University Network Traffic Patterns Prediction Using LSTM and RBM

OMONIYI, VICTORIA IBIYEMI[1], AKINYOKUN OLUYOMI KOLAWOLE[2]
[1]*Department of Software Engineering, Federal University of Technology, Akure, Ondo State, Nigeria*
[2]*Department of Cyber Security, Federal University of Technology, Akure, Ondo State, Nigeria*

Abstract - Accurate prediction of university network traffic is essential for efficient resource management, resource optimization, security enhancement, and optimal user experience. Traditional statistical methods often struggle with network traffic data's complex, nonlinear, and time-varying nature. These challenges have been successfully addressed through recent advancements in deep learning, particularly the development and application of Long Short-Term Memory (LSTM) networks. This paper introduces a novel approach to network traffic prediction by integrating Long Short-Term Memory (LSTM) networks and Restricted Boltzmann Machines (RBM). LSTM is a specific architecture within the family of recurrent neural networks, and it is adapted to predict network traffic Patterns in dynamic university environments. Comprehensive experiments are carried out utilizing real-world network traffic data collected from university environments. The findings reveal that the proposed LSTM-based model performs robustly across all major metrics, achieving low values for Test Loss, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), along with a high R^2 score, signifying outstanding accuracy and generalization capabilities. LSTM proves to be highly capable of handling time-series data or sequence-based tasks.

Index Terms: Network-Traffic, Prediction, Long Short-Term Memory, Restricted Boltzmann Machines

I. INTRODUCTION

In recent years, universities worldwide have experienced a significant digital transformation, integrating advanced technologies within various academic disciplines and administrative activities. This transformation has led to an exponential increase in network traffic within university campuses, driven by a multitude of factors such as online learning platforms, research collaborations, administrative systems, e-library, e-voting, computer-based tests/Exams, and the proliferation of internet-enabled devices.

The efficient management of university network infrastructure has become paramount to ensure seamless operations, optimal resource allocation, and enhanced user experience. Network administrators face the challenge of accurately predicting and managing network traffic behavior to meet the evolving needs of students, faculty, staff, and other stakeholders. [5].

Universities, as dynamic hubs of knowledge and innovation, increasingly rely on advanced network infrastructures to facilitate a wide array of activities, ranging from academic research and collaborative projects to online learning platforms. With the proliferation of digital technologies, the volume and complexity of network traffic within university environments have surged, presenting challenges for effective network management. The need to anticipate, understand, and optimize network Patterns has become paramount for ensuring seamless connectivity, resource allocation, and overall operational efficiency. [10].

This research investigates the prediction of network traffic patterns within university environments, leveraging the synergies between deep learning and optimization techniques. As the demand for online education, research collaboration, and data-intensive applications continues to grow, There is an urgent need to create advanced predictive models that can adapt to the changing dynamics of university network traffic Understanding and predicting the behavior of university network traffic is crucial for network administrators to proactively manage resources, optimize bandwidth allocation, and ensure a seamless digital learning environment. Traditional methods of network traffic analysis and management may fall short in handling the complexity and dynamic nature of modern university network. [3]. This research holds significance in the realm of network management for educational institutions.

Across several academic institutions, especially in Universities, detailed and up-to-date analysis of the distribution of network services is critical to delivering efficient academic and administrative services. With the development of the Internet and advanced web applications, the computer network is changing students, staff, and university community lives, the internet makes life easy. Effective internet traffic management is essential to support web-based applications and other internet-dependent devices. This can be accomplished by predicting university network traffic based on historical traffic data. [8]. Predicting network traffic patterns becomes crucial for optimizing resource allocation, ensuring quality of service, and preventing congestion.

[9] examined the properties of network traffic and discovered its self-similar nature. This self-similarity in time series can be quantified using the Hurst exponent, which they found to be greater than 0.5 for network traffic. The self-similar nature of network traffic makes it inherently predictable, and their research established a foundational basis for network traffic prediction. Since then, numerous scholars and researchers have conducted extensive studies in this area. [14].

When network traffic is accurately predicted, it can provide significant benefits by helping to manage network resources to ensure quality of service and assisting network administrators in improving availability and transmission speeds. Different research, designs, and experiments have been conducted or proposed for analyzing and predicting network traffic. Predicting network traffic patterns offers significant advantages across various domains, including dynamic bandwidth allocation, enhanced network security, optimized network planning, proactive congestion control, and efficient resource management. [8].

Deep learning encompasses a suite of machine learning techniques capable of autonomously extracting meaningful features from raw data through the construction of hierarchical feature representations. This has been fairly common knowledge in the machine-learning community for a long time. However, what has made deep learning so exciting in recent years is that the learning process can be directly applied to finding a useful representation of the data, rather than just learning a mapping from input to output. [13]. This is useful in many real world scenarios, for example an engineer might not have a good idea of what features to extract from raw sensor data in order to build an effective detector for a given event. If the engineer knew what features were best to extract, then building the detector could just be a simple optimization problem with respect to a classification function. Deep learning provides a means for the optimization to be performed directly on the detector, given a simple parameterization of the features. [11].

II LITERATURE REVIEW

There had been numerous researches on network traffic prediction, with a specific emphasis on utilizing deep learning methodologies for the accurate prediction of network traffic behavior. Such among them are:

Xueyan *et al.*, [17] propose a BWCL method for network traffic prediction, combining a Butterworth filter, CNN, and LSTM. The Butterworth filter smooths data, while CNN-LSTM models analyze different frequency bands, enhancing prediction accuracy by aligning data characteristics with suitable models. This method aids in resource planning and performance optimization during traffic bursts, offering precise insights into time series data dynamics.

Yuantao L. [18] introduced a deep learning model for network traffic prediction that integrates CNN, LSTM, and Bayesian optimization. The CNN extracts spatial features, LSTM captures temporal dependencies, and Bayesian optimization improves prediction efficiency. The model outperforms traditional methods like LSTM and GRU in predicting peaks, valleys, trends, and overall traffic patterns with greater accuracy.

Agnieszka *et al.*, [2], uses a regression approach based on the Multi-Layer Process (MLP) to predict traffic types in an application-aware backbone optical network. The model, a simple neural network, outperforms a baseline LR algorithm in all traffic types, achieving a mean absolute percentage error of 2%-10%.

Huaifeng [6] proposed the "AGG: A Novel Intelligent Network Traffic Prediction Method Based on Joint Attention and GCN-GRU," which combines GCN, GRU, and an attention mechanism to effectively capture spatial and temporal traffic patterns. This approach features an improved attention mechanism and an updated weight matrix calculation. When applied to the Milan traffic network dataset, the AGG model outperformed baseline models such as HA, ARIMA, SVR, GRU, and GCN-GRU, achieving better results across metrics like RMSE, MAE, accuracy, determination coefficient, and explained variance score.

Wang *et al.*, [16], presents a Network Traffic Prediction Method Based on LSTM, focusing on the importance of TCP/IP networks in modern society. The model uses real-world network traffic data from various sources, including European cities, UK educational networks, and China's education network. It incorporates autocorrelation coefficients to improve prediction accuracy. The model shows promise for real-world applications in network traffic prediction, allowing network service providers to optimize resources and enhance service quality. The study suggests exploring variations of recurrent neural networks, such as Gated Recurrent Unit, for further improvement.

Abdelhadi and Guy [1] explored a study titled "A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction." This research introduces an LSTM-based framework leveraging deep learning techniques to accurately predict network traffic parameters. The method converts each Traffic Matrix into a traffic vector (TV) for prediction, training the model on historical data and testing it with new inputs. The findings highlight the suitability of LSTM RNNs for traffic matrix prediction due to their capability to represent temporal sequences and capture long-range dependencies effectively.

Sebastian *et al.*, [12] proposes a Deep Learning-Based Traffic Prediction for Network Optimization, focusing on the importance of accurate network behavior for managing mobile and fixed network services. The study uses a dataset of traffic matrices from the Abilene network, processed into a vector representation, and inputs it into a prediction system. The system consists of a classical Artificial Neural Network (ANN) based on GRUs and an Evaluation Automatic Module (EAM). The model, particularly the GRU RNNs, shows high accuracy in predicting traffic matrices, enabling proactive resource allocation and optimization.

The ST-LSTM method introduced by Bi *et al.,* [4] offers a novel real-time network traffic prediction approach by integrating the Savitzky–Golay (SG) filter, temporal convolutional network (TCN), and long short-term memory (LSTM). This end-to-end methodology operates in three phases: noise removal, short-term local feature extraction, and long-term dependency modeling. The SG filter improves data quality, TCN identifies immediate patterns, and LSTM captures long-term trends. The method's advanced predictive capabilities make it ideal for diverse industrial applications.

Tiago *et al.*, [15] conducted a study titled "Computer Network Traffic Prediction: A Comparison between Traditional and Deep Learning Neural Networks." This research evaluated four prediction techniques based on Artificial Neural Networks (ANN), specifically the Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), and Deep Learning Stacked Autoencoder (SAE). The findings indicated that simpler models, such as RNN and MLP, often outperformed more complex models like SAE. The RNN, utilizing the Rprop training algorithm, emerged as the most effective method due to its ability to use prior observations for learning new data. The optimal results were obtained with the JNN and SRN models.

III METHODOLOGY

The methodology used in this research aimed to predict the network traffic using deep learning modelsFigure 1 illustrates the architecture of the predictive model. The research methodology is in six steps to achieve the research objectives. The steps are:

a. Data Collection
b. Data Preprocess
c. Algorithm selection and model development
d. Training/Testing
e. Prediction
f. Performance evaluation metric

Data Collection
The first phase of the system architecture is vital, since the quality of data gathered during this stage influences the overall performance of the system. The historical network traffic dataset examined in this study was obtained from Elizade University in Ilara Mokin Nigeria.

Data Pre-processing
In the creation of a deep learning model, data pre-processing serves as the crucial first step that initiates the entire process. Real-world data often presents challenges such as incompleteness, inconsistencies, inaccuracies, and the presence of errors or outliers. Data pre-processing involves preparing this raw data for analysis by a deep learning model, encompassing the necessary steps to transform or encode the data for optimal parsing by the model. The primary objective is to ensure that the algorithm can effectively interpret the features of the data, which is essential for achieving accurate and precise predictions. (Olayinka, *et al*, 2022).

Clean: The collected dataset was cleaned (e,g handle the missing values)

Normalization: Normalization is a method in which the values are adjusted to a range, typically between 0 and 1.

$$X_{i_{new}} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Equation 1. presents the formula for normalizing all features of xx, transforming them to a range between the specified *min* $_{new}$ and *max* $_{new}$

x_{inew} = normalized value

x_i = process value

x_{min} = minimum value in the dataset

x_{max} = maximum value in the dataset

The set of attribute with the highest accuracy is returned and selected *Kemal and Baha, 2018*

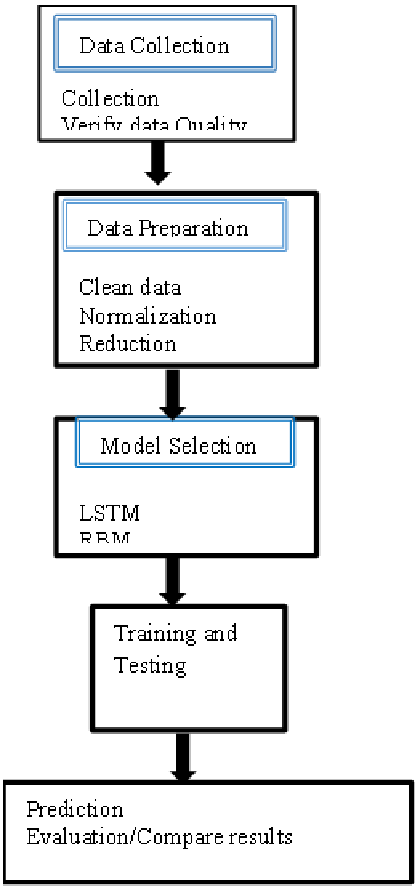

Figure 1: Framework of the prediction models.

Algorithm/model selection

The prediction models proposed for this research were as below:

I Long Short-Term Memory
ii Restricted Boltzmann Machine

Long short-term memory (LSTM)

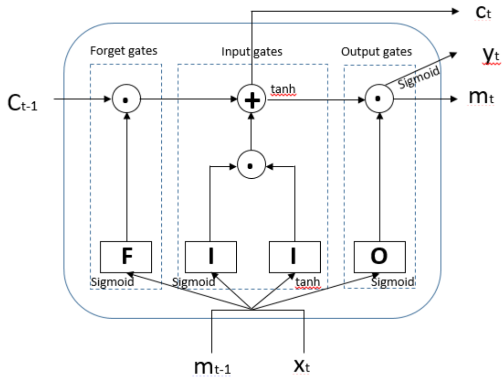

Figure 2: Department structure within LSTM.
Jihong and *Xiaoyuan (2022)*

Long Short Term Memory (LSTM) is a specific architecture of recurrent neural networks (RNNs) created to address the challenges traditional RNNs face in learning long-term dependencies. This is accomplished by employing a sophisticated structure featuring gates and memory cells, which regulate the information flow.

Components of LSTM

An LSTM unit consists of the following components

i. Cell State (C_t) Represent the memory of the network. It carries information across different time steps

ii. Hidden State(h_t): The output of the LSTM unit at each time step, which is also passed to the next unit.

iii. Gates: There are three main gates that control the flow of information.

(a) forget Gate (f): Determines which information to eliminate from the cell state.

(b) Input Gate (i): Determines which new information to incorporate into the cell state.

(c) Output Gates (o): Determines which portion of the cell state to output and transmit to the next time step

FORGET GATE

The forget gate is vital for determining which information from the previous cell state should be kept or discarded. This gate is key to the LSTM's capability to handle long-term dependencies and address challenges such as the vanishing gradient problem that often arises in traditional recurrent neural networks (RNNs). The output of the forget gate ranges from 0 to 1, where a value near 0 indicates that most of the information is discarded, while a value closer to 1 suggests that the information is largely retained. The process can be expressed as:

$$F = \sigma(W_f \bullet [m_{t-1}, x_t] + b_f) \quad (2)$$

Where:

σ is the sigmoid function

W_f represents the weight matrix

m_{t-1} represents the previous hidden state

x_t represents the current input

b_f represents the bias term.

INPUT GATE.

The input gate in a Long Short-Term Memory (LSTM) network is responsible for controlling how much of the new information generated at the current time step should be added to the cell state.

The input gate operation in an LSTM in involves two main components: the input gate itself and the candidate cell state. The process of the input gate itself can be expressed as:

a. Input gate

$$I = \sigma(W_I \bullet [m_{t-1}, x_t] + b_i) \quad (3)$$

b. The candidate cell state (I') uses a tanh activation function to create a vector of new information with values ranging between -1 and 1.

$$I' = \tanh (W_C \bullet [m_{t-1}, x_t] + b_C \quad (4)$$

Where

σ represents the sigmoid function

W_I and W_C is the weight matrix

m_{t-1} represents the previous hidden state

x_t represents the current input

b_i represents the bias term

tanh represents the hyperbolic tangent function

The input gate (I) and the candidate cell state (I'), are combined to update the cell state C_t:

$$C_t = f_t * C_{t-1} + I * I' \quad (5)$$

Where: f_t is the forget gate vector, which modulates the previous cell state (C_{t-1}). The input gate I determines the extent to which the candidate cell state should influence the new state.

OUTPUT GATE

In a long short-term memory (LSTM) network, the output gate regulates the information that is extracted from the cell state and forwarded to the next hidden state. This gate decides which aspects of the cell state should be made available to the hidden state, thereby affecting the calculations for the subsequent time step and ultimately impacting the network's output. The process can be expressed as:

$$O = \sigma(W_O \bullet [m_{t-1}, x_t] + b_O) \quad (6)$$

$$m_t = O \bullet tanh\,(C_t) \quad (7)$$

$$Y_t = sigmoid\,(W_t \bullet m_t) \quad (8)$$

Where:

m_t represents the hidden state at the current time step

σ represents the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$

tanh represents the hyperbolic tangent activation function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

W_O represents the weight matrix

$[m_{t-1}, x_t]$ represents the concatenation of the previous hidden state $[m_{t-1}]$ and current input $[\,x_t]$

b_O represents the bias vector for the output gate, learned during training

C_t represents the cell state at the current time step

Y_t represents the output of this time.

Restricted Boltzmann machine (RBM)

A Restricted Boltzmann Machine (RBM) is a generative stochastic neural network that learns probability distributions from its inputs. It features two layers: a visible layer that represents observable data, and a hidden layer that captures latent features. Notably, there are no connections between the units in the same layer, which distinguishes it as a restricted variant of the general Boltzmann Machine.

An RBM is composed of the visible units *V = (Vj) j∈M* with visible state vector *v = (v j) j∈M*, the hidden units *H = (Hi)i∈N* with hidden state vector *h = (hi)i∈N* , and the weight state matrix *w = (wji) j∈M,i∈N* , which connects the visible units and hidden units, where *M* is the number of visible units and *N* is the number of hidden units. The joint probability distribution *P(v, h)* in RBM is defined as follows:

$$P(v,h) = \frac{e^{-\varepsilon(v,h)}}{\sum v \sum_h e^{-\varepsilon(v,h)}} \quad \text{`(9)}$$

1. Energy Function:

The energy function E(v,h) for the visible units v and the hidden units h is defined as:

$$\mathrm{E}(v,h) = -\sum_i \alpha_i v_i - \sum_j h_j b_j - \sum_{i,j} v_i\, h_j w_{ij} \quad (10)$$

Where:

v_i and h_j are the states of the visible and hidden units. (where: v is binary state of visible unit of I, h is a binary state of hidden unit at j)

w_{ij} is the weight between visible unit v_i and hidden unit h_j

α_i and b_j are the biases of the visible and hidden units respectively.

The structure itself assigns a probability to each connection vector between hidden and visible units. This probability can be expressed mathematically using an energy function.

2. Probability Distribution:

The joint probability distribution of the visible and hidden units is given by:

$$p(v,h) = \frac{1}{z} e^{-E(v,h)} \quad (11)$$

Z here is the partition function is given by summing over all possible pairs of visible and hidden vector.

$$Z = \sum_{v,h} e^{-E(v,h)} \quad (12)$$

The marginal probability of a visible vector v is :

$$p(v) = \frac{1}{z} \sum_h e^{-E(v,h)} \quad (13)$$

3. Conditional Distribution.

The conditional probability of a hidden unit being activated, given the observed values of the visible units in the model, and vice versa are:

$$p\ (h_j = 1 \mid v) = \sigma\ (b_j + \textstyle\sum_i v_i w_{ij}) \quad (14)$$

$$p\ (v_i = 1 \mid h) = \sigma\ (a_i + \textstyle\sum_i h_j w_{ij}) \quad (15)$$

Where: $\sigma(\mathrm{x}) = \frac{1}{1+\exp(-x)}$ is the logistic sigmoid function.

4. Training (Contrastive Divergence):

The training process for an RBM involves iteratively adjusting the weights and biases of the model to minimize the difference between the distribution of the observed data and the probability distribution represented by the model. One common training algorithm is contrastive Divergence (CD).

* Positive Phase: Compute the expected value of the outer product of the visible and hidden vectors given the data.

$$(v_i, h_j)data = P(\ h_j - 1|v)v_i \quad (16)$$

* Negative Phase: Sample a reconstruction of the visible units from the hidden units and then compute the expected value of the outer product of the visible vectors given the reconstruction.

$$(v_i, h_j)\ model\ = P(\ h_j - 1|v')v_i' \quad (17)$$

The weights are updated as:

$$\Delta w_{ij} = \alpha(v_i h_j)data - (v_i h_j)model \quad (18)$$

Where a is a learning rate, $(v_i h_j)$ means the expectation over the associated distributions

Analysis of experimental results.
To prepare the data for model training and evaluation, the cleaned dataset was divided into a training set and a testing set. This split was used throughout the training and testing phases for both models (LSTM and RBM). Specifically, 80% of the data was allocated for training, while the remaining 20% was reserved for model evaluation. Figures 3 and 4 illustrate the comparison between the actual and predicted curve values derived from the test set for the LSTM and RBM models, respectively. This section will delve into the results from the training and testing processes.

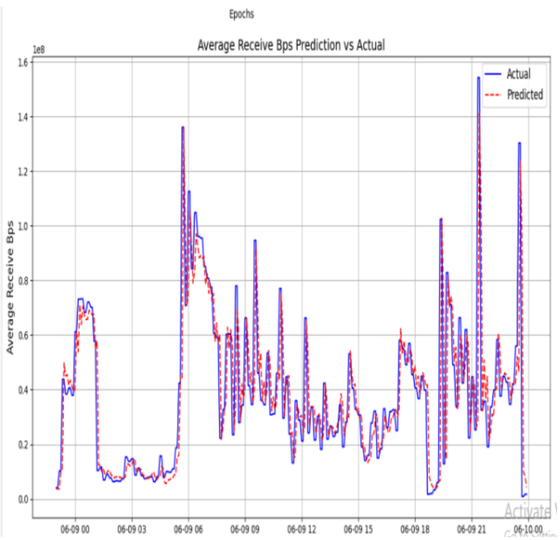

Figure 3: An evaluation of the LSTM model's ability to predict curve values by comparing actual and predicted curve data.

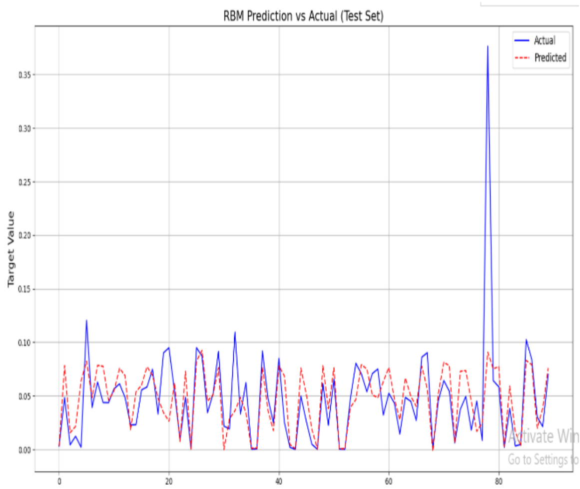

Figure 4: Assessing the RBM model's performance through a comparison of actual curve values with the curve values predicted by the model.

Comparison Experiments of Prediction Models

To assess the performance of both LSTM and RBM models, four distinct metrics were employed to quantify prediction error: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2).

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(T(k) - \dot{T}(k))^2} \quad (19)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (20)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (21)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (22)$$

TABLE 1: Comparison of Performance Metrics Evaluation between LSTM and RBM

Evaluation	Traffic Prediction Models	
	LSTM	RBM
MAE	0.009475	0.019081
MSE	0.000584	0.001253
RMSE	0.024170	0.032781
R^2	0.787276	0.356339

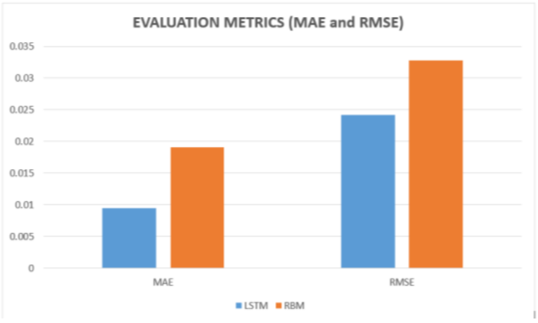

Figure 5: Graphical representation of performance Metrics comparison between LSTM and RBM.(MAE, and RMSE).

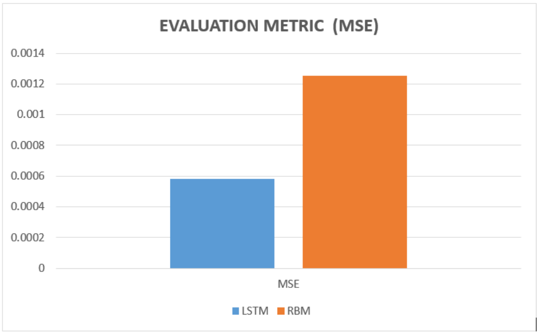

Figure 6: Visual comparison of MSE performance metrics for LSTM and RBM.

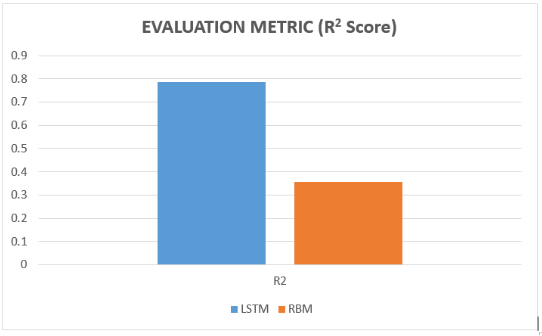

Figure 6: A graph illustrating the comparative R2 Score performance of LSTM and RBM.

Metrics

Test Loss: The LSTM model shows a clear reduction in test loss after optimization, which is crucial for better prediction accuracy. LSTM's optimized Test Loss is one of the lowest among the models, making it a strong performer in terms of minimizing error during training.

Mean Absolute Error (MAE): LSTM achieves a very competitive MAE (0.009115 after optimization).

Mean Squared Error (MSE): The LSTM model exhibits a strong reduction in MSE, and the LSTM model have the ability to capture sequential dependencies effectively Facilitates improved accuracy in predictive modeling.

Root Mean Squared Error (RMSE): LSTM maintains a very low RMSE after optimization (0.024089), close to OGRU. This indicates that LSTM has strong predictive power.

R^2 Score: The LSTM model shows a solid R^2 score of 0.7887 after optimization, which is a very competitive score. This indicates that LSTM is able to explain a significant portion of the variance in the data, making it a highly reliable model.

CONCLUSION

With LSTM having solid performance across all key metrics (Test Loss, MAE, MSE, RMSE, and R^2 score), LSTM proves to be highly capable of handling time-series data or sequence-based tasks. Its ability to model temporal dependencies is beneficial for complex predictions. While RBM showed improvement in R^2 score, it lags behind LSTM in terms of other key metrics, making it a less competitive choice for general prediction tasks. This research can further be optimize using any optimization technique to get more accurate prediction.

REFERENCES

[1] Abdelhadi A. and Guy P. (2017). “A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction”. arXiv:1705.05690v3 [cs.NI] 8.

[2] Agnieszka G., Bartosz S., Aleksandra K. and Krzysztof W. (2022). “Short-Term Network Traffic Prediction with Multilayer Perceptron”. 6th SLAAI International Conference on Artificial Intelligence. DOI: 10.1109/SLAAI-ICAI56923.2022.1000243 pp. 1- 6.

[3] Ahmed A. and Sudhakar G. (2020). “Network Traffic Prediction using Quantile Regression with linear, Tree, and Deep Learning Models”. IEEE 45th Conference on Local Computer Networks (LCN) Pp. 421- 424.

[4] Bi J., Xiang Z., Haitao Y., Jia Z., and Mengchu Z. (2021). “A Hybrid Prediction Method for Realistic Network Traffic with Temporal Convolutional Network and LSTM”. IEEE Transactions on Automation Science and Engineering 2021..

[5] Dahunsi, F. M. et al., (2014), Performance Evaluation and Modeling of Internet Traffic of an Academic Institution: a case study of the Federal University of Technology, Akure. Nigerian Journal of Technological Research Vol. 9 no 2: pg 58-63.

[6] Huaifeng S., Chengsheng P., Li Y., and Xiangxiang G. (2021). "AGG: A Novel Intelligent Network Traffic Prediction Method Based on Joint Attention and GCN-GRU". Security and Communication Networks, Volume 2021, Article ID 7751484, https://doi.org/10.1155/2021/7751484.

[7] Jihong Z., and Xiaoyuan H. (2022). "NTAM-LSTM models of network traffic prediction". MATEC Web of Conferences 355, 02007, https://doi.org/10.1051/matecconf/202235502007.

[8] Jihoon L. (2019). "Prediction of University Network Traffic Using Deep Learning Method", Journal of Information Technology & Software Engineering Vol. 9 Iss. 2 No: 260.

[9] Legend , G. and Taqqu, M.S. (1994) Stable Non-Gaussian Random Processes. Stochastic Models with Infinite Variance. Stochastic Modeling. Chapman & Hall, New York.

[10] Oluwadare et al.,(2019) Network Traffic Analysis Using Queuing Model And Regression Technique Journal of Information 2019 Vol. 5, No.1, pp. 16-26.

[11] Preeti Gulia (2019), Machine Learning and Deep Learning, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-12, October 2019.

[12] Sebastian T., Rodolfo A., Youduo Z., Guido M. and Achille P. (2018). "Deep Learning-based Traffic Prediction for Network Optimization." 2018 DOI: 10.1109/ICTON.2018.8473978.

[13] Shyam Srinivasan, Ralph J. Greenspan, Charles F. Stevens, and Dhruv Grover, 2018, Deep(er) Learning, The Journal of Neuroscience, August 22, 2018 • 38(34):7365–7374 • 7365.

[14] Tain ZD, Li SJ. A network traffic prediction method based on IFS algorithm optimised LSSVM. *Int J Eng Syst Model Simul*. 2017;9(4):200-213.

[15] Tiago P. O., Jamil S. B. and Alexsandro S. S. (2016). "Computer network traffic prediction: a comparison between traditional and deep learning neural networks". *Int. J. Big Data Intelligence, Vol. 3, No. 1, 2016 Pp. 28 -37.*

[16] Wang S., Zhuo Q., Yan H., LI Q., and QI Y. (2019). "A Network Traffic Prediction Method Based on LSTM". Zte Communications Vol. 17 No. 2 Pp. 19 – 25.

[17] Xueyan H., Wei L., and Hua H. (2024). "An intelligent network traffic prediction method based on Butterworth filter and CNN–LSTM". ScienceDirect, Computer Networks, DOI:10.1016/j.comnet.2024.110172.

[18] Yuantao L. (2023). "Deep Learning Network Traffic Prediction based on Bayesian Algorithm Optimization. Highlights in Science, Engineering and Technology" CMLAI 2023.