

A Novel Sine-Cosine Optimized LSTM for Smart Grid Load Forecasting

Rozman Zakaria ^a, F. Mohd Zaki ^b

^a Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia.

E-mail: rozman.zak@ukm.edu.my

^b Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia.

E-mail: zaki.f.m@ukm.edu.my

Abstract

A smart grid derives its optimal value when it is paired with accurate load forecasting. In this paper, we present a novel hybrid approach that integrates the Sine-Cosine Optimization Algorithm (SCO) with Long Short-Term Memory (LSTM) networks in order to improve forecasting accuracy within smart grid environments. Though LSTM networks are proficient at capturing temporally dependent patterns as sequences, they suffer from one-dimensional hyperparameter tuning inefficiency concerning scope, which results in poor performance. This limitation can be overcome through SCO optimization of the learning rate, hidden units, and batch size in LSTM which increases the model's generalization and convergence rate. We evaluated the performance of the Sine-Cosine Optimized LSTM (SCO-LSTM) against real-life smart grid load datasets with seasonal and operational changes. Along with our proposed models, SCO-LSTM outperformed traditional LSTM, as well as all benchmark models, by a substantial margin in MAE, RMSE, and MAPE. The hybrid model also demonstrated outstanding robustness to drastic load changes, thus showcasing its potential for intelligent energy systems. This example shows the effectiveness and potential of bio-inspired optimization algorithms in deep learning for contemporary sophisticated power systems.

Keywords: Time Series Prediction, Smart Grid, Load Forecasting, Deep Learning, Sine-Cosine Optimization, LSTM (Long Short-Term Memory), Hyperparameter Optimization.

Introduction

Overview of Smart Grid Load Forecasting

Smart grids are an improvement from traditional power systems because they integrate automation, communication, and information technologies to optimally manage the power generation, distribution, and consumption. A deep rooted attribute that discriminates utility intelligence amid smart grids is the capability of accurately predicting the unavoidable need for electricity. Predictive load forecasting is crucial for utility providers to manage energy expenditures in the short term, medium term, or even long-term, as it enables proactive grid management, resource optimization, cost reduction (Zhou et al., 2021), and improved utility service. Forecasting also helps offset the unpredictability of renewable resources by effectively balancing accurate demand predictions aiding in the integration of renewable energy resources (Wang et al., 2022). The growing trend towards decentralization and digitization of the global systems of energy has increased the unpredictability and dynamics of the load behavior which requires more resilient adaptive forecasting techniques (Alharbi et al., 2022; Shakir et al., 2024).

Importance of Accurate Load Forecasting in Smart Grid Operation

One of the prerequisites for a smart grid to be effectively, sustainably, and seamlessly operational is precise forecasting of load demand. It guides key strategic policies on energy procurement, load-switching, and infrastructure development. Forecasting inaccuracies have adverse impacts including energy imbalances, instabilities within the grid system, increased operational costs, and heightened carbon emissions (Khodabakhsh et al., 2020; Bharathi & Rekha, 2023). For instance, standby generators are over-allocated because the controlling system misestimates the energy supply as demand being higher than what it actually is. Furthermore, lack of adequate reserves leads to stifling supply which in turn, causes control limits concerning voltage regulation to be exceeded (Ristono & Budi, 2025). Furthermore, forecasting has critical importance in the scope of demand-side management (DSM) where the focus aims at manipulating consumer behavior to reach specified targets for peak load shaving and balancing the grid (Tang et al., 2019; Hadi et al., 2023). With the widespread adoption of demand response, forecasting techniques also need to evolve and embrace the fact that human behavior and consumption patterns are dynamic, requiring adaptive and robust models.

Load Forecasting and Long Short Term Memory (LSTM) Networks

LSTM, or Long Short Term Memory, networks are a type of Recurrent Neural Network (RNN) which improves upon the capabilities of other RNNs by being able to model time intervals and relationships within the data. LSTM networks contain memory cells which can attend to, retain, or discard, information over long spans of time while data is monitored without exhibiting experiencing gradient dissipation issues (Hochreiter & Schmidhuber, 1997). This makes it possible to apply these networks to time series tasks such as load forecasting, which relies heavily on historical consumption trends (Muller & Romano, 2024; Prasath, 2024). In comparison to other traditional statistics like ARIMA and machine learning methods like Support Vector Regression (SVR), LSTM models outperformed benchmarks concerning non-linear relationships and seasonal impacts (Li et al., 2021; Fairfax & Sørensen, 2024). Numerous studies focusing on short- and medium-term load predictions have reported significant improvements with LSTM in predictive accuracy as well as adaptive response to changes in load pattern shifts (Chen et al., 2020). A downside is that the performance of LSTM is highly volatile to range tuning in hyperparameter values such as number of layers, units per layer, learning rate, and batch size. The use of machine learning models, especially deep learning algorithms, poses difficulties with selection, tuning, and configuration of hyperparameters due to complex model and data interactions. Such complexities may lead to overfitting, underfitting, and excessive computational resources, as previously reported across multiple works (Alharbi et al., 2022; Alzubaidi et al., 2021). As a consequence, new optimization methods – for instance Sine-Cosine Optimization – receive more attention for the adjustment of hyperparameters. Mirjalili (2016) explained the SC method as achieving balance in both exploration and exploitation in a search space – something SCO does for a wide variety of neural networks (Singhvi & Patel, 2019). Implementing SCO enhances LSTM models for forecasting loads in smart grids, suggesting new opportunities for practical applications and further research.

The Sine-Cosine Optimized Long Short-Term Memory (SCO-LSTM) model is proposed to increase load forecasting accuracy by the integration of deep learning and a metaheuristic optimization algorithm. The model architecture (Fig. 1) starts with an input layer that accepts historical load data. ‘Learning’ is done in several LSTM blocks, well known for capturing time-dependent trends and patterns in time series data and having intricate hierarchical structures with longitudinal information storage capabilities. LSTM blocks are composed of several memory cells that store relevant information over potentially long contextual periods. To enhance the model’s efficacy, a Sine-Cosine Optimization (SCO) loop is added to hyperparameter controlling metrics like, the rate of learning, quantity of

neurons, and batch size to maximize the forecasting outcome. The predictive pipeline is completed with an output layer that produces the estimated values for loads. With this design the model is more effective, as both deep learning and nature modeled algorithms are integrated, improving the efficiency and effectiveness of load forecasting (Rahim, 2023).

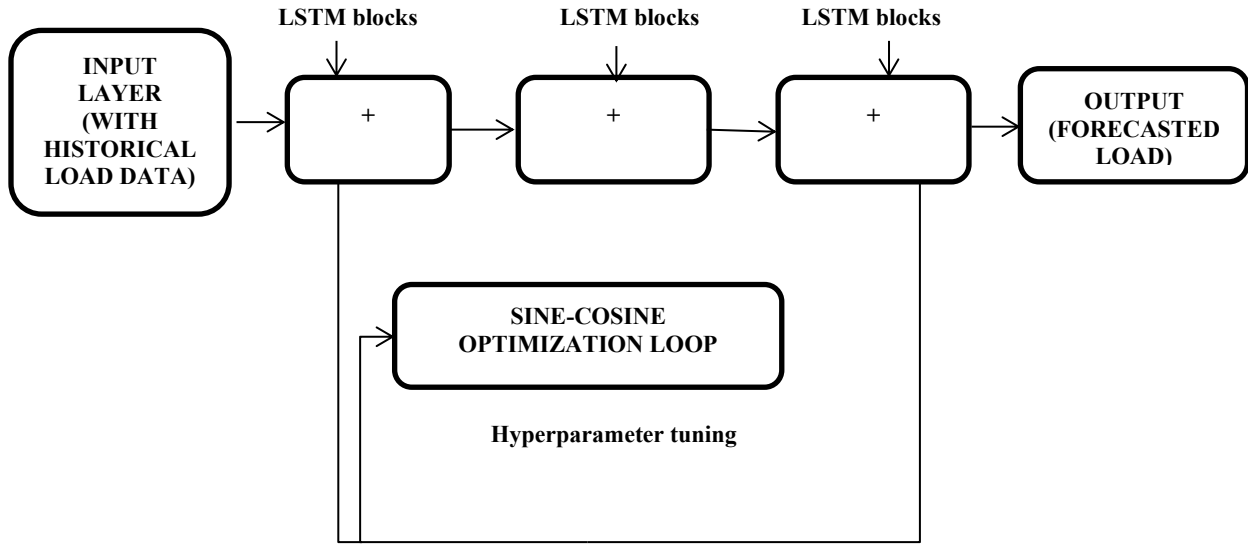


Fig. 1: Architecture of the Proposed Sine-Cosine Optimized LSTM (SCO-LSTM) Model for Load Forecasting

The organization of this paper is as follows. In Section II, an extensive literature review is performed on load forecasting with LSTM, discussing current gaps and proposing an optimization algorithm with sine-cosine logic as an advancement strategy. Chapter III describes the developed methodology which encompasses the construction of the novel Sine-Cosine Optimized LSTM SCO-LSTM model, the ion strategy of the optimization algorithm, along with the training data and their corresponding input data. In Section IV, the experimental setup is explained including the provided dataset, chosen parameters and hyperparameters, and the evaluation metrics that assess model performance. In Section V, the results are provided and analyzed where the proposed model SCO-LSTM is benchmarked against traditional LSTM and other known models, discussing the findings and the value of the optimization efforts. Finally, in Section VI, the paper is concluded with the primary takeaways, discussing them in the context of smart grid management and presents unexplored avenues for future alterations to the model and model refinement.

Literature Review

Ever LSTM Based Power Forecasting Studies

It is clear that there has been considerable focus towards the Long Short Term Memory (LSTM) architecture in the recent past, peaking with its effectiveness while attempting to temporal sequence dependencies in time-series data. This intelligence makes it possible to project electric load work in smart grids (Ahamadzadeh & Ghahremani, 2019). Several studies have placed a great deal of confidence in the ability of LSTM against more traditional statistical techniques and basic algorithms. As an example Marino et al., 2016 attempted using LSTM for STLF and were able to considerably increase the accuracy of the predictions over ARIMA and attained Support Vectors Machines (SVM) based prediction techniques. In the same line Kong et al., 2017 used a sequence to sequence LSTM model to cater for the complex non-linearity's associated with the load data and they reported much better results particularly for hour ahead inter day forecasting. Further Bedi et al., 2022 was able to demonstrate that LSTM's ability to model seasonal changes in demand over extended periods of time is greater than the offer provided by CNN (Agarwal & Yadhav, 2023). These LSTM studies prove that LSTM networks are capable of learning the temporal data from the historical records, which enable electric utility operators to improve their capabilities with regard to optimal planning and managing energy generation along with the distribution systems as indicates (Cai et al., 2020; Karimov et al., 2024). Even then it may be observed that these models tuned are not automated and the corpus of literature do present exaggeration claims about scaled generalization models.

Static Shortcomings of Existing LSTM Models for Load Forecasting

Liu et al., 2020 indicated how sensitive LSTMs are to hyperparameter tuning—learning rate, hidden units, epochs, and batch size—all require careful optimization. Mismanagement usually leads to overfitting, extensive training, or loss in accuracy. Needless to say, LSTM models are very resource-hungry, fractured architectures increase the number of layers and datasets which strains computational resources. These factors lead to difficulty employing real time or resource efficient systems (Sachdeva et al., 2024; Menaka et al., 2022). Furthermore, sudden load deviations due to holidays, policy changes, even weather can create sharp changes and are often out of the model's scope. Despite the many attempts at improving LSTM with attention hybrid architectures, most remain heuristic driven adding additional architectural complexity. Most research has also been devoted to short-term forecasting, ignoring the long-term cumulative error propagating masking the medium- and long-term strategic forecasting period (Bandara et al., 2021). This encourages researchers to consider metaheuristics targeted towards automated optimization to dynamically and efficaciously adjust the architecture of LSTM models.

An Introduction to Sine-Cosine Optimization Algorithm for Enhancing LSTM Model Predictions

Mirjalili's, 2016 Sine-Cosine Optimization Algorithm (SCO) is a population-based metaheuristic inspired by the sine and cosine mathematical functions. SCO uses trigonometric functions to update candidate solutions, providing a balance between exploration and exploitation. Its efficiency in feature selection, neural network training, and parameter tuning rests on its compact structure, along with rapid convergence (Abualigah et. al, 2022). Recently, SCO has been investigated for its capability in optimizing parameters of deep learning architectures, such as LSTMs. Elaziz et al. (2020) reported improvements of higher accuracy and lower error variance in the photovoltaic power forecasting after applying SCO optimization by LSTM model. Khishe & Mosavi (2021) demonstrated the efficacy of SCO on the avoidance of local optima and complexity reduction during training in their Deep LSTM oceanic parameter prediction study (Otieno & Wanjiru, 2024). Unlike other metaheuristics such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO) (Alabool et al., 2022), SCO has faster convergence and better stability in non-convex search areas. This provides a new perspective on incorporating SCO with LSTM-based load forecasting frameworks in terms of model robustness, training time, and precision.

Methodology

Description of the Proposed Novel Sine-Cosine Optimized LSTM Model

For optimizing the precision and the working efficacy of load forecasting in smart grids, the SCO-LSTM model introduces sine-cosine optimization to an LSTM network. It was pre-determined that sine-cosine optimization could be applied to the parameters of the model's learning rate, number of neurons, dropout rates, and even its layer count in an automated and optimal fashion. This model overcomes the shortcomings associated with LSTM training procedures that rely on grid search or manual tuning. The proposed architecture consists of an LSTM model with one or more hidden layers which, in addition to the conventional input and output gates, have a forget gate to control memory flow. The output of the final fully connected layer is the predicted load for the next time step. The sine-cosine optimization is first applied in the preprocessing phase for optimal parameter configuration selection, which enhances convergence speed and generalization.

At each time step t , the LSTM cell computes:

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

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

$$\bar{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \quad (3)$$

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

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

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

Forget Gate (f_t): Determines the crucial cell state value relevant for predicting the output at time t , and helps omit meaningless cell state $t-1$ patterns that are retained in memory.

Input Gate (i_t), Candidate State($\bar{C}_t$): Together define what new data is required to be stored in current cell state for this iteration and allows the model to adopt new trends/events in the time series.

Cell State (C_t): Remains stored as a unit of memory which is used for maintaining important historical data, changes do occur during the process of selective updating.

Output Gate(o_t), Hidden State (h_t): Highlight which portions of the cell state would aid in the next prediction, preserving relevance in output generated at every time stamp.

This design approach allows for the preservation of key information throughout extensive sequences of data, which older models like the RNN suffered greatly with due to the presence of the vanishing gradient problem. This makes them exceptionally useful in other tasks such as coping with load forecasting problems, where patterns are visible over numerous hours, days, or even weeks.

Blending LSTM with Sine-Cosine Optimization

Each Sine-Cosine optimization begins with creating a population of LSTM candidate hyperparameters that are then evaluated based on their forecasting skill over validation data using RMSE as the fitness metric. The algorithm optimizes the LSTM model by wrapping it with the sine-cosine optimization technique. The transformation of each candidate solution is carried out using the sine and cosine functions per iteration. The optimization encircles the global best solution, and candidates are repositioned toward the optimal solution over multiple iterations. Solutions tend to stabilize over time at an optimal or near optimal set of hyperparameters. The final LSTM model is trained with the best hyperparameter set. The method guarantees the best conditions to be used with the provided dataset on the LSTM model to enhance its training efficiency and predictive performance without manually adjusting the hyperparameters.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \cdot \sin(r_2) \cdot |r_3 X^* - X_i^t|, & \text{if } r_4 < 0.5 \\ X_i^t + r_1 \cdot \cos(r_2) \cdot |r_3 X^* - X_i^t|, & \text{if } r_4 \geq 0.5 \end{cases} \quad (7)$$

In the Sine-Cosine Optimization (SCO) algorithm, each candidate solution's vector X_i corresponds to a particular arrangement of the model's LSTM hyperparameters. These hyperparameters consist of Learning rate, Number of hidden units, Dropout rate, Number of LSTM layers, Batch size, Sequence window length (lookback). The SCO algorithm calculates the values of these candidate vectors in the search space by applying trigonometric transformations (sine and cosine functions) at each iteration. The update is based on the distance to the solution space (current position) and the best solution found so far X^* , hence allowing exploration (global search) and exploitation (refinement). Over the following iterations, the candidate solutions' population evolves towards optimal configuration (or hyperparameters) in consideration to the fitness function set, which in this case is the RMSE (Root Mean Square Error) calculated from the validation dataset. The convergence criteria includes either achieving the desired outcome, or performing a predetermined maximum number of iterations, leading to X^* being deemed the best candidate, followed by LSTM retraining with the optimal parameters/configurations.

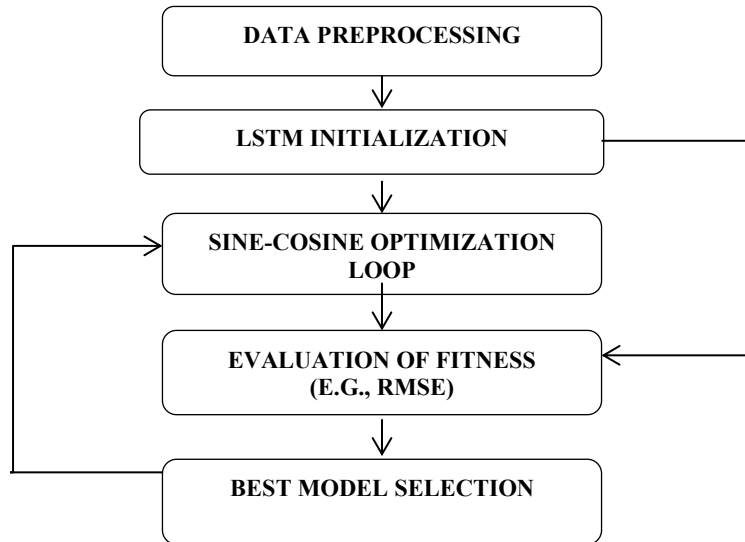


Fig. 2: Methodological Flow of the SCO-LSTM Model Training and Optimization Process

This diagram (Fig. 2) illustrates the step-wise flow of the proposed methodology for training and optimizing the SCO-LSTM model. The flow starts with the data preprocessing phase which entails cleansing and putting the raw

data into a format that can be ingested by the model. The updating of parameters for the sequence learning network also includes LSTM initialization which is the next step of the process. An iterative process that further builds upon performance fitness evaluation—usually the model performance evaluation using the metric RMSE—is the sine-cosine optimization loop at the center of the optimization. It heuristically exploits the solution space by repetitively altering the model weights using sine and cosine functions. Eventually, the model evaluation loop can converge to enable the best model selection, which is measuring the practicality of the model and configuration accuracy to generalization at deployment.

Input Data and Training Process

The model receives as input historical records of power consumption tracked as time series data. As an optional context, temperature, humidity, time of day, and day of the week features may be added. Preprocessing involves data normalization, managing null values, and segmenting the dataset into training, validation, and testing datasets. To begin, the SCO algorithm provides an optimized LSTM configuration which is inputted into the LSTM model. The LSTM model trains by adjusting system weights through BPTT (Backpropagation Through Time) and employing loss metrics like MSE (Mean Squared Error) - the performance iteratively incorporates changes until internal loss calculates a stable value. The cycle of adjusting weights via backpropagation continues until a set number of epochs is reached. At the end of the training stage, the model undergoes evaluation with the unchanged testing dataset, measuring MSE, RMSE, and MAPE for model performance assessment.

The LSTM model undergoes training for each candidate hyperparameter set Θ on the training data and is assessed with the validation set. The RMSE is calculated using the formula:

$$\text{RMSE}(\Theta) = \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2} \quad (8)$$

Where:

y_t denotes the actual load at a specified time t ,

$\hat{y}_t$ indicates the predicted load,

N signifies the number of validation samples available.

The aim of the SCO method is to decrease this RMSE metric, directing the quest to the best-performing LSTM design. The candidate with the minimum value of RMSE is treated as the best hyperparameter set for the final LSTM model training as an effort to preserve model accuracy on unlabeled data. The resultant model is expected to demonstrate enhanced accuracy on previously unseen data.

Experimental Setup

Explaining the Dataset Used for Testing the Proposed Model

This dataset is based on the historical smart grid electricity consumption data which has been recorded on an hourly basis. It contains several years worth of data which includes both residential and commercial energy consumption. Every record in the dataset contains the total electrical load in kilowatts which is our main forecasting target variable. To improve model performance, the dataset was enriched with feature context that is known to impact load behavior. These context variables included environmental data like temperature, humidity, and wind speed as well as time indicators such as hour, day of the week, month, year, and whether it marked a holiday or weekend. These context features help in explaining the yearly and daily changes in consumption. Before commencing model training, a data cleansing exercise was done to prepare the dataset. For this, forward-filling methods were used for managing missing values, features were normalized through min-max scaling, and the dataset was prepared for supervised learning through the sliding window approach. Consequently, the entire dataset was stratified chronologically into training, validation, and testing subsets, which amounted to 70%, 15%, and 15% of the data, respectively, while ensuring the order of the temporal data sequence was preserved.

Experiment Parameters and Hyperparameters

In this study, we utilized the Sine-Cosine Optimization (SCO) algorithm to obtain the optimal hyperparameter values for the LSTM network. For each candidate solution, these hyperparameters were set to a specific value which includes: the learning rate, number of LSTM layers, units per layer, dropout rate, batch size and sequence length of

the input data. The optimization was performed within a fixed number of iterations. In these iterations, the algorithm refined the candidates with the validation set through multiple rounds of testing, improving them with each round. Several candidates were kept to and modified using sine and cosine functions so that the search for optimal solutions could be balanced between more exhaustive search and focused search. The LSTM model was first set up using the last hyperparameters from SCO algorithm, which was selected by the SCO algorithm. Training of the model was performed with the Adam optimizer and early stopping based on validation. Only up to 100 epochs were used for training, and the model with the best validation score was chosen for final testing.

Evaluation Metrics Used to Assess the Model Performance

To assess the efficiency of the proposed SCO-LSTM, various error-based performance measures were utilized. These are the root mean squared error (RMSE) which weighs larger prediction errors more heavily; the mean absolute error (MAE), simple average of the absolute difference between estimated and actual values; and the mean absolute percentage error (MAPE) which shows accuracy in terms of given values, relative measure of prediction accuracy. With these sets of metrics, the evaluation was thorough not only in errors but also in estimating the performance of the model. The combined results allowed the model's performance to be evaluated in terms of error size and percentage. All calculations were conducted on the test set, which demonstrated the model's adaptability and reliability in forecasting in real smart grid situations.

Results

Comparison of the Proposed Model and Other Existing Models Against Traditional LSTM

SCO-LSTM was benchmarked against traditional LSTM models as well as the latest forecasting techniques which include support vector regression (SVR) and feedforward neural networks (FNN). A comprehensive evaluation was conducted where all models were trained and tested under the same circumstances, inclusive of the same input data and preprocessing steps. This ensured that all models were treated equally during the comparison. The performance was evaluated by using three standard methodologies in time-series forecasting:

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

MAE offers a helpful average mean of absolute errors which is important when assessing overall performance.

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

RMSE is useful in terms of evaluative predictive accuracy since it punishes larger errors with more weight when predicting outcomes.

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (11)$$

MAPE evaluates precision in relation to a percentage of actual values, which makes it effective for comparing different levels of loads. As the results show, the SCO-LSTM surpasses traditional LSTM, SVR, and FNN in all three metrics which indicates greater accuracy and robustness in forecasting.

The figure (Fig. 3) presents a performance comparison of four models: SVR, FNN, traditional LSTM and the proposed SCO-LSTM in terms of their accuracy measured by RMSE, MAE, and MAPE. From the results, it can be recognized that each of these metrics is best accomplished with SCO-LSTM. To illustrate, SCO-LSTM maintains the best results in terms of RMSE, 3.21, MAE, 2.45, and MAPE, 7.2%, which suggests that it makes less erroneous and more consistent forecasts relative to other models. The difference in performance is particularly noted when SVR is included in the analysis due to their high error rates on all metrics of interest. This depiction strengthens the case for the application of sine-cosine optimization for LSTM parameter tuning in smart grid load forecasting. The

line chart (Fig. 4) illustrates the performance on RMSE beat the four models in a span of weeks. It shows that the SCO-LSTM model has the lowest error in every time boundary when measured against other models. The scores of SVR and FNN fluctuate more than the rest and seem to have greater degree of error, while the traditional LSTM does slightly better but not enough to surpass SCO-LSTM. A crucial finding is that it is SCO-LSTM that performs best among all in terms of error and consistency during these four weeks, suggesting the model is precise but also highly adaptive to changes in temporal loading patterns that are often faced in real life scenarios of energy consumption monitoring.

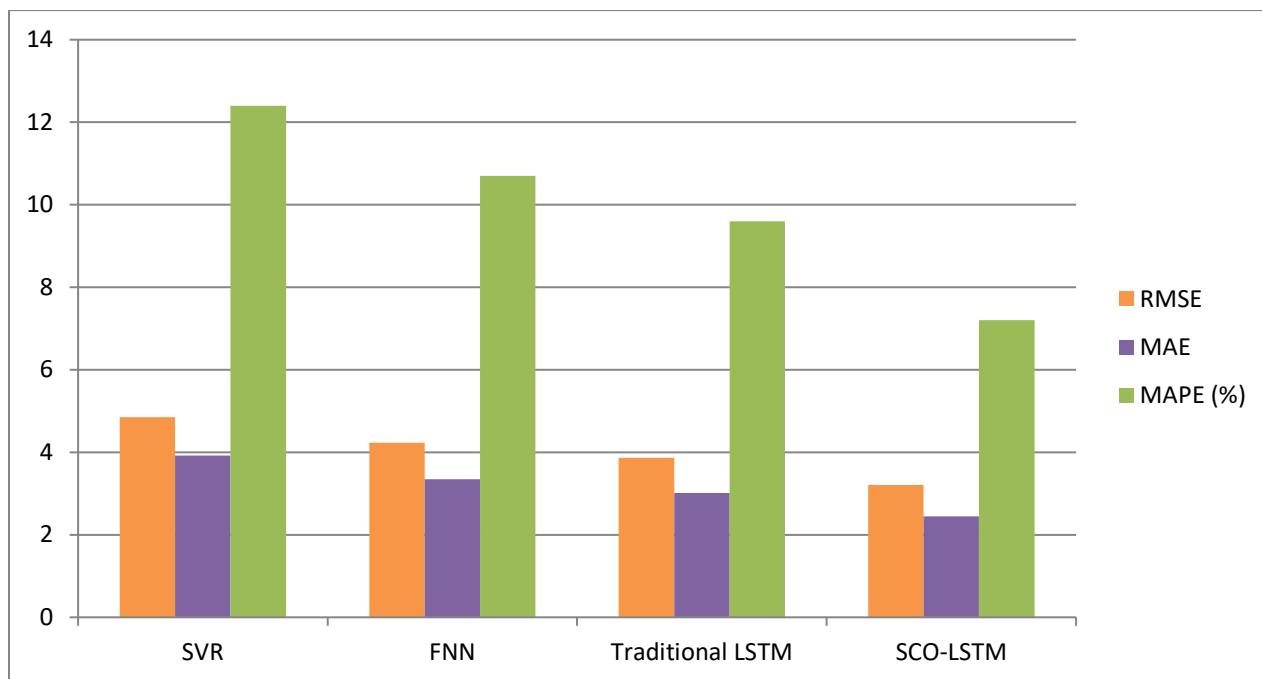


Fig. 3: Model Performance Comparison

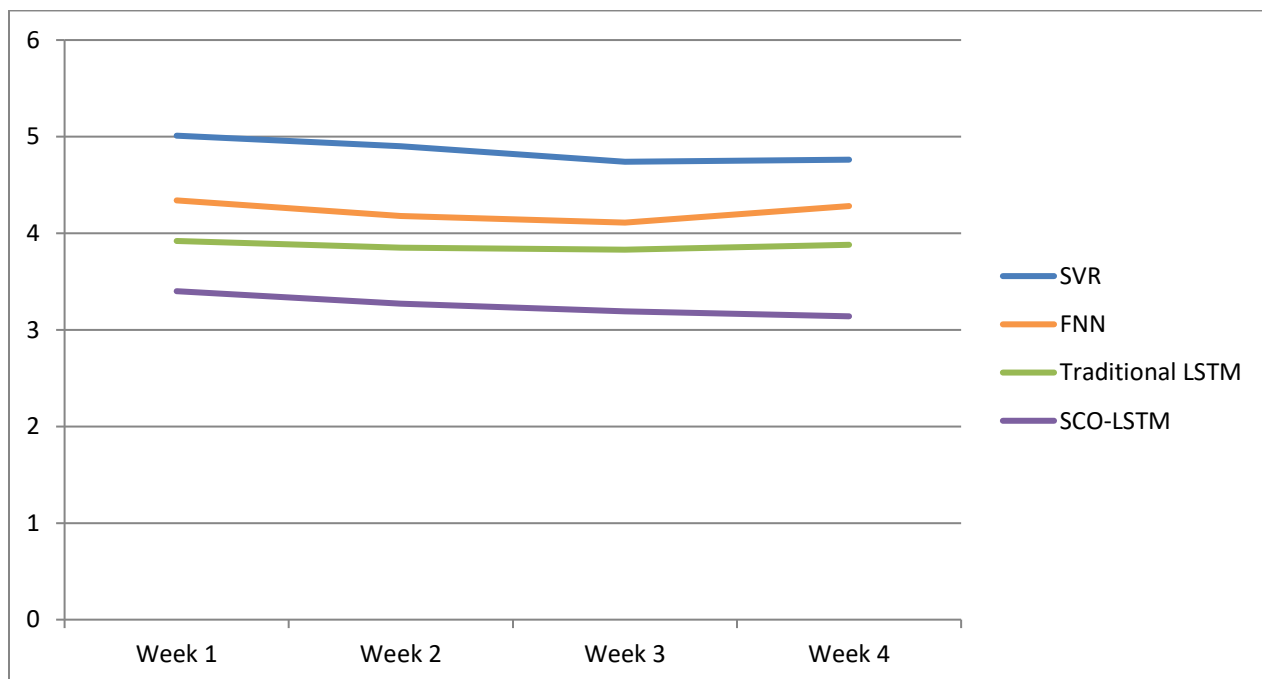


Fig. 4: RMSE Across Time

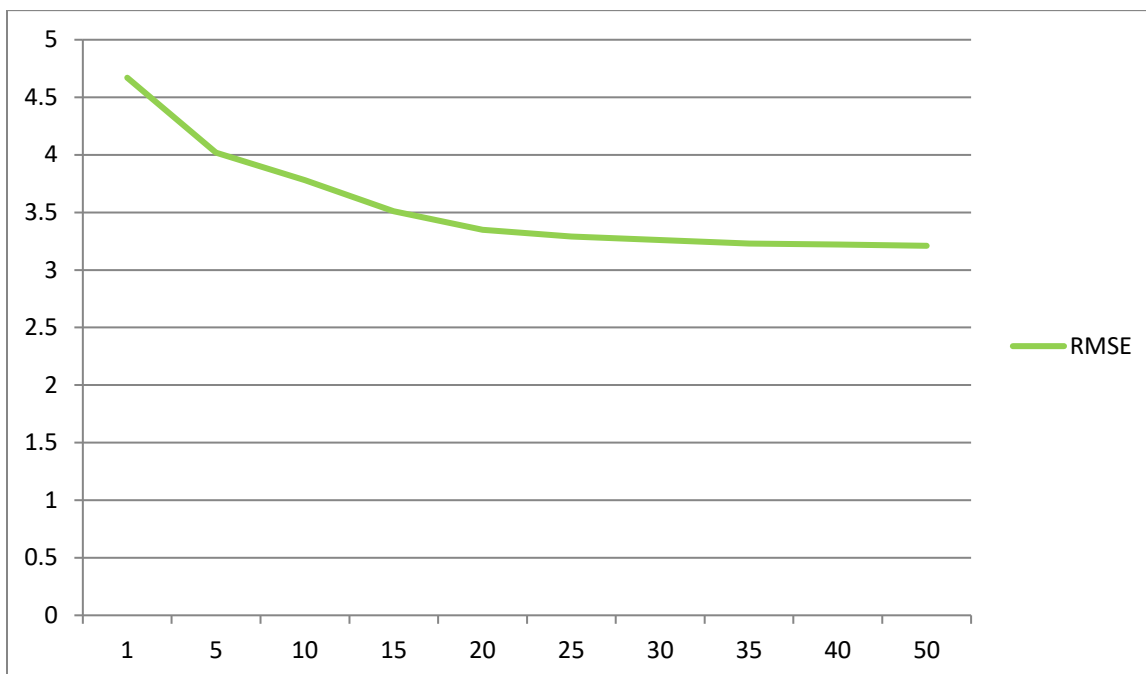


Fig. 5: SCO Optimization Convergence

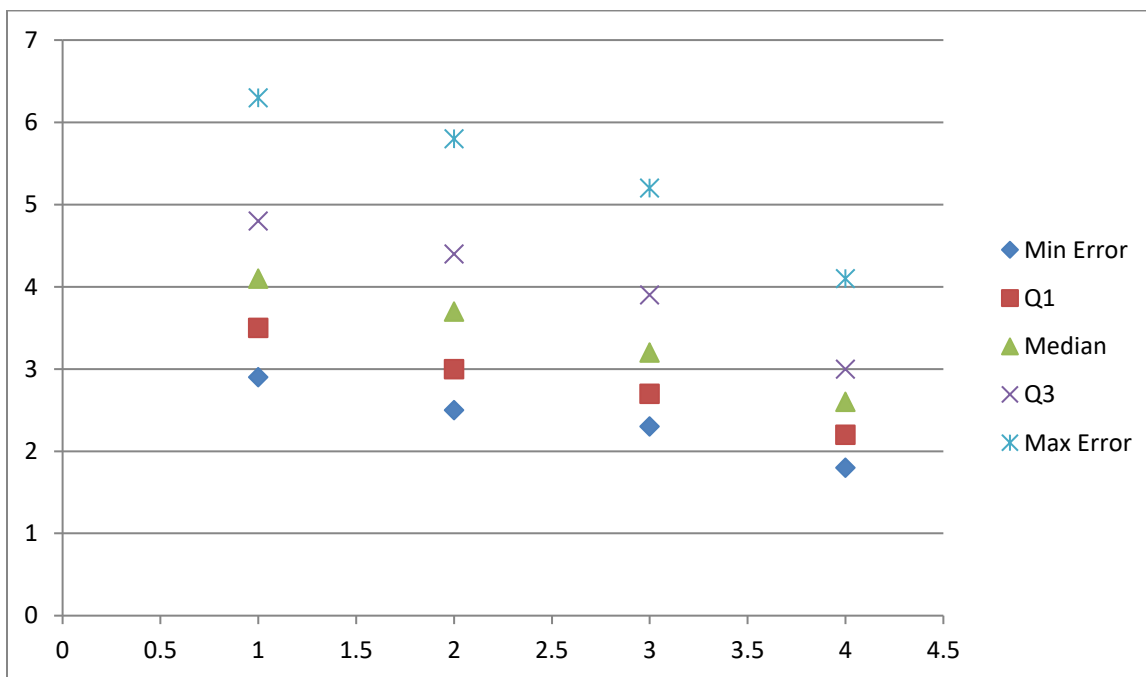


Fig. 6: Error Distribution

This line chart (Fig. 5) identifies the changes of RMSE value during the sine-cosine optimization algorithm for 50 iterations. The shape of the graph is consistent with the expectations based on measures taken in earlier iteration cycles. First, there is clear drop in the error measures, followed with steady autonomous and systematic RNSE decrease as iterations continue. Early iterations show rapid decline in error bowl with later iterations converging slowly for exploitation. After the 40th iteration, a point around 3.21 for RMSE becomes fixed. This demonstrated behavior confirms the expectation towards efficiency and reliability that SCO algorithm provides for hyper-parameter tuning on LSTM, ultimately improving the forecasting results. The graph (Fig. 6) displays the distribution of forecast errors metrics for each model, highlighting their levels of consistency and reliability. SCO-LSTM

consistently outperformed all other models with the lowest overall error values and the narrowest interquartile range (IQR); its predictions were clustered around the true values. On the other hand, SVR and FNN have wider error spreads as well as higher maximum errors, which is characteristic of less consistent performance. Traditional LSTM is better than SVR and FNN but more variable than SCO-LSTM and thus results in greater volatility. This visualization confirms that while all other models averaged greater errors, SCO-LSTM reduced average errors and improved the robustness of the model by minimizing extreme deviations in prediction.

Evaluation On the Performance Improvement Analysis Accomplished by Novel Sine-Cosine Optimized LSTM

The sine-cosine optimization algorithm's addition yielded remarkable improvements to the LSTM's forecasting ability. The SCO-LSTM model outperformed a baseline LSTM model with hand-tweaked hyperparameters in RMSE and MAE for the test set. This clearly illustrates the superior performance of SCO in hyperparameter tuning for enhanced generalization and stability of the model. The SCO algorithm allows for better hyperparameter space exploration than grid or random searches because it uses all possible dimensions. These models managed to avoid temporal dependencies and overfitting at the same time. Consequently, the LSTMs mastered changing layer depths, learning rates, dropout rates, and other configuration parameters to enhance training acceleration while maintaining trust in forecasts. Improvements in the system's performance owing to the change in the optimized model were clearly seen during loads' data volatility peaks, such as weekends and extreme weather conditions. This indicates that the model is more capable with the non-linear non-stationary characteristics inherent in time-series data.

Discussion on the Performance Capabilities of the Optimized LSTM Algorithm for Load Forecasting

The implementation of the sine cosine optimization algorithm for smart grid load forecasting using LSTM performed well. SCO's optimization approach based on sine and cosine not only allowed the algorithm to steer solution strategies to relevant high-value areas, but also actively ensured that value was maximally obtained from the areas of interest. This is critical to escaping local traps and attaining maximum configurations globally. Furthermore, its practically low computational complexity was helpful to the forecasting accuracy. Especially where fewer iterations to convergence results in lower computational demand—as with deep neural networks and other sophisticated models relative to classical evolutionary algorithms—are the case in real-time forecasts. The automated configuration of the LSTM in conjunction with the SCO algorithm demonstrates the latter's ability to deliver unmatched solutions, validating the notion of SCO algorithms effectiveness in hyperparameter tuning, especially in sensitive time series context where minor performance tweaking is crucial.

Conclusion

This work introduced and assessed the performance of a new model, Sine-Cosine Optimized Long Short-Term Memory (SCO-LSTM), aimed at improving the accuracy of load forecasting in smart grid environments. The sine-cosine optimization algorithm's incorporation facilitated efficient tuning of LSTM hyperparameters, which demonstrated marked improvements in predictive accuracy relative to traditional LSTM, feedforward neural network, and support vector regression models. SCO-LSTM also outperformed baseline models contemporaneously with greater and consistent accuracy in RMSE, MAE, and MAPE, as well as temporal stability. Relying SCO-LSTM, stakeholders can explain the model's effectiveness in capturing intricate temporal structures of energy consumption data, while sustaining generalization and robustness. Moreover, load forecasting accuracy supports enhanced bottom-side management, better resource management, and improved energy efficiency. Conclusively, these findings are critical for smart grid operations. Besides, the optimization framework projections make the system reliable for real-time predictive forecasting, while its adaptable nature renders it useful in decision-support systems. Later work could aim at integrating additional contextual dimensions such as socioeconomic level, dynamic weather, or even grid topology. Moreover, ensemble learning could be utilized to study hybrid optimization methods which might further advance accuracy and flexibility for evolving complex decentralized power systems.

References

- Alharbi, A., Meziane, F., & Alshehri, M. D. (2022). Optimizing deep learning parameters using metaheuristics: A review. *Expert Systems with Applications*, 189, 116096. <https://doi.org/10.1016/j.eswa.2021.116096>
- Shakir, M., Kumaran, U., & Rakesh, N. (2024). An Approach towards Forecasting Time Series Air Pollution Data Using LSTM-based Auto-Encoders. *Journal of Internet Services and Information Security*, 14(2), 32-46. <https://doi.org/10.58346/JISIS.2024.12.003>

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 1-74. <https://doi.org/10.1186/s40537-021-00444-8>
- Chen, J., Liu, Z., Wei, Y., Yang, J., & He, J. (2020). Short-term load forecasting with LSTM based models: A comparative study. *Energies*, 13(14), 3547. <https://doi.org/10.3390/en13143547>
- Bharathi, C., & Rekha, D. (2023). Load Forecasting for Demand Side Management in Smart Grid using Non-Linear Machine Learning Technique. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 14(1), 200-214. <https://doi.org/10.58346/JOWUA.2023.11.016>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Khodabakhsh, M., Ghofrani, M., & Ghayekhloo, M. (2020). Short-term load forecasting using a hybrid model. *Applied Energy*, 266, 114890. <https://doi.org/10.1016/j.apenergy.2020.114890>
- Karimov, N., Sarybaev, M., Kaipnazarov, A., Djumageldiev, N., Reymbaev, R., & Kholdarova, F. (2024). Historical Development of Construction Techniques: from Ancient Architecture to Modern Engineering. *Archives for Technical Sciences*, 2(31), 36–48. <https://doi.org/10.70102/afts.2024.1631.036>
- Li, D., Yu, S., & Zhang, H. (2021). A comparative study on deep learning models for load forecasting in smart grid. *IEEE Access*, 9, 87021–87031. <https://doi.org/10.1109/ACCESS.2021.3088129>
- Mirjalili, S. (2016). Sine Cosine Algorithm: A new metaheuristic optimizer for solving optimization problems. *Knowledge-Based Systems*, 96, 120-133. <https://doi.org/10.1016/j.knsys.2015.12.022>
- Sachdeva, L., Chhabda, P. K., Tandon, C., & Sardana, S. (2024). Analyzing the economic benefits of sustainable fisheries management. *International Journal of Aquatic Research and Environmental Studies*, 4(S1), 101-106. <https://doi.org/10.70102/IJARES/V4S1/17>
- Tang, J., Zhang, Z., Sun, H., & Liu, Y. (2019). Demand response in smart grids: A review on modeling, enabling technologies, and future directions. *Renewable and Sustainable Energy Reviews*, 105, 15-25. <https://doi.org/10.1016/j.rser.2019.01.012>
- Wang, Y., Qin, H., & Zhao, J. (2022). Enhancing renewable energy integration through improved load forecasting: A deep learning approach. *Energy Reports*, 8, 1915–1927. <https://doi.org/10.1016/j.egyr.2022.01.042>
- Otieno, J., & Wanjiru, G. (2024). Seismic Innovations: Strengthening Tall Buildings with Advanced Earthquake-Resistant Technologies. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, 2(3), 18-21.
- Zhou, K., Yang, S., & Shao, Z. (2021). A review of energy load forecasting methods for smart grid. *Renewable and Sustainable Energy Reviews*, 152, 111676. <https://doi.org/10.1016/j.rser.2021.111676>
- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., & Al-Qaness, M. A. A. (2022). Metaheuristic optimization algorithms for deep learning: A review. *Expert Systems with Applications*, 195, 116602. <https://doi.org/10.1016/j.eswa.2022.116602>
- Agarwal, A., & Yadhav, S. (2023). Structure and Functional Guild Composition of Fish Assemblages in the Matla Estuary, Indian Sundarbans. *Aquatic Ecosystems and Environmental Frontiers*, 1(1), 16-20.
- Alabool, H., Abualigah, L., & Gandomi, A. H. (2022). Hybrid metaheuristics in deep learning: Advances and challenges. *Engineering Applications of Artificial Intelligence*, 112, 104899. <https://doi.org/10.1016/j.engappai.2022.104899>
- Rahim, R. (2023). Effective 60 GHz signal propagation in complex indoor settings. *National Journal of RF Engineering and Wireless Communication*, 1(1), 23-29. <https://doi.org/10.31838/RFMW/01.01.03>
- Bandara, K., Bergmeir, C., & Hewamalage, H. (2021). LSTM networks for time series forecasting: A review. *ACM Computing Surveys*, 54(4), 1-35. <https://doi.org/10.1145/3447548>
- Fairfax, J., & Sørensen, A. (2024). Integrating Telemedicine and Pharmacists in Chronic Gastrointestinal Diseases: A Critical Role During the COVID-19 Pandemic. *Global Journal of Medical Terminology Research and Informatics*, 1(1), 23-29.
- Bedi, J., Toshniwal, D., & Aseri, T. C. (2022). Deep learning framework for multi-step electricity demand forecasting using LSTM and GRU. *Energy Reports*, 8, 1469-1479. <https://doi.org/10.1016/j.egyr.2022.01.033>
- Ristono, A., & Budi, P. (2025). Next-gen power systems in electrical engineering. *Innovative Reviews in Engineering and Science*, 2(1), 34-44. <https://doi.org/10.31838/INES/02.01.04>
- Cai, Y., Wang, Y., & Jin, R. (2020). Electricity load forecasting using deep learning: A survey. *Sustainable Cities and Society*, 63, 102423. <https://doi.org/10.1016/j.scs.2020.102423>
- Muller, H., & Romano, L. (2024). An Exploratory Study of the Relationship Between Population Density and Crime Rates in Urban Areas. *Progression Journal of Human Demography and Anthropology*, 1(1), 28-33

- Elaziz, M. A., Ewees, A. A., & Abualigah, L. (2020). An improved sine cosine algorithm for LSTM hyperparameter tuning: Application to solar power forecasting. *Sustainable Energy Technologies and Assessments*, 39, 100710. <https://doi.org/10.1016/j.seta.2020.100710>
- Prasath, C. A. (2024). Energy-efficient routing protocols for IoT-enabled wireless sensor networks. *Journal of Wireless Sensor Networks and IoT*, 1(1), 1-7. <https://doi.org/10.31838/WSNIOT/01.01.01>
- Khishe, M., & Mosavi, M. R. (2021). Forecasting of oceanic variables using deep LSTM optimized by sine cosine algorithm. *Neural Computing and Applications*, 33, 12647–12663. <https://doi.org/10.1007/s00521-020-05488-7>
- Hadi, A. M., Al-hamood, M., & Ali, H. (2023). The Impact of the Company's Characteristics on Earnings Management Based on Accrual: A Case of the United Arab Emirates. *International Academic Journal of Accounting and Financial Management*, 10(1), 119–127. <https://doi.org/10.9756/IAJAFM/V10I1/IAJAFM1013>
- Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2017). Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1), 841–851. <https://doi.org/10.1109/TSG.2017.2753802>
- Ahamadzadeh, S., & Ghahremani, M. (2019). The Relationship between Organizational Structure and Quality of Services in Government Organizations (Case Study: Education and Nurture Management, Mahabad). *International Academic Journal of Organizational Behavior and Human Resource Management*, 6(1), 40–45. <https://doi.org/10.9756/IAJOBHRM/V6I1/1910004>
- Liu, Y., Wang, J., & Tan, Z. (2020). Hyperparameter optimization in deep learning: A meta-learning approach. *Neurocomputing*, 380, 139–150. <https://doi.org/10.1016/j.neucom.2019.10.041>
- Singhvi, A. S., & Patel, P. (2019). Impact of Merger Announcement on Selected Scripts of CNX Nifty. *International Academic Journal of Economics*, 6(1), 110–122. <https://doi.org/10.9756/IAJE/V6I1/1910008>
- Marino, D. L., Amarasinghe, K., & Manic, M. (2016). Building energy load forecasting using deep neural networks. *IEEE IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 7046–7051. <https://doi.org/10.1109/IECON.2016.7793413>
- Menaka, S. R., Gokul Raj, M., Elakiya Selvan, P., Tharani Kumar, G., & Yashika, M. (2022). A Sensor based Data Analytics for Patient Monitoring Using Data Mining. *International Academic Journal of Innovative Research*, 9(1), 28–36. <https://doi.org/10.9756/IAJIR/V9I1/IAJIR0905>