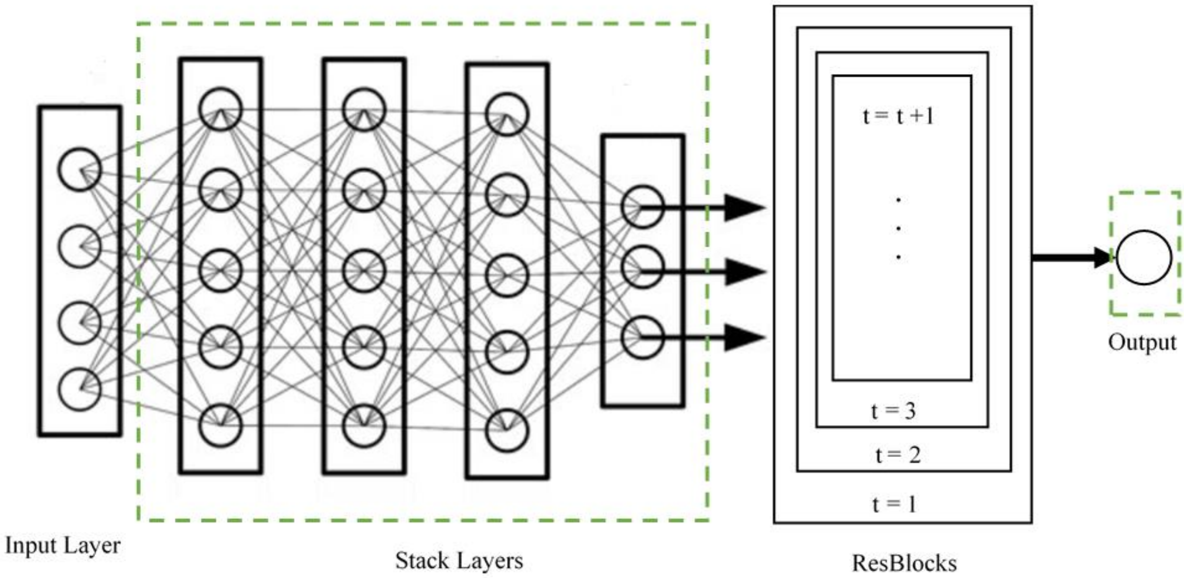

FIGURE 4. Structure of RNLSTM

Equation (11) express iterative procedures of the RNLSTM model for 't ≥ 1'

$$\begin{aligned} y &= F(x_1, g_1) + W_{s0}(x_0) \\ F(x_1, g_1) &= F(x_1, g_1) + W_{s1}(x_1) \\ &\vdots \\ F(x_t, g_t) &= F(x_{t+1}, g_{t+1}) + W_{st}(x_t) \end{aligned} \tag{11}$$

where, $F(x_t, g_t)$ denotes the output of the stacked layers in ResBlock t+1 with input x_t, t+1 is the number of iterations, y is the final output of the RNLSTM model, g represents the weights and biases associated with the model, and W_{st} denotes the linear projection used to match potential dimension changes.

The RNLSTM model, with its innovative use of iterative ResBlocks within an LSTM framework, provides a robust solution for capturing and forecasting the complex spatio-temporal patterns in residential electricity consumption. This approach not only improves forecasting accuracy but also ensures that the deep learning capabilities of the model are effectively utilized.

4. PERFORMANCE METRICS

The performance of the proposed model is evaluated using three well-established error metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as defined in equation (12), (13) and (14) respectively. These metrics are critical in assessing the accuracy of the model in forecasting residential electricity load, particularly in capturing the inherent variability across different time periods and consumption patterns.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i)^2} \quad (12)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}| y_i - y_i | \quad (13)$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{y_i - y_i}{y_i}| \quad (14)$$

where n is the number of observations, $\hat{y}_i$ is the predicted value at time iii, and y_i is the actual value. RMSE is highly sensitive to larger errors due to the squared differences in the formula, which penalizes significant deviations between predicted and actual values. In the context of this work, where accurate load forecasting is essential for effective energy management, RMSE ensures that large prediction errors particularly during peak demand periods are minimized. This sensitivity is critical for maintaining stability in the power system, as substantial forecast deviations can lead to costly inefficiencies or operational challenges.

On the other hand, MAE measures the average magnitude of errors in the forecast, treating all errors equally without emphasizing larger discrepancies. Unlike RMSE, MAE does not square the error terms, making it less sensitive to outliers. This makes MAE particularly useful for evaluating the overall consistency of the model in predicting typical load values. In this study, MAE provides an understanding of how well the model captures everyday variations in residential electricity usage, ensuring it performs well across all seasons without disproportionately emphasizing extreme values.

Lastly, MAPE expresses the forecast error as a percentage of the actual values, offering a normalized error metric that allows for easy comparison across different datasets or scales. In the case of residential electricity load forecasting, MAPE is particularly valuable because consumption patterns can vary widely across seasons and time periods. By providing a relative error measurement, MAPE ensures that the model performs consistently across both high and low consumption periods, allowing for effective comparison of model accuracy regardless of the absolute load values.

Together, these metrics demonstrate the robustness of the K-RNLSTM model in handling short-term load forecasting tasks across varying residential electricity consumption behaviors.

5. RESULT AND DISCUSSION

The performance of the proposed K-RNLSTM model was evaluated using three error metrics: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). These metrics provide a comprehensive assessment of forecasting accuracy across four seasons (Winter, Spring, Summer, and Autumn) and allow for comparison with benchmark models such as ELM, STL, GRU, and LSTM, as shown in Table 1.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

TABLE 1.
K-RNLSTM forecasting result.

Season	Metrics	ELM	STL	GRU	LSTM	K-RNLSTM
Winter	RMSE	1.23	0.93	0.93	0.93	0.89
	MAE	0.86	0.68	0.64	0.83	0.62
	MAPE	2.79	2.29	2.11	2.01	1.50
Spring	RMSE	1.58	0.60	0.74	0.69	0.53
	MAE	1.14	0.43	0.50	0.50	0.36
	MAPE	3.63	1.38	1.56	1.50	1.20
Summer	RMSE	1.20	0.91	0.80	0.82	0.79
	MAE	0.87	0.65	0.50	0.50	0.50
	MAPE	2.58	1.96	1.47	1.44	1.40
Autumn	RMSE	1.00	0.65	0.57	0.51	0.48
	MAE	0.68	0.45	0.37	0.38	0.34
	MAPE	2.23	1.52	1.21	1.21	1.20
Average	RMSE	1.39	0.73	0.65	0.63	0.71
	MAE	0.88	0.51	0.42	0.42	0.43
	MAPE	2.91	1.71	1.36	1.36	1.31

5.1 ACCURACY AND ERROR MINIMIZATION

Across all seasons, the K-RNLSTM model consistently outperformed the traditional models in terms of error minimization. For example, during Winter, K-RNLSTM recorded an RMSE of 0.89, lower than that of ELM (1.23), STL (0.93), GRU (0.93), and LSTM (0.93). The reduction in RMSE is particularly significant because it highlights the model's ability to handle large deviations, which is critical in power systems where substantial errors during peak demand can lead to inefficiencies or increased costs.

Similarly, MAE and MAPE further underscore K-RNLSTM’s superiority. In Winter, K-RNLSTM achieved a MAE of 0.62 and a MAPE of 1.50%, outperforming all other models. These metrics demonstrate that the model not only reduces overall error magnitudes but also consistently improves relative forecast accuracy. MAPE, being a percentage-based measure, is crucial for comparing errors across different seasons with varying load levels. In this case, the low MAPE indicates that the K-RNLSTM model generalizes well, providing highly accurate forecasts regardless of the specific season or load behavior.

5.2 PERFORMANCE ACROSS SEASONS

This trend of superior performance by K-RNLSTM continues across other seasons. During Spring, for example, K-RNLSTM achieved an RMSE of 0.53, compared to 1.58 for ELM, 0.60 for STL, 0.74 for GRU, and 0.69 for LSTM. This reduction in RMSE during Spring reflects the model’s ability to capture seasonal variations effectively. Additionally, K-RNLSTM recorded a MAE of 0.36,

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

significantly lower than that of ELM (1.14) and STL (0.43), which indicates the model's robustness in minimizing the magnitude of errors across typical load patterns.

The model’s average performance across all seasons, as shown in the last row of Table 1, confirms its overall reliability. With an average RMSE of 0.71, MAE of 0.43, and MAPE of 1.31%, the K-RNLSTM model consistently performs better than

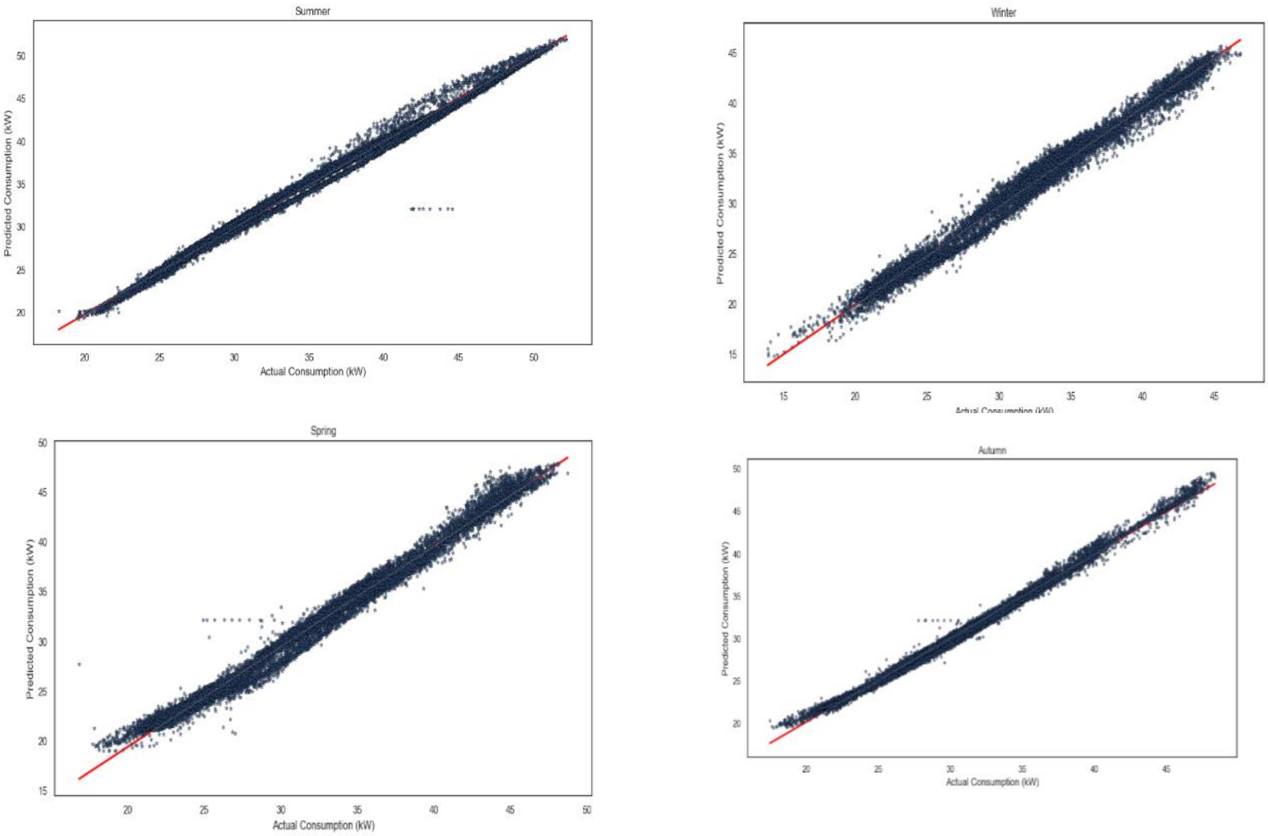

FIGURE 6. Actual and Predicted consumption for each of the season

the other methods, showing that it can accurately forecast short-term load under various seasonal conditions. The GRU and LSTM models, while competitive in certain seasons, do not achieve the same level of accuracy or consistency as K-RNLSTM.

Figure 6 offers a visual comparison of the actual and predicted consumption values for each season, highlighting the accuracy of the K-RNLSTM model. Most data points lie close to the ideal red line, which represents perfect predictions, demonstrating the model's strong ability to match predicted load values with actual consumption. The tight clustering of points around this line indicates the high correlation between predicted and actual values, showcasing the model's predictive accuracy.

However, Figure 6 also reveals a few outliers, where the predicted values deviate from actual values. These deviations suggest that while the model performs well in general, there are instances, particularly during extreme consumption events, where its accuracy could be improved. Despite these few outliers, the overall trend is linear, demonstrating that the K-RNLSTM model captures the general load patterns effectively across all seasons.

5.3 ERROR SENSITIVITY AND IMPLICATIONS

The superior performance of K-RNLSTM can be attributed to its design, which integrates K-means clustering and iterative ResBlocks within an LSTM framework. The K-means clustering step groups similar consumption patterns, which reduces variability and simplifies the learning process. The ResBlocks enhance the model’s depth by allowing it to learn complex patterns without suffering from performance degradation commonly seen in deep learning models.

The significance of using RMSE, MAE, and MAPE together lies in their ability to provide a well-rounded view of model performance. RMSE, with its sensitivity to larger errors, ensures that large deviations are penalized more heavily, which is particularly important for preventing major forecasting errors during peak load periods. MAE, on the other hand, offers a straightforward measure of error magnitude without overemphasizing extreme errors, which makes it useful for understanding the model’s performance in everyday load forecasting. Lastly, MAPE provides a relative measure of accuracy, allowing for easier comparison across different datasets or periods with varying scales of consumption.

6. CONCLUSION

This study presents an enhanced short-term residential load forecasting model, K-RNLSTM, which integrates K-means clustering with iterative Residual Long Short-Term Memory (LSTM) networks. By leveraging the clustering of similar load patterns and iterative ResBlocks, the proposed model effectively captures both spatial and temporal correlations in electricity consumption data. The model consistently outperforms traditional methods such as ELM, STL, GRU, and standard LSTM across all seasons in terms of RMSE, MAE, and MAPE. The results demonstrate the K-RNLSTM model's ability to significantly reduce large forecasting errors, particularly during peak demand periods, while maintaining robust performance across different seasons.

This research highlights the importance of combining clustering techniques with deep learning models to improve the accuracy of short-term load forecasting. The proposed model offers a scalable and reliable solution for power system operators, enabling better energy distribution, reduced operational costs, and more efficient demand response strategies. Future work may explore further model refinements or integration with additional external factors such as weather and economic data to enhance forecasting precision.

REFERENCES

[1] J. Priesmann, L. Nolting, C. Kockel, and A. Praktiknjo, “Time series of useful energy consumption patterns for energy system modeling,” *Sci Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1038/s41597-021-00907-w.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

[2] S. Khan, "Short-Term Electricity Load Forecasting Using a New Intelligence-Based Application," *Sustainability (Switzerland)*, vol. 15, no. 16, Aug. 2023, doi: 10.3390/su151612311.
[3] B. Nepal, M. Yamaha, A. Yokoe, and T. Yamaji, "Electricity load forecasting using clustering and ARIMA model for energy management in buildings," *JAPAN ARCHITECTURAL REVIEW*, vol. 3, no. 1, pp. 62–76, Jan. 2020, doi: 10.1002/2475-8876.12135.
[4] S. Chakrabarti, A. Mukherjee, J. University of Engineering & Management, Institute of Electrical and Electronics Engineers. Delhi Section, and Institute of Electrical and Electronics Engineers, *IEMECON 2019 : the 9th Annual Information Technology, Electromechanical Engineering and Microelectronics : [13th-15th March, 2019]*.
[5] W. Ahmad *et al.*, "Towards short term electricity load forecasting using improved support vector machine and extreme learning machine," *Energies (Basel)*, vol. 13, no. 11, Jun. 2020, doi: 10.3390/en13112907.
[6] H. Yiling and H. Shaofeng, "A Short-Term Load Forecasting Model Based on Improved Random Forest Algorithm," in *Proceedings - 2020 7th International Forum on Electrical Engineering and Automation, IFEEA 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 928–931. doi: 10.1109/IFEEA51475.2020.00195.
[7] Y. Kunqiao and J. Jiandong, "Short-term load forecasting based on ELM combined model," in *Proceedings - 2021 International Conference on Computer, Blockchain and Financial Development, CBFD 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1–6. doi: 10.1109/CBFD52659.2021.00008.
[8] 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. 55785–55797, 2020, doi: 10.1109/ACCESS.2020.2981817.
[9] Institute of Electrical and Electronics Engineers, *2017 52nd International Universities Power Engineering Conference (UPEC) : 28-31 Aug. 2017*.
[10] F. Ünal, A. Almalaq, and S. Ekici, "A novel load forecasting approach based on smart meter data using advance preprocessing and hybrid deep learning," *Applied Sciences (Switzerland)*, vol. 11, no. 6, Mar. 2021, doi: 10.3390/app11062742.
[11] K. Olaniyan, B. C. McLellan, S. Ogata, and T. Tezuka, "Estimating residential electricity consumption in Nigeria to support energy transitions," *Sustainability (Switzerland)*, vol. 10, no. 5, May 2018, doi: 10.3390/su10051440.
[12] R. Bareth and A. Yadav, "Day Ahead Load Demand Forecasting based on LSTM Machine Learning Model," in *2024 3rd International Conference on Power, Control and Computing Technologies, ICPC2T 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 404–408. doi: 10.1109/ICPC2T60072.2024.10474902.
[13] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-term electricity load forecasting model based on EMD-GRU with feature selection," *Energies (Basel)*, vol. 12, no. 6, 2019, doi: 10.3390/en12061140.

[14] 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. 55785–55797, 2020, doi: 10.1109/ACCESS.2020.2981817.

[15] K. Zuo, "Integrated Forecasting Models Based on LSTM and TCN for Short-Term Electricity Load Forecasting," in *2023 9th International Conference on Electrical Engineering, Control and Robotics, EECR 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 207–211. doi: 10.1109/EECR56827.2023.10149951.

[16] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep Learning for Time Series Forecasting: A Survey," Feb. 01, 2021, *Mary Ann Liebert Inc.* doi: 10.1089/big.2020.0159.

[17] Y. Zhou, A. S. Nair, D. Ganger, A. Tripathi, C. Baone, and H. Zhu, "Appliance Level Short-term Load Forecasting via Recurrent Neural Network," Nov. 2021, [Online]. Available: http://arxiv.org/abs/2111.11998

[18] Z. Tang *et al.*, "Comparison of K-Means and Hierarchical Clustering Used in Power Consumption Analysis," in *EI2 2022 - 6th IEEE Conference on Energy Internet and Energy System Integration*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1696–1700. doi: 10.1109/EI256261.2022.10117180.

[19] S. Wang, A. Song, and Y. Qian, "Predicting Smart Cities' Electricity Demands Using K-Means Clustering Algorithm in Smart Grid," *Computer Science and Information Systems*, vol. 20, no. 2, pp. 657–678, Apr. 2023, doi: 10.2298/CSIS220807013W.

[20] L. Zhu and B. Liu, "Prediction of User Electricity Consumption based on Adaptive K-Means Algorithm," in *Proceedings - 2022 IEEE 7th International Conference on Smart Cloud, SmartCloud 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 127–132. doi: 10.1109/SmartCloud55982.2022.00026.

[21] C. S. Bojer, "Understanding machine learning-based forecasting methods: A decomposition framework and research opportunities," *Int J Forecast*, vol. 38, no. 4, pp. 1555–1561, Oct. 2022, doi: 10.1016/j.ijforecast.2021.11.003.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)