A Real-Time Charge Predictive Model for Intelligent Networks

Monia Bartouli
Laboratory of Micro-Optoelectronics and Nanostructure, University of Monastir, Monastir, Tunisia
bartouli.monia@hotmail.com (corresponding author)

Amina Msolli
Laboratory of Micro-Optoelectronics and Nanostructure, University of Monastir, Monastir, Tunisia
amina.msolli@yahoo.fr

Abdelhamid Helali
Laboratory of Micro-Optoelectronics and Nanostructure, University of Monastir, Monastir, Tunisia
Abdelhamid.helali@gmail.com

Hassen Fredj
Laboratory of Micro-Optoelectronics and Nanostructure, University of Monastir, Monastir, Tunisia
hassenfredj@gmail.com

Received: 15 May 2024 | Revised: 3 June 2024 | Accepted: 12 June 2024

ABSTRACT

A smart grid is a modern electrical system that uses information technology, including sensors, measurement tools and communication devices, to monitor and improve the efficiency of the power system. However, real-time forecasting remains a challenge due to its complexity. This paper presents a forecasting framework that combines Convolutional Neural Networks (CNN) and Bidirectional Long-Short Term Memory (BiLSTM) for real-time load forecasting in smart grids. Compared to traditional methods like ARMA and Decision Trees (DTs), the proposed CNN-BiLSTM model demonstrates superior performance in terms of prediction accuracy, reaching up to 99% - higher than Long-Short Term Memory (LSTM) (93%) and Support Vector Machine (SVM) (84%). Additionally, the CNN-BiLSTM model requires fewer computational resources, with 90 Gigaflops (G) and 94 Million (M) parameters, compared to 151 (G) and 120 (G) for ARIMA and CNN-LSTM, respectively. These results indicate the proposed model's ability to accurately predict power system loads in real time with high computational efficiency.

Keywords-smart grid; Bayesian optimization; load forecasting; BiLSTM; CNN

I. INTRODUCTION

Predicting load in real-time for clever grids refers to using device learning, data mining, and related techniques to forecast the electricity demand inside the electrical grid machine at that moment [1]. This enhances the basic reliability and performance of the grid by assisting grid operators to intelligently allocate strength sources. The primary challenge in real-time predictive modeling for electrical loads in intelligent grids is the varied and complicated nature of the data.

The variables that affect the amount of power used by the grid machine include time, location, and kind of load, in addition to possible noise and anomalies. Utilizing the appropriate feature extraction and statistics training procedures is crucial to raising the accuracy and reliability of the predictions. Thus, this study is driven by the increasing demand for more accurate and faster load forecasting response to developments in the smart grid age. Due to the fact that conventional load forecasting techniques typically do not succeed in meeting these shifting demands, an additional novel and real-time approach is being investigated. Given that Deep Learning (DL) models can process substantial volumes of data and autonomously learn styles and capabilities, they are frequently deployed in power plant load forecasting. The hyperparameter technique, which applies Bayesian optimization, further improves the prediction capabilities of these DL models. A valuable outcome of this technique, is the incorporation of DL models into the power system load predictions. Developing an accurate and reliable load forecasting system is the primary goal of this research, which aims to improve the efficiency, effectiveness, and reliability of clever grid architectures.

Many intelligent grid applications, including energy purchasing and selling and energy dispatching, are capable of real-time load prediction. Recurrent neural networks, device learning, and conventional time-series modeling are common approaches for smart grid load prediction in the present day.

The classic modeling time-established record approach, which includes methods like ARMA and ARIMA, does not safeguard exogenous variables. It mainly depends on endogenous factors, instead. Despite their apparent simplicity, these styles have limits in terms of their ability to depict non-linear connections. Their over-reliance on time series data, which demonstrate stability or on data that use analysis to appear as steady information is the cause of these limitations. Precise predictions are challenging to be obtained by deploying the traditional sequential information modeling technique owing to the specifics of the modern strength used in intelligent grid systems. Furthermore, it is difficult to guarantee the version's continuous effectiveness in a setting where real-time loads are always changing. The primary reason for this is that conventional time collection forecasting algorithms do not adjust after initial training.

The device learning approach forecasts modifications to time series data by analyzing and evaluating datasets and employing methods, such as logistic regression, Bayesian Optimization (BO), and the SVM version. This version's speed gives it the advantage of being intelligible and adaptable to handling interactions across non-linear capabilities. However, one limitation of intelligent grids is that real-time demand depends on many different factors. As a result, a number of preferences may remain unfulfilled, and thus the efficacy of gadget mastering techniques may not be enhanced. The recurrent neural community approach uses state-of-the-art DL algorithms, such as the Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Generative Antagonistic Networks (GAN), etc., to automatically extract features from large datasets. One of the many notable benefits of this model is the strong learning skills it possesses. Larger datasets also improve its flexibility and performance. It does, however, have several drawbacks, such as high technology requirements and limited mobility. Additionally, its interpretability can be restricted, and its effectiveness greatly depends on the availability of information.

Taking into account the benefits and drawbacks of the previously described models, this work provides a forecast model that combines a CNN-dependent device with a neural network with BiLSTM. Additionally, the BiLSTM network improves the accuracy of the LSTM model by refining the output data. The outcomes are then exposed to BO for real-time adaptability in smart grid load forecasting. This is accomplished by dynamically modifying parameter values while utilizing Bayesian ideas. This system leads to the proposed model's production. The primary contributions of this study are:

- Model Creation: Based on the CNN-BiLSTM architecture, a neural community version is developed that utilizes DL to forecast hundreds of electrical devices. In order to provide accurate loading predictions in the future, this model efficiently utilizes CNN to extract spatial features and BiLSTM to report temporal collection patterns.
- Hyperparameter optimization: The BO procedure is deployed to alter the hyperparameters in order to increase the version's predictive capacity. Using a changeable lookup location to improve the version's stability and capacity generalization, this approach quickly determines the optimal collection of hyperparameters.
- Actual-time load forecasting: The version is employed to forecast loads using cutting-edge technology, and its predictive performance is verified by utilizing actual-time records. Accurately anticipating/estimating demand fluctuations is essential for intelligent dispatching and grid operations. This specific reason was predicted to double safety and efficiency in power systems.

Moreover, non-linear connections, which could cause issues in conventional time collection modeling techniques, may be controlled through this model's implementation. In contrast to conventional methods, the proposed model may be used in a wider range of applications. It also exhibits improved mastery and interpretability. In contrast to conventional DL models, namely the RNN, GAN, and GRU, the employment of the BiLSTM model enhances the speed at which sequential statistics are processed, more accurately catches long-term patterns in records, and applies BO to further enhance prediction accuracy.

II. COMPARABLE WORKS

A. GRU Model

GRU belongs to the RNN category. It features a more basic network architecture and operates similarly to the LSTM. Three new gate features have been added by the LSTM, the enter, forget, and exit gates. With the GRU model, there is one "door" less. The qualities are comparable and often become more reasonable despite this diversity. The Gate recurrent unit is commonly used for two purposes, language interpretation and dialogue. Due to its ability to handle sequential facts, it is ideal for tasks, such as language modeling, device translation, and textual content technology [2].

B. Decision Tree (DT) Model

DT [3] is a type of supervised learning model used in Machine Learning (ML) for prediction and classification. To properly categorize data, the procedure involves examining the features, choosing the best possibilities, and dividing the dataset into segments recursively. DTs may not be as reliable when used with unknown data although they often work well on datasets used in education.

C. ARMA Model

The ARMA model is substantial for the evaluation of the data collected over time [4]. It is frequently used in studies to perform an in-depth research. It integrates components of the Auto-Regressive version [5] and the moving average model [6]. It evaluates data stationarity to see if time series data are suitable for modeling. The adaptability of the ARMA model lies in its ability to be utilized for a wide range of data collection time frames. This also simplifies model analysis, making evaluation easier. The ARMA model has the advantage of being able to be deployed in multiple time series and to assess the model quality during the model diagnosis procedure, which is extremely beneficial for forecasting. However, when utilizing the ARMA model to predict data, the prediction error increases constantly in accordance with the elapsed time compared to the short-term prediction results.

III. METHODOLOGY

The electrical load data used in this study were sourced from two major databases: Elia [9] and ISO-NE [10]. The data span a time period from January 2014 to December 2015, including variables such as hourly load measurements and temperature. Detailed pre-processing steps involve handling missing values, normalization of the data, and segmentation into training, validation, and test sets.

A. Synopsis of the Network

This work proposes a CNN-BiLSTM model based on BO with the goal of forecasting real-time load data for smart grids. The model automatically modifies hyperparameters to enhance prediction performance using a BO technique. It combines the advantages of CNN with BiLSTM neural networks.

The BO model mainly deploys CNN to extract spatio-temporal features from load data, BiLSTM to model spatio-temporal features and realize real-time load prediction, and a BO style algorithm to fine-tune model hyperparameters, such as batch size and learning rate, to improve prediction performance and model applicability in general.

The three key components of the model – BO, BiLSTM, and CNN work together to provide an accurate, real-time electricity forecast for smart power grids. The CNN extracts spatiotemporal capabilities, the BiLSTM models these features, and BO adjusts the hyperparameters to increase the accuracy and utility of the forecasting model.

The workflow of the model is to introduce the real-time load data of the smart grid into the process, preprocess and normalize the data input layer, and apply the CNN unit to extract the features of the dataset. The model uses a one-dimensional convolution layer to maximize feature extraction and subsequently generates a final feature series after pooling, sampling, combining, and restructuring the data through the fully connected layer.

Finally, after feeding the feature data into the BiLSTM layer for additional training, the BO procedure identifies the proper model parameters. This approach simultaneously improves the CNN-BiLSTM structure and forecast accuracy to give more precise forecast results for real-time load data in smart grids.

Figure 1 presents the flow chart of the model, which describes the steps involved in real-time load anticipation for the smart grid. The first basic input is obtained from the smart grid real-time load data. This data stream is then subjected to a Softmax function, which simplifies initial processing and normalization. The output of the Softmax function is received by a CNN to further process the data and extract relevant features. Using a CNN with an one-dimensional convolutional layer helps reduce dimensionality and more efficiently extract features for the full dataset. After completing the CNN analysis, the data are decomposed to generate an one-dimensional array from the multiplied CNN output. Subsequently, a BiLSTM, famous for its ability to integrate data sequences in both directions, seamlessly integrates all functionality. The interaction between the BiLSTM layer and the BO makes it possible to improve the model hyperparameters and enhance the accuracy of load forecasts as the flowchart evolves. Ultimately, this method can reliably predict load patterns in the smart grid, leveraging real-time data to improve network performance.

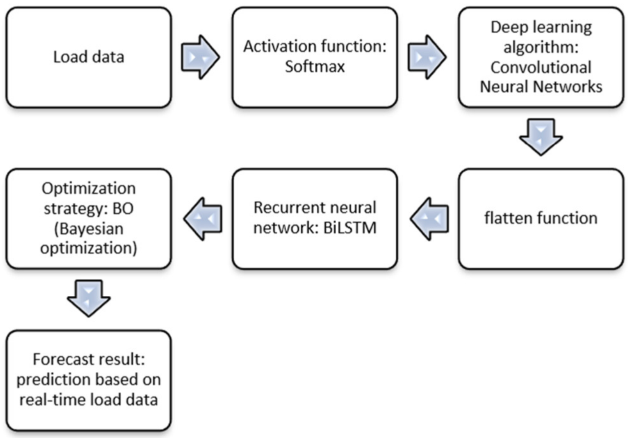

Fig. 1. The method followed for the smart grid to forecast load in real time.

B. Data Preprocessing and Normalization

To ensure efficient training, the input data of the smart grid load-forecasting model are first preprocessed and normalized. This includes:

- Handling missing values: Any missing data points in the time series of smart grid load measurements are imputed using interpolation techniques. This could involve simple linear interpolation or more advanced methods like spline interpolation to estimate the missing values based on the surrounding data points.
- Feature scaling: The input features, which may include the raw load data as well as engineered temporal features, are standardized by subtracting the mean and dividing it by the standard deviation of each feature. This ensures that all features have a similar numerical range, preventing certain features from dominating the model training process.
- Temporal feature engineering: In addition to the load data, the preprocessing step also involves extracting relevant temporal features that can help the model capture patterns in electricity consumption. Examples of such features include: Time of the day (hour of the day), day of the week, Holiday indicators (whether the day is a holiday or not), and Seasonal indicators (e.g. month of the year).

C. CNN Architecture

CNN is a popular DL technique [7] that excels in local feature recognition. CNN has become one of the most popular models due to its adaptability and ability to handle a variety of tasks, such as target detection and picture type [8].

The convolution layer's function is the primary purpose of feature extraction. The input picture is analyzed and processed using convolutional kernels in order to extract many features. It is possible for many kernels to operate independently. The resulting feature map is gathered in the stratum of pooling. This approach notably lowers the quantity of statistics by eliminating redundant data while keeping important elements.

The mathematical expression of the one dimension convolution layer of the convolution process with a convolution kernel is:

Let p_a be the set of input features, denoted as:

$$p_a = \{p_{a_1}, p_{a_2}, \dots, p_{a_{n-1}}\} \quad (1)$$

where n is the number of input features. Let K be the convolution kernel. The output of the Y convolution can be calculated as:

$$Y_i = \sum_{j=0}^{m-1} X_{i+j} * K_j \quad (2)$$

where Y_i is the value of the output at index i, X_{i+j} is the value of the entry at index $i + j$, K_j is the value of the convolution kernel at index j, and m is the size of the convolution kernel.

D. BiLSTM Model

The LSTM model effectively processes data from the forward sequence to make neural network predictions. However, it might be difficult to comprehend the substance of backward data during model training, which can result in problems like disappearing gradients or gradient explosions. By combining forward and backward LSTM units, the BiLSTM model solves this issue and improves the preservation of data from far-off nodes. BiLSTM improves performance with huge temporal sequence data by creating dual hidden layer representations for each input by combining forward and backward calculations. The BiLSTM structure, with its integrated forward and backward units, performs better in time series forecasting than LSTM units do even though the latter are excellent at using forward data for forecasts. Each LSTM cell in the BiLSTM structure, portrayed in Figure 2, has three gating mechanisms, an input gate, an output gate, and a forgetting gate.

It is possible to observe that the forward LSTM structure in the BiLSTM network is computed similarly to a single LSTM. By combining the forward hidden layer state and the reverse hidden layer state, the hidden layer state of the BiLSTM network can be obtained.

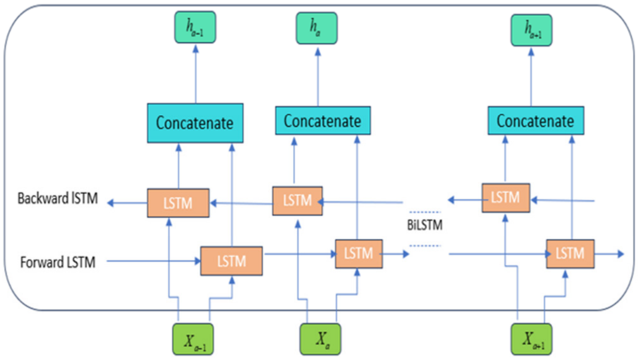

Fig. 2. Calculating BiLSTM units for real-time load forecasting in smart grids.

E. Bayesian Optimization

BO is an approach that leverages data from previous search points to identify the next search point, with the aim of solving small black box optimization problems. This sequential optimization process by using a model makes it possible to obtain a quasi-optimal solution to generate/for a model with a very low evaluation cost [9]. BO is frequently utilized to classify texts, predict several categories of texts in real time, and distinguish sentiments. At the same time, BO is also being used more to predict sequential data. Its structure is illustrated in Figure 4.

The architecture of the proposed model, which is divided into three main portions, is displayed in Figure 3. The first section utilizes three sub-blocks of a CNN, namely input data, an one-dimensional maximum pooling convolutional layer, and a dropout. In order to prevent overfitting, this approach entails processing input data via an one-dimensional convolutional layer with maximum pooling and then performing a dropout operation. In the middle segment, a BiLSTM network is formed by employing LSTM layers to identify long-term relationships in data sequences. To improve model performance through hyperparameter adjustment, BO is incorporated into the model. The resulting scores are transformed into probabilities deploying the SoftMax activation function. The output section displays the model's final result.

Figure 4 depicts the architecture of the model, which uses neural networks for learning and BO. The flowchart illustrates a series of actions and choices, including crucial phases like model startup, BO, and result output. CNNs were utilized in the proposed model to handle the input data, long-term dependencies were modeled using LSTM cells, and hyperparameter tuning was accomplished following a BO procedure. After that, the model's output is used to direct behavior to a specific setting.

IV. EXPERIMENT

In the experimental setup, critical components are defined to ensure the success of the real-time load forecasting model. Data preprocessing steps involve gathering historical load and weather data, cleaning the data, engineering features, and normalizing the data for consistency. Model parameter settings include defining the architectures of CNN and BiLSTM layers, setting the learning rates, batch sizes, and regularization techniques. Evaluation metrics, such as Mean Absolute Error, Root Mean Squared Error, R-squared, and accuracy are utilized to assess model performance. The experimental setup includes hyperparameter tuning using BO, model training, validation to prevent overfitting, testing on a separate dataset, and iterative improvement based on the evaluation results for model refinement and enhancement.

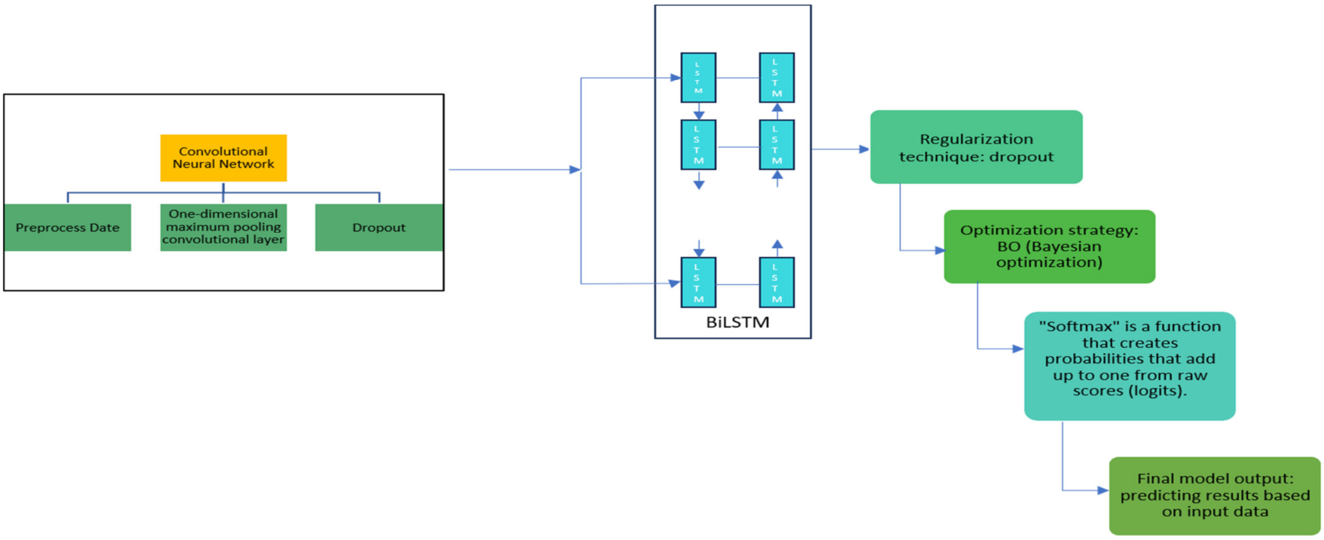

Fig. 3. The architecture of the proposed model.

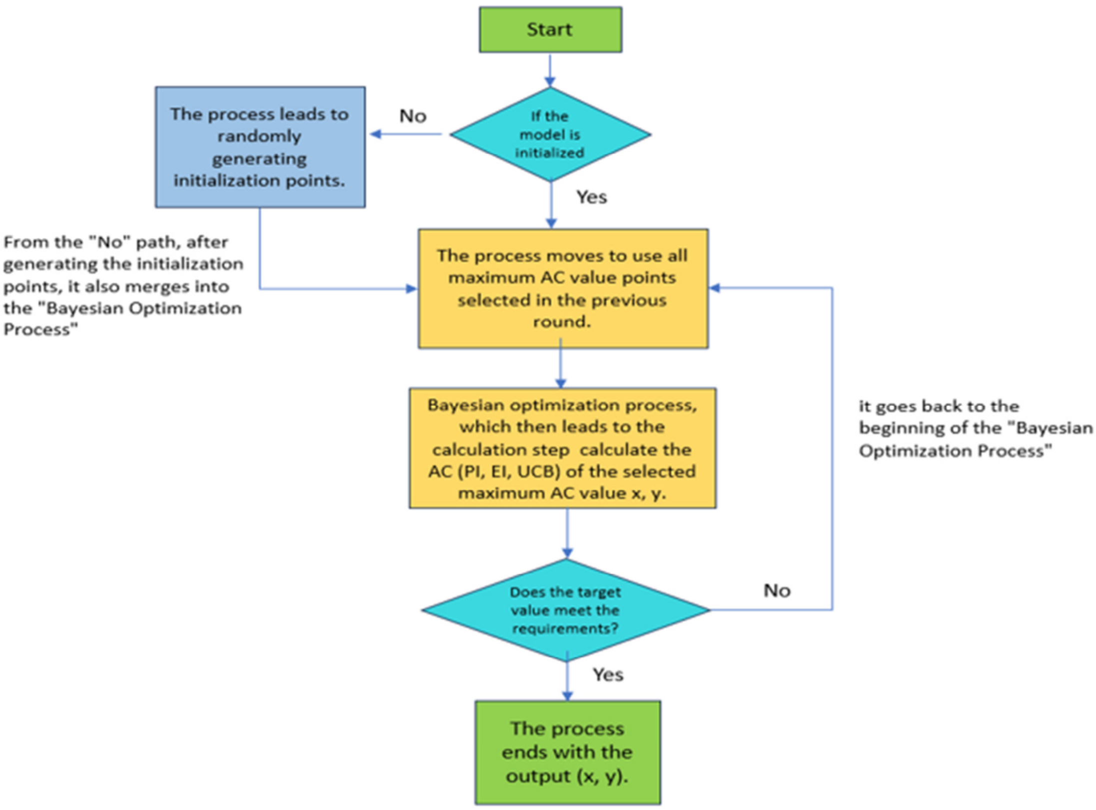

Fig. 4. Detailed flowchart of the proposed model based on Bayesian optimization to adjust and optimize hyperparameters.

For this study's experiments, load data from the Elia and ISO-NE datasets, covering the tome period from January 2014 to December 2015, were utilized. Data pre-processing included normalizing the load values to a range from 0 to 1, addressing missing entries through regression imputation, and dividing the data into 80% for training and 20% for validation and testing. It was ensured that the data used for the model evaluation were not included in the training process to avoid data leakage and certify unbiased performance metrics.

A. Collections of Data

Elia and ISO-NE were the statistical resources for the data utilized in this work. ISO-NE is the organization responsible for overseeing the power and electrical sectors in the New England region, including managing the market, energy devices, and ensuring reliable electricity supply [11]. Elia is the high-voltage transmission grid operator in Belgium, responsible for managing the transmission network, market operations, and enabling cross-border energy exchanges to promote Belgium's social and economic growth [10].

B. Configuration and Specifics of the Experiment

Several tests were carried out to verify the introduced model's effectiveness and demonstrate its performance. Initially, the proposed approach was contrasted with different approaches in terms of how long it took to draw conclusions from the complicated data. It was then displayed how much better the proposed model was than the other models by contrasting its training time with that of the other models and evaluating how well each model performed at various levels of complexity. Additionally, experiments were conducted to test the number of parameters. Lastly, the accuracy and calculation time of the two datasets were compared under the examination of various models. It was found that the proposed model performs better than other approaches in terms of computation speed, quantity of parameters needed, and the experimental results. As a consequence, the proposed model is able to forecast smart grid real-time load data more accurately.

C. Analysis and Outcomes of the Experiment

Figure 5 clarifies that the proposed model performs better than the other two with reference to the inference duration for identical complicated data, while its overall performance is correspondingly better.

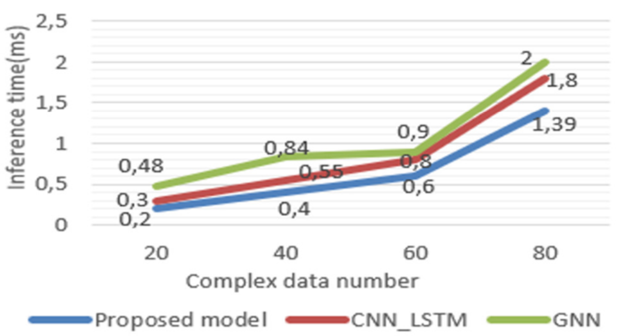

Fig. 5. Evaluation of inference time in different models for complex data.

The training times of the various models using the complex data are showcased in Figure 6. The differences in training times between the SVM and LSTM models were examined. It is evident that the proposed model takes a shorter time compared to that of the SVM and LSTM models, regardless of the volume of the data. This allows the model to make more contributions at once and drastically cut down on training time.

Fig. 6. Assessment of data training duration across various models.

In this series of tests, as seen in Figure 7, each model's computational flop was tested and evaluated. The experimental findings indicate that the ARIMA model requires the greatest computational effort. While the proposed model does not perform the best in this set of tests, it does perform much better after using BO, which yields solid experimental findings that support the viability of the proposed approach.

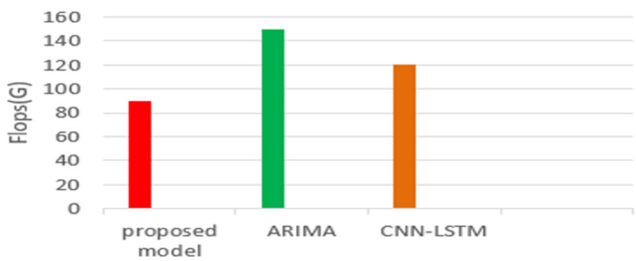

Fig. 7. Number of failures, or computations, needed for various models.

Figure 8 exhibits the number of parameters needed by the various models. After a number of tests were carried out, it can be concluded that in comparison with the selected models, the GNN model uses the greatest number of parameters, but the LSTM operation utilizes a significantly smaller number of parameters than GNN. The proposed approach functions incredibly well regarding the number of parameters required for operation. Reducing the number of variables in a model can enhance its ability to calculate data.

D. Steps of the Model

The steps of the proposed model are:

- Real-time load data from the clever grid are obtained, preprocessed, and normalized in the fact's entry layer.
- One-dimensional CNN is used to process the dataset in order to extract features.
- The typical collection is generated after the dimensionality is discounted and after additional processing.
- The BiLSTM layer receives feature-related records in order to provide real-time load feature acquisition. Then, by

employing BO to find the nice model parameters, prediction accuracy is increased.

- The forecast's output is provided with astonishing precision.

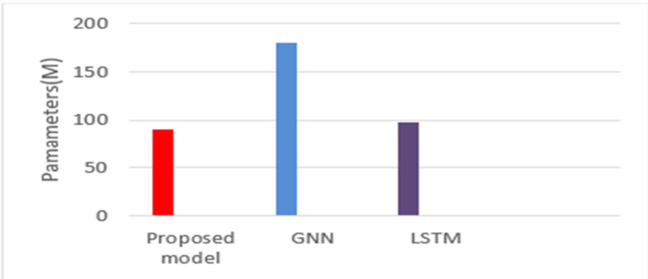

Fig. 8. Number of parameters in different models.

Two datasets were employed for the experiments, which were included in multiple models in this set of tests (Figure 9). It is easy to see that the results of the experiments on the two datasets show that the proposed model takes the shortest computation time, while GNN takes the longest.

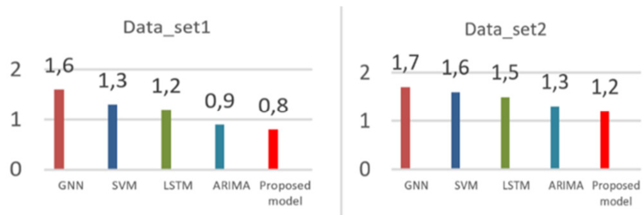

Fig. 9. Comparison of the various models' calculation times.

In the final set of tests, portrayed in Figure 10, tests were run on the two selected datasets, using three models in order to gauge how accurate the results were.

The experimental findings reveal that the accuracy of the two chosen models, LSTM and SVM, differs dramatically when faced with diverse data sets compromising the accuracy of the experiment. The test results are preferable if the models do show steady experimental stability when compared to other data. The proposed model performs equally well on the two chosen datasets leading to more accurate and persuasive test results.

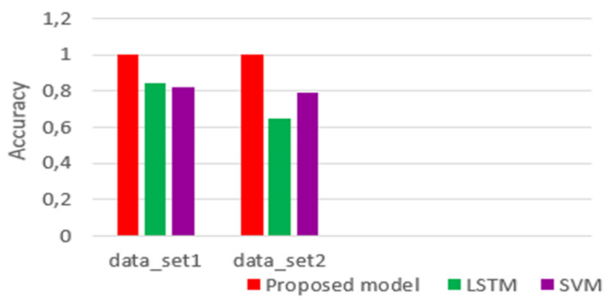

Fig. 10. Differences in model accuracy on the dataset.

V. CONCLUSION

The study highlights the success of the proposed forecasting framework that combines Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) for real-time load forecasting in smart grids. The CNN-BiLSTM model achieved an exceptional prediction accuracy of up to 99%, surpassing the LSTM and SVM models, which scored 93% and 84% accuracy, respectively. Moreover, the model demonstrated efficiency by requiring fewer computational resources compared to the other methods. These results emphasize the model's accuracy and computational efficiency in forecasting power system loads in real-time. Accurate real-time load forecasting is crucial for enhancing electric device functionality, promoting renewable energy utilization, and improving market performance in smart grid systems. By enabling a precise estimation of electricity demands, the CNN-BiLSTM model can enhance the planning and control of power system operations, leading to increased efficiency and reliability in smart grid management. The study suggests future work should be done in certain areas. This involves exploring complex model architectures and integrating external factors while also acknowledging limitations, like data quality, generalizability, and model interpretability that need to be addressed for further research and model enhancement in real-world smart grid applications.

REFERENCES

[1] V. S. Aravind, M. S. Anbarasi, P. Maragathavalli, M. Suresh, and M. Suresh, "Smart Electricity meter on Real time Price Forecasting and Monitoring system," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, Mar. 2019, pp. 1–5, https://doi.org/10.1109/ICSCAN.2019.8878836.

[2] R. Zhang, F. Zeng, X. Cheng, and L. Yang, "UAV-Aided Data Dissemination Protocol with Dynamic Trajectory Scheduling in VANETs," *2019 IEEE International Conference on Communications, ICC 2019 - Proceedings*, pp. 1–6, May 2019, https://doi.org/10.1109/ICC.2019.8761170.

[3] F. Xu, Y. Zheng, and X. Hu, "Real-time finger force prediction via parallel convolutional neural networks: a preliminary study," *Annual International Conference IEEE Eng Med Biol Soc*, vol. 2020, pp. 3126–3129, Jul. 2020, https://doi.org/10.1109/EMBC44109.2020.9175390.

[4] Z. Song, R. M. Johnston, and C. P. Ng, "Equitable Healthcare Access During the Pandemic: The Impact of Digital Divide and Other SocioDemographic and Systemic Factors," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 4, no. 1, Art. no. 1, Feb. 2021.

[5] H. Zhang, F. Zhang, B. Gong, X. Zhang, and Y. Zhu, "The Optimization of Supply Chain Financing for Bank Green Credit Using Stackelberg Game Theory in Digital Economy Under Internet of Things," *Journal of Organizational and End User Computing*, vol. 35, pp. 1–16, Jan. 2023, https://doi.org/10.4018/JOEUC.318474.

[6] R. Estrella, G. Belgioioso, and S. Grammatico, "A shrinking-horizon, game-theoretic algorithm for distributed energy generation and storage in the smart grid with wind forecasting," in *15th IFAC Symposium on Large Scale Complex Systems*, Delft, The Netherlands, May 2019, pp. 126–131, https://doi.org/10.1016/j.ifacol.2019.06.022.

[7] M. M. H. Milu, M. A. Rahman, M. A. Rashid, A. Kuwana, and H. Kobayashi, "Improvement of Classification Accuracy of Four-Class Voluntary-Imagery fNIRS Signals using Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, Art. no. 2, Apr. 2023, https://doi.org/10.48084/etasr.5703.

[8] X. Ning, P. Duan, and S. Zhang, "Real-Time 3D Face Alignment Using an Encoder-Decoder Network With an Efficient Deconvolution Layer,"

IEEE Signal Processing Letters, vol. 27, pp. 1944–1948, Nov. 2020, https://doi.org/10.1109/LSP.2020.3032277.

[9] Z. Zou, M. A. Abdul Careem, A. Dutta, and N. Thawdar, "Joint Spatio-Temporal Precoding for Practical Non-Stationary Wireless Channels," *IEEE Transactions on Communications,* vol. 71, no. 4, pp. 2396–2409, Nov. 2022.

[10] G. Shen, Z. Wenbin, Y. Zhang, C. Han, and P. Liu, "Determination of the average maintenance time of CNC machine tools based on type II failure correlation," *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, vol. 19, pp. 604–614, Sep. 2017, https://doi.org/10.17531/ein.2017.4.15.

[11] C. Ziheng, F. Silvestri, G. Tolomei, J. Wang, H. Zhu, and H. Ahn, "Explain the Explainer: Interpreting Model-Agnostic Counterfactual Explanations of a Deep Reinforcement Learning Agent," *IEEE Transactions on Artificial Intelligence*, vol. 99, pp. 1–15, Jan. 2022, https://doi.org/10.1109/TAI.2022.3223892.