International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2023: 1.843

Smart Routing: Applying Deep Learning for Efficient Network Traffic Management

Omar Hisham Rasheed Alsadoon

College of Islamic sciences, Al-Iraqia University, Baghdad, Iraq
Email: *omaralsadoon345[at]yahoo.com*
https://orcid.org/0009-0008-1768-8813

Abstract: ***The rapid expansion of modern networks in our current era, enhanced by the proliferation of connected devices with massive applications, has imposed challenges on network traffic management methods. In an expanding dynamic environment, the efficient use of network resources must be ensured while maintaining the reduction of latency, packet loss, and congestion. Deep Learning (DL) has emerged as an effective part of Artificial Intelligence (AI) and a technology capable of addressing the challenges by modeling complex patterns and adapting to network expansion. The proposed method is based on the neural network structure and exploiting feedback and back propagation in controlling the redesign of the neural network by calculating its weights and determining the priorities of extracted parameters. The variables are stored in vectors to be classified and find the best design to suit the network's adaptation to the dynamic expansion. The proposed model adapts to the network conditions and predicts appropriate paths in real-time. The good results prove the worth of the proposed model in terms of accuracy of 97.9% and prediction of 98%. The proposed model performs excellently in real time due to the dynamic ability to controlling the extracted parameters weight. In the future, hybrid algorithms based on machine learning and deep learning can be applied to obtain better results.***

Keywords: Smart routing, Deep learning, Traffic management, Feedback, Neural network.

1. Introduction

Computer networks have become more complex, and the number of applications used by clients and servers has increased, hence the problem of network resource management [1]. There are serious challenges facing network maintenance and design specialists, especially those that consume computer resources, which may lead to a decrease in system performance [2]. Analysis of these resources is no longer useful, especially after the development of information technology, which is considered one of the traditional methods, and decision-making according to these analyses is no longer sufficient in improving networks. Hence, it requires improvement processes of a special kind in network management technology in a way that reduces resource consumption [3]. Regarding computer networks, there is always a need to develop the method of managing them, due to the increasing use of them and the rapid development of information technology. The huge demand for data transmission and the ever-increasing growth in real-time Internet usage are some of the biggest challenges for traditional network traffic management systems [4]. To ensure seamless connectivity, optimal bandwidth utilization, and reduced latency, efficient routing of data packets is of utmost importance. The complexity of traffic patterns and dynamic network conditions make traditional routing methods insufficient to meet today's interconnected requirements.

Smart routing, powered by deep learning, represents a suitable approach to address these challenges, unlike traditional approaches that rely on fixed algorithms [5]. Deep learning enables the network to dynamically adapt to any sudden changing conditions by learning large amounts of data in advance, so that traffic congestion can be predicted in advance and intelligent path selection can be made by optimizing network resources in real time. Deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have proven to be very successful in solving complex problems in networks and other areas, such as natural language processing, image processing, and predictive analytics [6]. Therefore, it is worthwhile to apply them to network traffic to analyze data, predict the best paths, and avoid bottlenecks. Such models help in making future decisions to build a more efficient system, and this is the purpose of choosing deep learning model in this study.

This study focuses on integrating deep learning techniques into network traffic management, and exploring intelligent routing using modern models. The main goal of the research is to reduce latency and packet loss to maximize system utilization and performance. The research also examines the challenges associated with using deep learning in a real-world, scalable, and data-privacy network environment. In an era where digital infrastructure underpins almost every aspect of society, the importance of intelligent, efficient, and adaptive traffic management systems cannot be overstated. In this study, we seek to pave the way for next-generation intelligent networks that are not only robust, but also capable of meeting the ever-evolving demands of users and applications.

2. Related Work

Recently, the traffic flow in networks has evolved significantly, this evolution includes its transition from fixed algorithms to systems that rely on artificial intelligence such as machine learning and deep learning. Previously, the Dijkstra and Bellman-Ford algorithms were basic and good, but with the development and expansion of networks, they became difficult to adapt to changing network conditions [7]. The improvement in adaptability came from the use of dynamic routing protocols such as OSPF and BGP, which suffer from their dependence on pre-defined metrics, which affected their performance when traffic loads fluctuated [8].

Then the emergence of machine learning algorithms and supervised ones such as Support Vector Machines (SVM) and decision trees contributed to the detection of anomalies in traffic when network conditions changed [9]. Such algorithms are very effective, but they often suffer from poor performance when networks expand and new additions to the data path. Reinforcement learning contributed to the process of interacting with networks and working with updating paths, but it suffered from slow convergence in high-dimensional data [10]. Deep learning has played a key role in optimizing network traffic, where CNNs were used to extract spatial features from network paths during training to predict future paths, while another algorithm, represented by RNNs, excelled in extracting temporal dependencies, drawing traffic patterns, and controlling data packets traveling in the network [11]. An algorithm based on transformers was shown to handle complex network situations in terms of congestion and path changes, which resulted in accurate and effective routing decisions, but in a relatively high time compared to other algorithms. These algorithms paved the way for predictive routing systems and adaptive routing according to network dynamics. There are hybrid methods that rely on combining traditional algorithms and deep learning models, thus overcoming some of the limitations that had a negative impact in the past [12]. In this context, the deep learning algorithm, which relies on neural networks managed by weights that change according to the productivity in a single path, has achieved a reduction in the time required to obtain data in the network. It works well in the bandwidth, but is inefficient in solving paths when the network suddenly becomes complex [13]. Despite the development in algorithms that have previously addressed problems, challenges still exist, and the computational complexity of deep learning has imposed some barriers to actual processing. In addition, processing large data raises some concerns about privacy and security in general. Current studies have shown that deep learning helps improve the management of complex networks, as correct prediction leads to a reduction in time through packet arrival time and proactive prediction [14]. Deep Reinforcement Learning (DRL) has contributed to opening up horizons for dynamic adaptation of the algorithm with changing paths and increasing network size, thus surpassing traditional methods that suffer from limitations [15].

This study aims to bridge the gaps that previously plagued approaches by proposing a framework based on deep learning and controlling the feedback and back propagation weights derived from the change in neural network architecture when the network complexity increases and accurately predicting future paths.

3. Network Optimization with Deep Learning

Network traffic efficiency has become a critical challenge in the modern digital environment, in the context of the exponential growth in data transfer and the complexity of infrastructures. Traditional approaches are based primarily on pre-defined heuristics and static network configurations, and struggle to adapt to real-time traffic dynamics and unpredictable network paths. Deep learning offers a transformative solution to address these challenges through its ability to analyze massive amounts of data, discover changing patterns, and thus predict decisions about future paths [16].

Deep Learning in Network Traffic Management

Deep learning is a part of machine learning that uses artificial neural networks to help solve complex problems by automatically recognizing extracted features and patterns. Deep learning can predict congestion in the network, optimize routes, and improve quality of service (QoS).

1) Traffic prediction and load balancing
   Deep learning algorithms such as long short-term memory networks (LSTMs) and recurrent neural networks (RNNs) outperform other algorithms in analysis by analyzing time series data, which increases their ability to predict traffic or bottlenecks to balance loads and reduce congestion.
2) Intelligent packet routing
   Convolutional neural networks (CNNs) can be used effectively to analyze network topologies and data flow through them. This makes routing paths optimal to reduce time and data loss.
3) Providing adaptive QoS
   Deep learning algorithms increase QoS by dynamically adapting the network and prioritizing mobile data that needs precedence, such as video streaming and voice calls.

Benefits of Deep Learning in Network Optimization

1) Real-time adaptability
   One of the advantages of deep learning models is that they are constantly learning, which makes them adapt to changes and network conditions. This reduces network outages.
2) Scalability
   Network expansion leads to its complexity, and complexity increases with the increase in network width. Deep learning is able to handle complexity while maintaining performance.
3) Enhanced security
   Anomaly detection in the network is processed through deep learning to identify irregular network traffic patterns that are indicative of cyber attacks and enhances network resilience.
4) Energy efficiency
   Regulating network traffic leads to energy control by reducing network congestion.

4. Methodology

This study proposes a deep learning-based model to improve network traffic management using feedback control and classification to select hidden layers and nodes in the proposed neural network. This model addresses congestion and the time taken by the packet in the network in addition to suggesting future paths to resolve the bottlenecks. The proposed methodology consists of several main stages starting from data collection and ending with model design and evaluation as shown in the Figure 1.

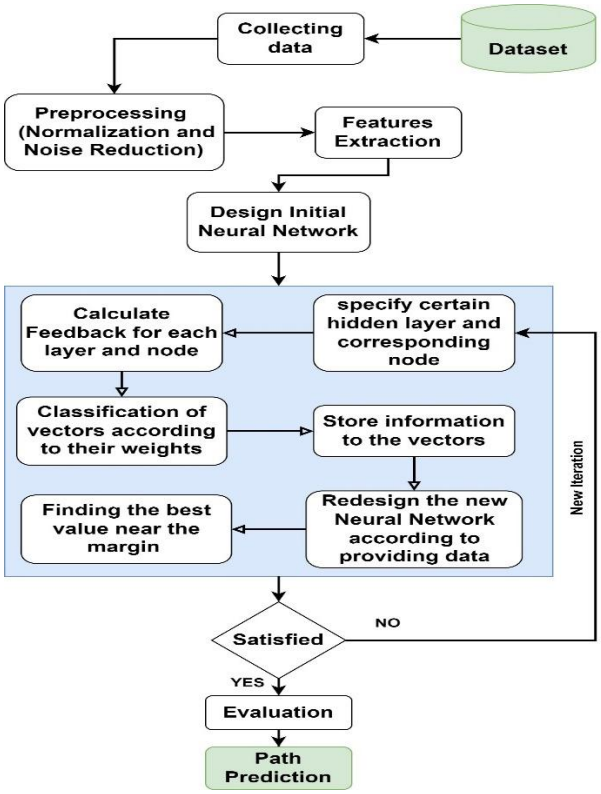

Figure 1: General framwork for proposed method

Data Collection and Pre-Processing: in this context, network data is acquired during the training phase from a standard dataset, including packet loss, bandwidth, latency, and traffic load. Real-world data can be obtained from internet traffic logs or from network simulation data. Much of the data is collected from the dataset to be considered as input by extracting features from it. Data pre-processing is done by cleaning (noise removal) and normalizing to obtain the best features that can be processed later.

The next step involves extracting features from the data, which is essential to feed the deep learning workflow. The work involves isolating and identifying the features that are strongly relevant to the data, have a high impact on the work, and help in prediction. The goal of feature extraction is to reduce the dimensions of the data, and to ignore the duplicate data that is not relevant to improving the result, in order to focus on the most important patterns. As shown in Figure 2.

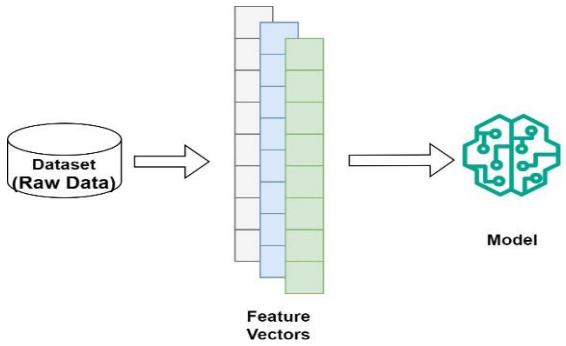

Figure 2: Feature Vectors within Model

In the neural network design phase, the goal is to create a structure capable of making decisions in real time and predicting future paths. For input layer can accept the vector of feature$X = [x_1, x_2, \dots, x_n],\ X \in \mathbb{R}^n$, where *n* consider the input features (latency, bandwidth, congestion level, etc). During the training hidden layer updated according to the sequence of linear transformation and activation function, the neuron constructed within each layer by calculating weight and bias term as the equation bellow:

$$z_j^{(l)} = \sum_{i=1}^{n} w_{ij}^{(l)}\, a_i^{(l-1)} + b_j^{(l)},\ \ j = 1,2,\dots,m \qquad (1)$$

Where l consider as input layer, n_l neuron number within l, $w_{ij}^{(l)}$is weight connecting two layers, $b_j^{(l)}$ is Bias, $z_j^{(l)}$ consider the pre-activation value, and , $a_i^{(l-1)}$ is output of activation function and can find by:

$$a_j^l = f(z_j^{(l)}) \qquad (2)$$

For the first iteration the output layer consist of prediction and decision for routing task, regression task of the output layer computed by:

$$\hat{y} = \sum_{i=1}^{mL} w_i^{(L)} a_i^{(L-1)} + b^{(L)} \qquad (3)$$

In classification task the data go through the hidden layers using softmax activation function as:

$$\hat{y}_j = \frac{\exp(z_j^{(L)})}{\sum_{k=1}^{C} \exp(z_k^{(L)})},\ \ i = 1,2,\dots,C \qquad (4)$$

C consider number of class (represent possible routing).

For each iteration during training the feedback will calculated through $z_j^{(l)}$by controlling the activation functiona_j^l. The back propagation process is then calculated by calculating the error for each hidden layer and the output layer in the neural network.

At this stage, the weights are calculated for the variables that come out of the nodes in the back layer, and the data is stored with the weights in special vectors to be presented to the next stage, which is the classification stage, to show which data can be used and to ignore the data that constitutes few weights. Then the weight vector ***w*** converted into weight matrix *W* of connected layers in dimension of:

$$W \in \mathbb{R}^{n_{outputs} \times n_{inputs}} \qquad (5)$$

Where n_{inputs}neuron number of previous layer, $n_{outputs}$ neurons number of current layer as the following: $\boldsymbol{w}=[w_1, w_2, \dots, w_n]$ and the weight matrix should be

$$W = \begin{matrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_n \end{matrix}$$

Then determine the appropriate weights to update during training of using back propagation and depending on the gradient of the loss function:

$$\Delta w_{ij} = -\delta \frac{\vartheta \gamma}{\vartheta w_{ij}} \qquad (6)$$

Where δ is learning rate, $\frac{\vartheta\gamma}{\vartheta w_{ij}}$ is the gradient of the loss with corresponding weight w_{ij}. And for the next round of neural network the weight updated as:

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \quad (7)$$

At this stage, the contribution of this paper is represented, as controlling feedback and back propagation are among the most important components of the success of deep learning. The contribution lies in extracting the highly influential weights in the feedback to the previous stages of any of the layers of the neural network, which works to change the outcome of the layers in the neural network. All feedback through the layers affects the result in a certain way and to varying degrees. This process is done several times (within iteration) until a result with the highest accuracy that matches the label in the training mode is obtained. As shown in the Figure 3.

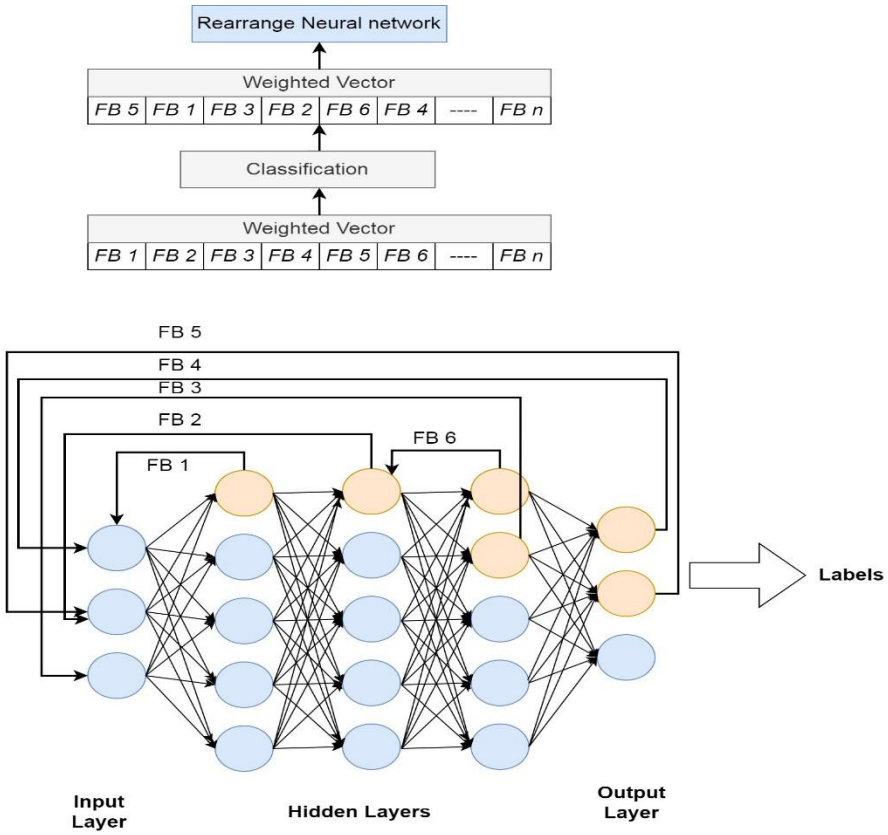

Figure 3: Structure of neural network with proposed method through controlling feedback and back propagation

The process of restructuring the neural networks based on controlling weights that produce the best state for the specific node in the specific layer is considered the basis for the work of the proposed method, which is considered a process of collecting important variables that give the most effective desired result. The work of deep learning here is to identify any element or data that would increase the accuracy of the output and the process of organizing it so that we can make the best prediction. This currency goes through several cycles (iterations) and may repeat itself to previous cycles until it reaches, through the learning process, an accuracy that represents the ideal result. As illustrate in Figure 4.

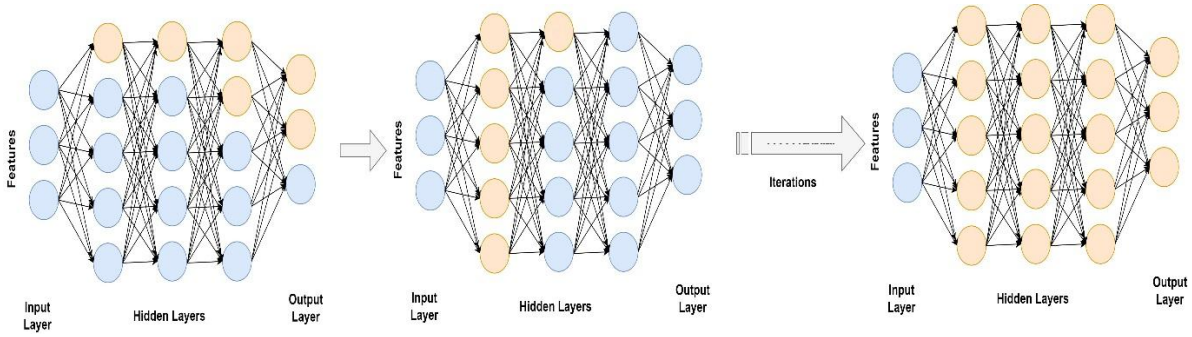

Figure 4: Reconstruction process through iteration of training mode that will repeated until neglect the unusual nodes in all layers

During training, the model improves the weights and the feedback path according to the percentage of matching the prediction with the actual result. Here, the loss function compares the results and updates the neurons until it reaches

the best result. This illustrate the relation between input layer and output layer through hidden layers during processing.

5. Result and Discussion

The practical part of this study is very important, through which the work is evaluated and whether the proposed algorithm can be relied upon. First, we note that the algorithm is trained on real data derived from a standard dataset that contains various types of networks, including particularly complex ones. We divide the original data into two branches, training and testing (Figure 5). Therefore, the training group is also divided into two parts: training and validation. The final division of the data is to verify the validity of the test dataset. The division necessarily has a set of restrictions, and for the test data to be good, we do this procedure. When the test data is large, the data must be verified before starting the test, and hence training depends on the label data, which often contains all the features of the network or any path in it.

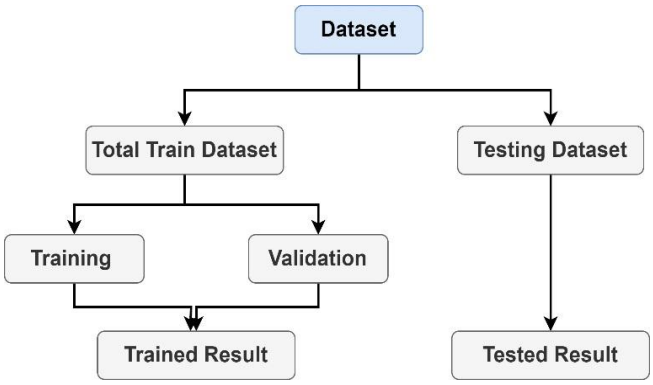

Figure 5: Architecture of dataset

Features are extracted from paths in a given network in the dataset to be classified into vectors for work on them. Each feature has an understanding of the proposed algorithm to work on it.

The data is collected to be interpreted later, and the collection mechanism represents one of the most important processes that must be adhered to. It works based on data derived from the network, such as the access time, the percentage of time estimated to it, the length of the path, and the size of the data. *clusters_model= array ([1,0,4,…,0,0,4], dtypeint=32).*

In classification in general, it is important to determine the performance evaluation, and it is very useful, especially in the difference in classifications during the training phase; it is a measure of the accuracy of the work of the proposed classifier and the extent of its success. The confusion matrix, or as it is called the contingency table, measures the accuracy of the classifier through the columns that represent the predicted and the rows that represent the actual. As shown in Figure 6.

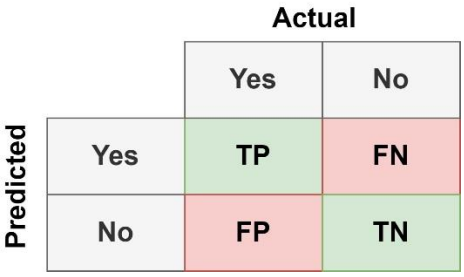

Figure 6: Confusion Matrix representation where (TP) is True Positive, (TN) is True Negative, (FP) is False Positive and (FN) is False Negative

The accuracy can be find by the Equation:

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

Precision tells us what proportion of proper path we detect and have found in the network and can calculated by the Equation:

$$Precition = \frac{TP}{TP+FP} \quad (9)$$

The results can be translated using the confusion matrix to be clearer, more realistic, and compared to the predicted results. As in Figure 7.

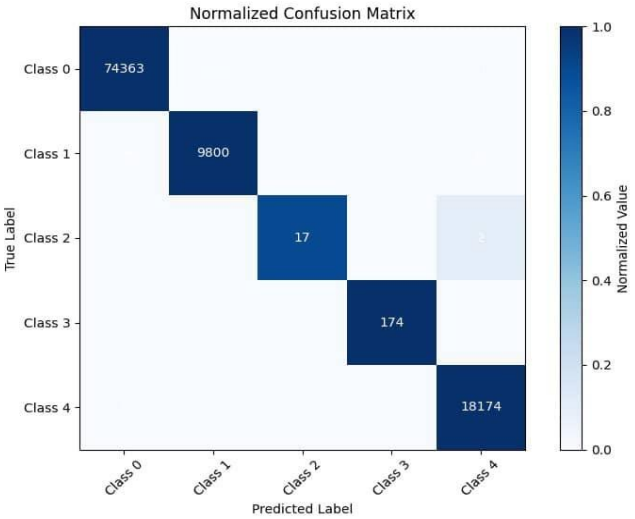

Figure 7: Confusion matrix of proposed algorithm

The test set contains 239,472 samples, which represents 70% of the dataset during training, and the rest 30% will be in testing mode.

The paths in the network are divided into sections to facilitate their calculation, and then the characteristics are searched for when requesting a specific destination to calculate the best path to be preferred over the remaining paths. The short path is not necessarily the best. It may suffer from future bottlenecks. This is the reason for choosing artificial intelligence algorithms to predict future paths and to avoid bottlenecks, especially in complex networks. Table 1 illustrates the evaluation of the network during training mode.

Table 1: Evaluation of network during iteration

Path 0		Path 1	
Accuracy	98.93%	Accuracy	97.90%
Precision	99.95%	Precision	99.94%
Recall	99.96%	Recall	99.94%
F1-score	99.95%	F1-score	99.93%
Path 2		Path 3	
Accuracy	98.97%	Accuracy	98.84%
Precision	100%	Precision	100%
Recall	99.96%	Recall	100%
F1-score	99.94%	F1-score	100%
Path 4		Path 5	
Accuracy	97.95%	Accuracy	98.91%
Precision	99.90%	Precision	99.96%
Recall	99.96%	Recall	100%
F1-score	99.98%	F1-score	99.98%

In Figure 8, we show the training accuracy and overall accuracy of the DL-based model across multiple training epochs to identify the optimal prediction performance. We observe that the accuracy improves significantly, indicating that the system becomes more effective at making predictions as training progresses. This reflects the strong learning advantages of the DL model and its ability to achieve high performance.

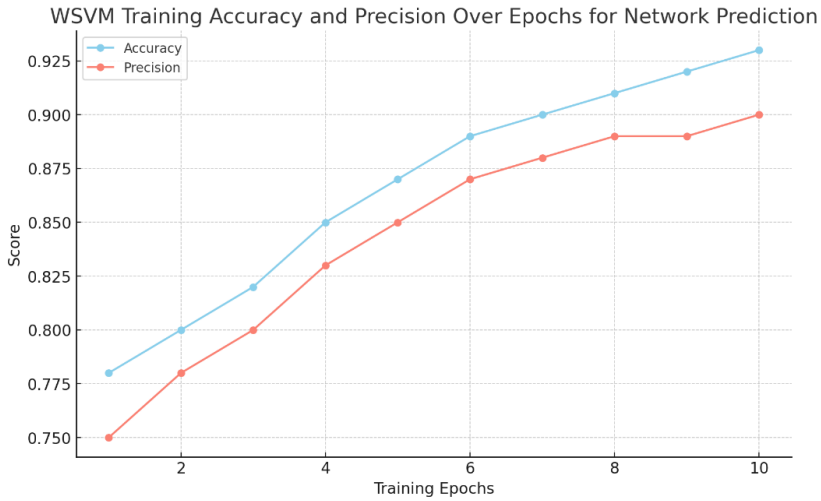

Figure 8: Accuracy and precision through the training

Since the weights vary during the training phase and get better as the amount of time spent on the data increases, we can observe the difference in the improvement of the DL model when comparing the F1 Score with other classifiers. In other classifiers, such as Random Forest or traditional DL, increasing training does not mean increasing the accuracy of prediction, as shown in Figure 9, which represents the difference. This analysis of a complex network with numerous bottlenecks.

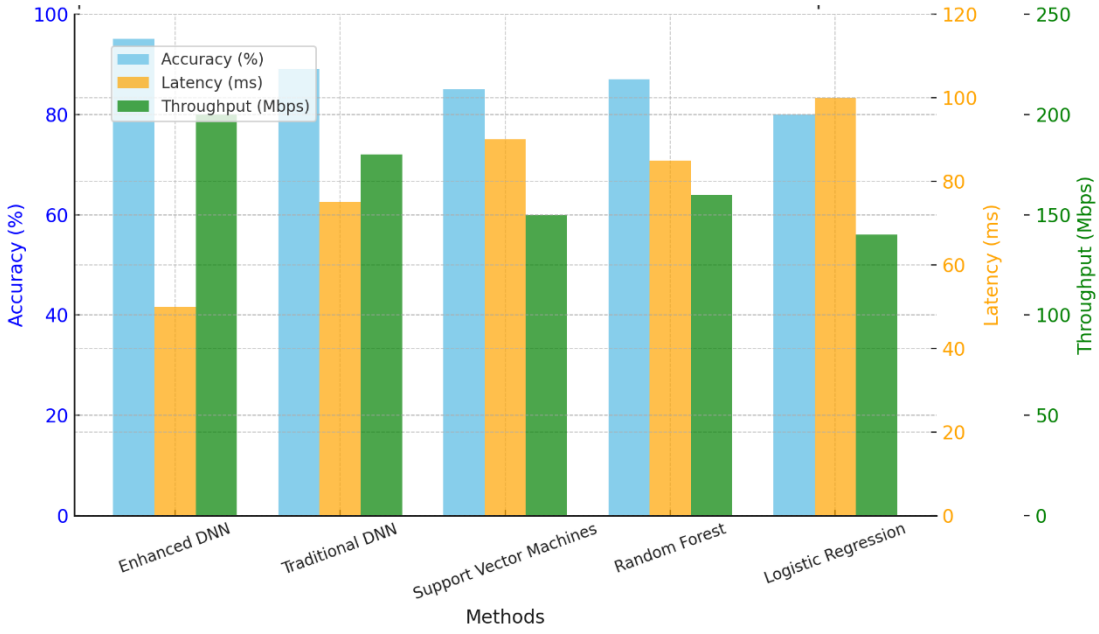

Figure 9: Evaluation of different method in term of Accuracy, Latency, and Throughput

The above analysis is based on extracting features that help controlling feedback to identify each path and choose the best one. The accuracy value reached 93%, outperforming other methods. This is due to adjusting the parameters of feedback and pack propagation of the best features. This performance can contribute to the strength of prediction. Because the DL depends on the fully extracted features and the number of training iterations. The latency performance of the network in terms of access time or bandwidth is of great importance.

When evaluating network behavior prediction and choosing the best path, criteria that we must pay attention to and that play a fundamental role in choosing the appropriate algorithm, and we can summarize it as follows:

Latency (L): which is represent the time taken of packet traveling in the network from source to destination and measured in millisecond. Then we can find as:
$L = T_{propagation} + T_{transmission} + T_{processing} + T_{queuing}$

The nonlinear relationship between network load and response time can increase with increasing peak times and waiting delays, which sometimes makes accuracy less accurate. Also, jitter variance is caused when the response time is not constant, which leads to fluctuations in the accurate prediction. One of the things that contribute to the inaccuracy of the prediction is that the response time from point A to point B is not necessarily the same in the opposite direction from B to A. In addition to the cumulative delay in complex networks with multiple hops, the cumulative time is the sum of the cumulative hops.

Figure 10 illustrate the latency within training in complex network and in congestion time with the same dataset.

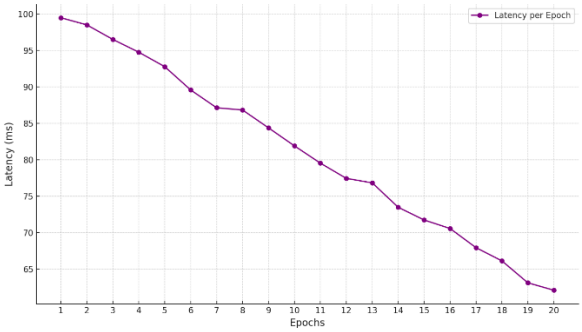

Figure 10: latency in training within proposed method

Bandwidth: represent to the maximum rate of data transfer through network route, and measure as bits /second.
Available bandwidth ($B_{available}$) in the network for certain path can represent as:

$$B_{available} = B_{link} - B_{used}$$

Where B_{link} considers the total link capacity and B_{link} considers the bandwidth used already by the existing path. Available bandwidth depends on the load in the network and real-time issues. For mixed network wireless or Ethernet, bandwidth is not fixed but updated for each prediction path. In multi-link networks, the combination is used as a single virtual path to calculate effective bandwidth, with the possibility of unequal capacities and patterns. The protocol overhead depends on bandwidth usage and packet size and to a lesser extent, on error correction mechanisms. Accuracy depends on overhead and is difficult to quantify.

Figure 11 shows how DL network traffic, using features that are based on identifying normal and attack traffic. Using dynamic weight from feedbacks on extracted features such as packet size and transmission frequency, we can see improved decision boundaries to separate the two classes, which improves prediction accuracy. This shows that some features are better than others at helping to improve prediction accuracy in network traffic patterns.

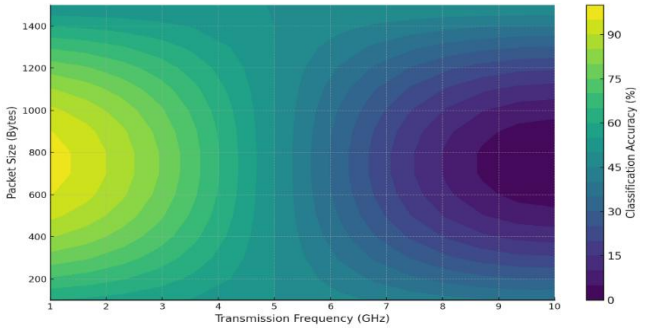

Figure 11: behavior of DL within network congestion

The program was simulated in Python and executed in Jupiter environment for the time- and resource-intensive training process. The training process for a database of 239,472 samples and a size of 1.4 GB took 16 hours (in training). This program and environment is suitable for training online, which saves time and efforts.

6. Conclusion

This study presents a smart approach to predictive modelling of network traffic management using deep learning enhanced with dynamic control of feedback and back propagation with weight adjustments. Unlike traditional DL models, our method assigns variable importance to data points and changes priorities of feedback controlling by the weights to adapt the rapid response to changing network circumstances. The dynamic control of DL parameters to the model deals with network traffic and handles more than one critical condition, resulting in more accurate and responsive predictions. Through comprehensive testing, the smart routing of DL models with dynamic weights have been shown to outperform various types of networks with low false prediction rates, which is evidence of the model's ability to adapt and demonstrate flexibility in operation. Our results, with 97.9% accuracy and 98% prediction based on dynamic feedback weights, indicate that the proposed model plays a crucial role in enhancing the prediction accuracy in traffic management. The proposed approach helps in developing the modeling of variable network performance and diverse data sources and opens the horizons for future work in applying other machine learning and deep learning strategies and benefiting from the predictive capabilities in all fields.

References

[1] Erdelj, M., Król, M., & Natalizio, E. (2017). Wireless sensor networks and multi-UAV systems for natural disaster management. *Computer Networks*, *124*, 72-86.

[2] Favale, T., Soro, F., Trevisan, M., Drago, I., & Mellia, M. (2020). Campus traffic and e-Learning during COVID-19 pandemic. *Computer networks*, *176*, 107290.

[3] Zhou, Z., Abawajy, J., Chowdhury, M., Hu, Z., Li, K., Cheng, H., ... & Li, F. (2018). Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, *86*, 836-850.

[4] Bellavista, P., Fogli, M., Giannelli, C., & Stefanelli, C. (2023). Application-aware network traffic management in mec-integrated industrial environments. *Future Internet*, *15*(2), 42.

[5] Huang, L., Ye, M., Xue, X., Wang, Y., Qiu, H., & Deng, X. (2024). Intelligent routing method based on Dueling DQN reinforcement learning and network traffic state prediction in SDN. *Wireless Networks*, *30*(5), 4507-4525.

[6] Xu, Z., Yuan, J., Yu, L., Wang, G., & Zhu, M. (2024). Machine Learning-Based Traffic Flow Prediction and Intelligent Traffic Management. *International Journal of Computer Science and Information Technology*, *2*(1), 18-27.

[7] Alamoudi, O., & Al-Hashimi, M. (2024). On the Energy Behaviors of the Bellman–Ford and Dijkstra Algorithms: A Detailed Empirical Study. *Journal of Sensor and Actuator Networks*, *13*(5), 67.

[8] Shahid, K., Ahmad, S. N., & Rizvi, S. T. H. (2024). Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems. *Future Internet*, *16*(9), 339.

[9] Dong, S. (2021). Multi class SVM algorithm with active learning for network traffic classification. Expert Systems with Applications, 176, 114885.

[10] Wei, H., Zheng, G., Gayah, V., & Li, Z. (2021). Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, *22*(2), 12-18.

[11] Lumazine, A., Drakos, G., Salvatore, M., Armand, V., Andros, B., Castiglione, R., & Grigorescu, E. (2024). Ransomware detection in network traffic using a hybrid cnn and isolation forest approach.

[12] Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W., & Camtepe, S. (2021). Ae-mlp: A hybrid deep learning approach for ddos detection and classification. *IEEE Access*, *9*, 146810-146821.

[13] Abbasi, M., Shahraki, A., & Taherkordi, A. (2021). Deep learning for network traffic monitoring and analysis (NTMA): A survey. *Computer Communications*, *170*, 19-41.

[14] Yi, T., Chen, X., Zhu, Y., Ge, W., & Han, Z. (2023). Review on the application of deep learning in network attack detection. *Journal of Network and Computer Applications*, *212*, 103580.

[15] Donta, P. K., Srirama, S. N., Amgoth, T., & Annavarapu, C. S. R. (2023). iCoCoA: intelligent congestion control algorithm for CoAP using deep reinforcement learning. *Journal of Ambient Intelligence and Humanized Computing*, *14*(3), 2951-2966.

[16] Wong, M. L., & Arjunan, T. (2024). Real-Time Detection of Network Traffic Anomalies in Big Data Environments Using Deep Learning Models. *Emerging Trends in Machine Intelligence and Big Data*, *16*(1), 1-11.