Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Deep learning model for traffic flow prediction in wireless network

A. K. Kavitha & S. Mary Praveena

To cite this article: A. K. Kavitha & S. Mary Praveena (2023) Deep learning model for traffic flow prediction in wireless network, Automatika, 64:4, 848-857, DOI: 10.1080/00051144.2023.2220203

To link to this article: https://doi.org/10.1080/00051144.2023.2220203

Published online: 21 Jun 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 848–857
https://doi.org/10.1080/00051144.2023.2220203

Deep learning model for traffic flow prediction in wireless network

A. K. Kavitha and S. Mary Praveena

Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore, India

ABSTRACT
In wireless networks, the traffic metrics often play a significant role in forecasting the traffic condition in traffic management systems. The accuracy of prediction in data-driven model gets reduced when it is influenced by non-routing or non-recurring traffic events. The analytical data model used in the proposed method takes into account not only traffic volume and congestion, but also the characteristics of individual applications and user behaviour. This allows for more accurate traffic prediction and better traffic management in wireless networks. The simulation conducted in the paper evaluates the performance of the proposed method in terms of connection success probability and latency. The results show that the proposed method achieves a connection success probability of 93% and a latency of less than 2 ms, demonstrating its effectiveness in improving traffic prediction and management in wireless networks.

ARTICLE HISTORY
Received 15 March 2023
Accepted 26 May 2023

KEYWORDS
Quality of services; bit error rate; deep learning; wireless network

1. Introduction

Over the past few years, NTMA (Network Traffic Management using Artificial Intelligence) has garnered a great deal of attention due to its central role in the development of networking performance. NTMA approaches are both employed by industry and academia in network management [1]. New networking technologies and concepts complicate the creation of efficient networks. In networks where the number of nodes is in the thousands, such as the Internet of Things (IoT), regular monitoring is required to ensure performance [2,3]. Active approaches include the use of probes to place and later remove probe traffic to identify the state of the network. When we inject test traffic into the network and then analyse various network performance indicators, we are saying that these tests are more precisely known as test traffic injection.

NTMA, or Network Traffic Management using Artificial Intelligence, is a technique that employs machine learning algorithms to enhance network traffic management and boost network performance. This method is becoming increasingly popular as a result of the surge in network traffic and the need for more effective network management solutions. NTMA strategies entail gathering network traffic data, analysing it using machine learning algorithms to recognize patterns and trends, and employing the knowledge gained to make informed decisions about network traffic management. These decisions may involve load balancing, traffic prioritization, congestion control, and network routing optimization, among other things. Overall, NTMA is a promising technique that employs machine learning to enhance network traffic management and improve network performance. As the demand for more efficient and reliable network management solutions continues to rise, NTMA methods are anticipated to play a more important role in the networking sector.

The primary way to control Service Level Agreements (SLAs) for services is through active monitoring techniques [4–6]. Passive methods are more often utilized to evaluate and monitor real network traffic in the network, whereas active methods are typically employed to study theoretical network traffic. A great deal of interest has been generated within the industry for management and planning purposes for passive techniques [7,8]. Passive techniques don't need to have another site in the network in order to take advantage of them. This kind of traffic monitoring can be utilized for cases like post-incident scenarios [9].

There are distinct everyday difficulties for NTMA because of the number of people who utilize these communication systems and networks, as well as the amount of generated traffic. Increasing the number of linked nodes and the volume of data raises the complexity of the network, which needs continued monitoring and research to maintain network performance [10]. More importantly, the large and diverse traffic dataset mandates the use of novel monitoring and analysis methodologies for network management data [11]. Many NTMA-focused works focus on a single NTMA process, such as anomaly detection, traffic classification, or quality of service (QoS) [12].

Problem Definition: Traffic data collecting creates major difficulties for NTMA, especially when it comes

CONTACT A. K. Kavitha akkavitha081@gmail.com, kavi4mtamil@gmail.com Sri Ramakrishna Institute of Technology, WWR2+M8R, Perur Chettipalayam, Pachapalayam, Coimbatore 641010, Tamil Nadu

to measuring critical network parameters that must be tested regularly over time. To replicate network traffic, an active probe initiates the activity. Then, to test the end-to-end performance, the probe emits simulated network traffic throughout the network. Passive probes give a separate vision of the network compared to active probes. Links in the network have passive probes that monitor all the traffic that passes through the link.

One way to describe the diverse network requirements and traffic data gathering purposes is to describe the distinct objectives of traffic data collection. To put it another way, in a traffic acquisition task, it is not needed to capture all the accessible data from a network. This also means that network packets are the fundamental pieces of data that need to be analysed in the task of gathering traffic data. Probes are capable of utilizing both of these strategies while gathering network information, but Deep Packet Inception (DPI) has a few drawbacks, the most significant of which is that: The results of user data analysis could compromise the privacy of the users. It takes more time and resources to process the entire packet than the header.

Stateful Packet Inspection (SPI) is utilized in most modern NTMA probes due to the difficulties previously described when utilizing DPI [13–15]. Next, we describe the issues that NTMA faces in obtaining a large amount of trustworthy traffic data. Some useful tools to meet this difficulty, such as [16] and [17], have been developed throughout the years for data collecting and data collection. Though a ubiquitous and efficient data collection strategy that can be employed in big and heterogeneous modern networks is still unavailable [18].

Network packets are the most common method of data traffic gathering for networks. Also, the above methods are inadequate when used with high-speed networks, and therefore rendered ineffectual. Another prominent data collection mechanism is flow-based data gathering. In Gigabit networks, flow-based data gathering approaches outperform packet-based data gathering mechanisms. Despite this, packet and flow filtering make things more difficult for these approaches. In the academic literature, several surveys have been published, such as [19–21].

Because of the emergence of new phenomena known as big data, modern networking technologies are being put under strain. One of the primary consumers of large amounts of data is the NTMA approach [22]. Once you have raw traffic data, NTMA approaches should execute a series of processes to extract meaningful information. There are several obstacles and issues with typical big data analytics methodologies, including accuracy, rapid analytics, and real-time processing of massive data. The more connected devices there are, the raw data they generate every day, and thus we will need better approaches to monitor and analyse this data.

In this paper, a ResNet and DenseNet models are used to learn the network traffic and predict the future network traffic. Both ResNet and DenseNet models utilize traffic parameters from different scenarios to study the traffic condition.

The main contribution of the paper involves the following:

- The author contributed to the development of analysing the network traffic using Residual neural network architecture and dense neural network architecture based on several parameters associated with network traffic.
- The parameters involving traffic scheduling and routing enables the users to accommodate the network condition without lapses in communication.
- The combination of ResNet and DenseNet is designed as a flow-based approach that provides outstanding classification of traffic instances based on the condition of the network.

This research work discusses the challenges in traffic data collection and management for wireless networks. The paper proposes a deep learning analytical model that can learn from network traffic and provide accurate traffic predictions. The simulation results show that the proposed method achieves high success rates and reduced latency. The paper highlights the importance of active and passive monitoring techniques for maintaining network performance and service level agreements. The paper also discusses the challenges in obtaining trustworthy traffic data and the limitations of current data gathering methods. Finally, the paper emphasizes the role of big data analytics in network management and the obstacles and issues associated with it.

2. Related works

The current work is the only one we know of that explores the interaction between NTMA and DL and highlights the use of DL models in NTMA [23]. Several research papers exist in the literature that investigates the usage of data mining techniques and classical machine learning models.

The research findings given in [24] examined various DL-based traffic classification models. However, the application did not include any other NTMA applications for evaluation, as discussed in the current research.

In [25], the authors performed an in-depth review of the common mathematical models used in malware analysis. The challenges and issues associated with this subject are further examined in this study. Additionally, the researchers did not study the critical role of DL in malware identification and analysis.

This study on network traffic analysis [26] is quite in-depth. They use three classification systems to assign relevant works to three classifications. Several algorithms, such as C4.5 decision tree, Naive Bayes, k-means, and Random forest, were tested, analysed, and considered. The focus is on mobile devices and includes discussion on methods of analysis, validation, and outcomes obtained. Furthermore, the focus of was on conventional ML methods, which are exclusively based on Machine Learning, whereas our study focuses on DL models. The authors in [27] applied a big data technique to the National Transportation and Safety Board (NTSA). Using big data methods, the authors studied works that use network traffic data. They also described machine learning ability to handle four NTMA-oriented tasks: classification, prediction, management, and security. It also collected real-time massive IoT data for their study. The current network data analytics methodologies used in real-time IoT networks are examined in this paper. The fundamentals of real-time IoT analytics, applications, and platforms are also explored in that study. One of the major challenges associated with using deep learning (DL) models for traffic classification is the requirement for a large amount of labelled data for training. DL models often require a significant amount of training data to generalize well and make accurate predictions, but collecting and labelling data for traffic classification can be a time-consuming and costly process. In addition, traffic patterns and behaviours can change rapidly, requiring the constant updating and retraining of models with new labelled data. Another challenge is the class imbalance problem, where some traffic classes may be significantly underrepresented in the labelled data. This can make it difficult for the DL model to learn to classify these classes accurately, leading to biases in the model's predictions. Finally, DL models can be computationally expensive and require significant hardware resources for training and inference, making them less accessible for smaller organizations or those with limited resources. That paper did not explore DL models for data analytics; rather, it studied DL models for information retrieval. Most of the methods considered above focus on the network infrastructure and laying new routes for optimal traffic delivery. However, these method takes less network parameters into consideration for optimal traffic classification and management.

3. Proposed model

The ResNet and DenseNet look at wireless network traffic parameters, as well as the routing and traffic scheduling. To accommodate the needs of primary and secondary users, as well as the handover rate, the study takes into consideration the locale of the primary and secondary users. The transport planning and optimization of network traffic are precisely suited to deep learning models.

3.1. Flow-based ResNet and DenseNet

The classification technique handles both encrypted and normal traffic, a flow-based classifier is the most suitable model for the task. This study uses ResNet and DenseNet for traffic classification, a flow-based approach. While the DL (Deep Learning) model accuracy is outstanding, a substantial amount of labelled data is required for modelling. Figure 1 shows the flow-based DL model.

This paper examines the significance of utilizing Artificial Intelligence (AI) in Network Traffic Management (NTMA) to ensure optimal performance in large and intricate networks like the Internet of Things (IoT). Active and passive approaches are utilized for monitoring network performance, with passive techniques being more commonly employed for management and planning. However, traffic data collection for NTMA poses challenges, particularly in measuring essential network parameters that require regular testing over time. While Deep Packet Inception (DPI) is available, it is not commonly used due to privacy concerns and resource requirements. On the other hand, Stateful Packet Inspection (SPI) is utilized in most modern NTMA probes, and flow-based data gathering mechanisms are more effective than packet-based ones in Gigabit networks. To analyse network traffic and predict future traffic, this paper suggests utilizing ResNet and DenseNet models based on different parameters linked to network traffic. The paper's primary contribution is the development of Residual neural network architecture and dense neural network architecture for analysing network traffic using various traffic parameters.

Recurrent Neural Networks (RNNs) have been widely used in network traffic analysis due to their ability to model sequential data. In the proposed model, RNNs can be used to capture the temporal dependencies of network traffic data over time. By processing the data in a sequence, RNNs can learn the patterns and trends of network traffic and use them to make predictions about future traffic. This can be especially useful in predicting network congestion and identifying potential network failures before they occur.

3.2. Mathematical model

The complex channel coefficient $h_{u,k}$ represents the communication quality between the User Equipment (UE) and eNodeB over Resource Block (RB) at each Transmission Time Interval (TTI). It considers various propagation effects over the LTE channel, including shadowing, path loss, and small-scale fading. The system has good antenna gains and a coherence channel bandwidth larger than the RB bandwidth, leading to flat fading and a complex channel response denoted by the complex channel coefficient. This coefficient is mainly

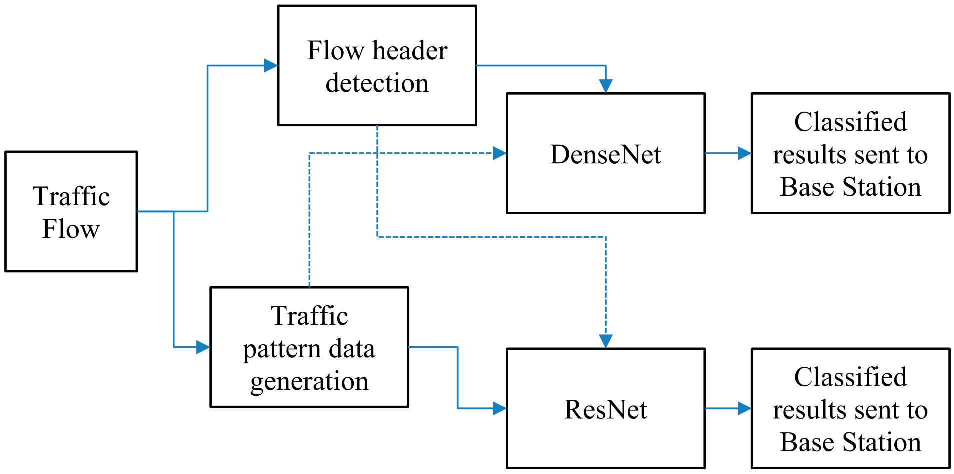

Figure 1. Flow-based DL model.

associated with the subcarrier (middle range) and the first OFDM symbol. The $h_{u,k}$ is estimated using UE through pilot symbols and gets transmitted using eNB. Then the channel estimated is $\hat{h}_{u,k}$, which is modelled as:

$$\hat{h}_{u,k} = \sqrt{(1-\xi)h_{u,k}} + \sqrt{\xi_\eta} \tag{1}$$

where the channel estimation degradation is $\xi \in (0, 1)$ and channel estimation error is $\eta \in C$. This is considered as the random variable, which is called as Zero Mean Circularly Symmetric Complex Gaussian (ZMCSCG), with $E\{|\eta|^2\} = E\{|h_{u,k}|^2\}$.

The present study finds the CSI imperfections associated with the channel estimation error and the parameter ξ is evaluated based on its impact and finally the reports are considered at each TTI and without any delay the eNB acquires the measures.

Further, the instantaneous SNR $\hat{\gamma}_{u,k}$ estimated with individual UE u on k during each TTI is calculated as:

$$\hat{\gamma}_{u,k} = \frac{p_{u,k}|\hat{h}_{u,k}|^2}{\sigma^2} \tag{2}$$

where $p_{u,k}$ represents the power of eNB used for transmitting the UE u over k; σ^2 is the average AWGN power.

Similar to LTE, the link adaptation mechanism is used to select the MCS (m) using eNB from the set of MCSs (M).

The MCS selection uses $\hat{\gamma}_{u,k}$ and considers $M = |M|$ with varying MCS, where $|\cdot|$ defines the cardinality, when applied over the set. The MCS $m_{u,k}$ related to the UE u in k is calculated as:

$$m_{u,k} = f(\hat{\gamma}_{u,k}) \tag{3}$$

where $f\ (\hat{\gamma}_{u,k})$ represents the function of link adaptation.

The eNB in this work selects the better MCS over given UE u on k leading to the maximized data rate for power allocated in the network.

The required value of block error rate is used to acquire the link adaptation curve with minimum SNR, $\hat{\gamma}_{u,k,m}$ and the eNB transmit information to UE u in k using m^{th} MCS and this guarantees the block error rate value. The minimum SNR is calculated as:

$$\hat{\gamma}_{u,k,m} = f^{-1}(m_{u,k}) \tag{4}$$

where $f^{-1}(\cdot)$ defines the inverse function of link adaptation.

The $\hat{\gamma}_{u,k,m}$ value obtain the $p_{u,k,m}$ transmit power (minimum) linked with MCS m. The flat fading is considered over the RB during each TTI and the value $\hat{h}_{u,k}$ is made constant for each $\{u,k\}$ pair.

The throughput $r_{u,k,m}$ of UE u over k through MCS m is given as,

$$R_u = \sum_{k=1}^{K=|K|} \sum_{m=1}^{M=|M|} r_{u,k,m} x_{u,k,m} \tag{5}$$

where $x_{u,k,m}$ represents the assignment index allocation of RB in UEs.

Finally, the QoE τ_u of the desired UE u is acquired from rate R_u through the function $\phi(\cdot)$ that maps into MOS, as

$$\tau_u = \phi(R_u) \tag{6}$$

3.3. ResNet

These networks primarily focus on using an identification shortcut that misses at least one or more levels. This block is referred to as the Residual Block, as may be seen in the diagram below. The output from the previous layer is only used in the next layer. However, for some values of x, $F(x)$ does not have the same dimensions. With regard to spatial resolution, a 32×32 convolution will result in an image with 30×30 dimensions.

Increased identity mapping occurs because the shortcut channels map to the residuals, allowing the identity to be expanded. Thus, the two additional input

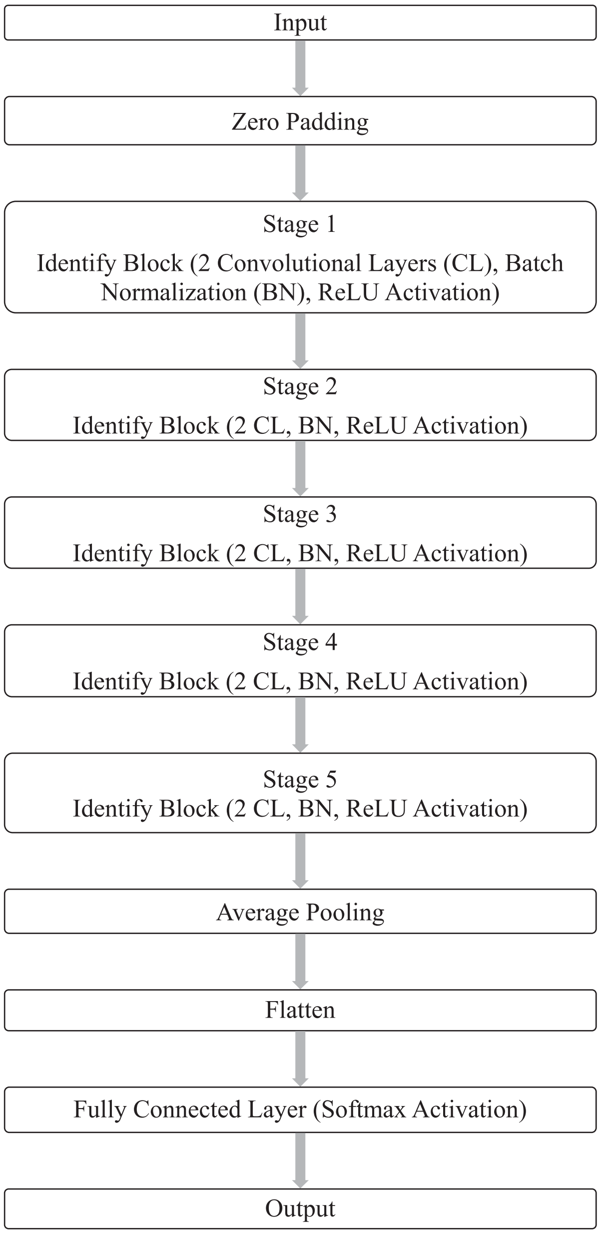

Figure 2. ResNet model.

variables x and $F(x)$ are introduced to the subsequent level as inputs. Connections in the Skip Layer group are used to connect the outputs of previous layers to the stacked layer outputs. What this means is that networks can create far deeper relationships than before.

This ResNet 50 employs a 5-stage model consisting of a convolution and an identity block. On each level of the convolution, three layers are contained, and for each degree of identity, three layers are included as in Figure 2.

The model can be built with two fully connected layers, which include a SoftMax activation and a 50% activation drop. Even if no sound file is present, the user can input the binary value once the model is established. The parameters used for this example include a learning rate of 0.001, a batch size of 32, binary cross-entropy as the loss function, and the Adam optimizer as the optimization function.

Hence, to resolve the optimization problem, the study uses the following objective function:

$$f_F^* = \arg\min_f L(X, y, f) s.t f \in F \tag{7}$$

where X is the features from the input dataset; y is the label; F is the class of function f.

3.4. DenseNet

DenseNet-169 + is a type of dense network that uses dense links to connect all layers directly, from layers to dense blocks. This helps to maintain the classification characteristics of the feature maps by allowing each subsequent layer to receive inputs from all the earlier layers and pass on its feature maps to the subsequent levels. Consider a single image x_0 via the initial layer of the network i.e. convolutional network. The DenseNet-169 + network is designed with L layers or strata, where each layer adopts a non-linear $H_l(\cdot)$ transformation, in which the l layers are considered as indexed. The composite function $H_l(\cdot)$ of such operations including convolution, pooling, rectified linear units and batch normalization. The study indicate "l^{th} layer" output as x_l. This DenseNet employs a 5-stage model consisting of a convolution and an pooling layer which is shown in Figure 3.

3.4.1. ResNets

The l^{th} layer output act as an input to the consecutive layer i.e. $(l+1)^{th}$ layer is connected through a traditional convolutional feeding network, which results in transition on a layer:

$$x_l = H_l(x_{l-1}) \tag{8}$$

ResNets further adds a skip connection using a following identity function to bypass the operations of non-linear transformations:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{9}$$

ResNets have the advantage that the gradient can directly pass from conventional layers to previous layers via identity function. The output of H_l however hinders the data flow in the network.

3.4.2. Dense connectivity

The study proposes a distinct connectivity pattern to further improve the flow of data between the layers i.e. from any layer and to improve the dense learning, the study introduce direct connections to all the following layers. Figure 2 shows the layout schematically of the DenseNet.

3.4.3. Composite function

The study defines $H_l(\cdot)$ as a three-fold composing function that includes batch normalization, rectified linear unit and 3 × 3 Conv.

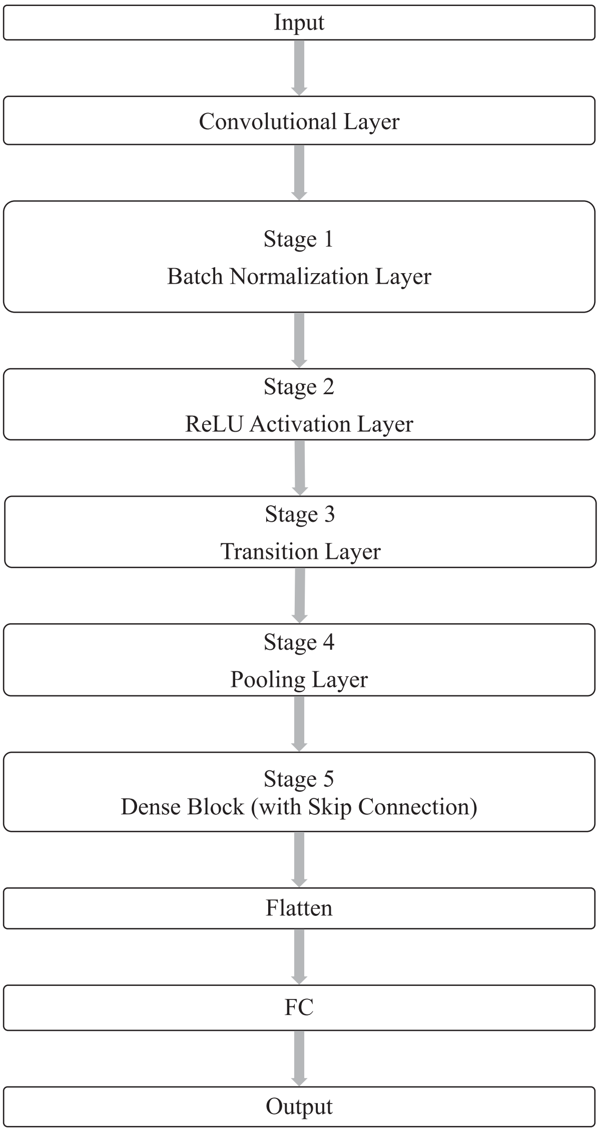

Figure 3. DesNet model.

3.4.4. Pooling layers

Concatenation in Equation (3) is based on the feature map size. Nevertheless, the down sampling layers that change the feature map size that are required for dense networks are used in the network as an essential element. Down-sampling is simplified in networks where the layers are closely connected, with the results of the down sampling operation contained in those blocks. The study uses the term transitional layers to refer to block layers that come together and pool.

3.4.5. Growth rate

The number of feature maps provided by each H_l function in DenseNet-169 + tends to be k, where the l^{th} layer with $k_0 + k \times (l-1)$ feature-maps with k_0 channels in its input layer. The network growth rate hyperparameter is referred to as k. The study shows a reduced growth rate is enough to achieve the most advanced data sets on which the study is tested.

Table 1. Simulation parameters.

Parameters	Value
System Bandwidth	20 MHz
Subcarrier per Resource Block	12
Subcarrier spacing	15 kHz
Total subcarriers	1200
Total Resource Block	100
FFT Size	2048
Modulation	64QAM
Total Macrocells	7
Inter-cell distance	500 m
Carrier Frequency	2.1 GHz
AWGN power per sub-carrier	−123.24 dBm
Users speed	1 m/s
TTI	1 ms
eNB antenna radiation pattern	Three-sectored
Subcarriers per RB	12
Number of antenna	1 Transmitter and 1 Receiver

3.4.6. Bottleneck layers

Although the output characteristics of each layer are k only, typically there are many more entries. It is observed that before each three to three convolutions, the 1×1 convolution could be introduced as a bottleneck layer to reduce the number and thus increase the computational efficiency of inputs.

4. Results and discussion

The ResNet and DenseNet algorithms are implemented in MATLAB, and are present in the MATLAB environment. In the simulation, a map with an area of 2000 m × 2000 m grid pattern is considered. Network environments have nodes with a speed between 10 and 40 km/h. When various characteristics, such as vehicle speed, traffic type, and network density, are used, the DenseNet and ResNet performance is compared to that of the RNN. The parameters for simulation are given in Table 1:

In Figure 4, the overall network performance varies depending on how many vehicles are arriving in different transmission ranges with varied vehicle arrival rates. The conclusion drawn from this experiment is that as the average distance grows, the connectedness will decrease. When the average distance is 50 m, the likelihood of connecting goes to zero. Therefore, it is proven that as the traffic density grows, network connectivity improves to within 50 m of the original position.

The cumulative distribution function (CDF) among DenseNet, ResNet, and existing RNN models, along with the likelihood of connection among 100 nodes, 200 nodes, and 300 nodes, is demonstrated in Figure 5–7. Using the DenseNet and ResNet technique, the algorithm is tested for several mobility conditions. The density of mobile nodes, which means the likelihood of frequent network connections, is larger than 0.9. When comparing morning, noon, and night time connection rates with other methods, the successful connection rate was determined to be 77%, 88%, and 93%.

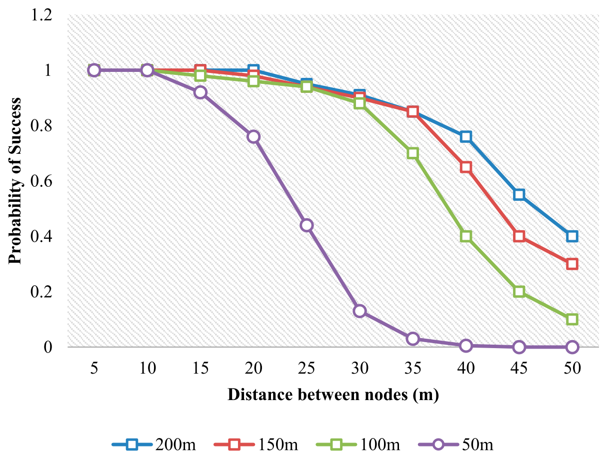

Figure 4. Network connectivity.

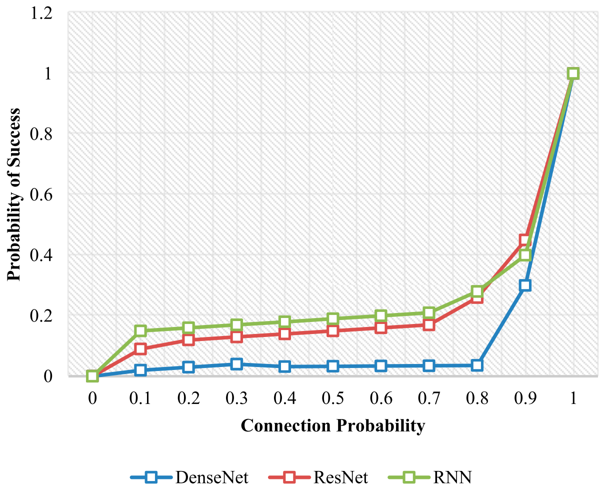

Figure 5. Successful connection probability with 100 nodes.

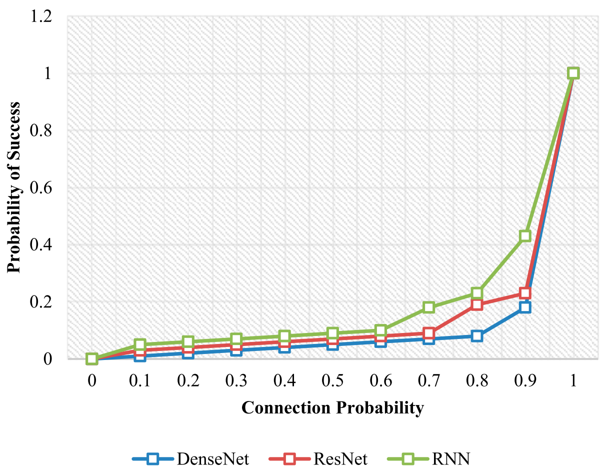

Figure 6. Successful connection probability with 200 nodes.

According to Figure 8, the DenseNet and ResNet approach is contrasted with vehicle velocity. When vehicle speed increases, average latency increases. This paper varies the total number of nodes between 100 and 500. Higher densities of vehicles in the network cause an increase in average delay.

Due to the relay vehicle optimal selection of forwarders, the higher average delay is present. In

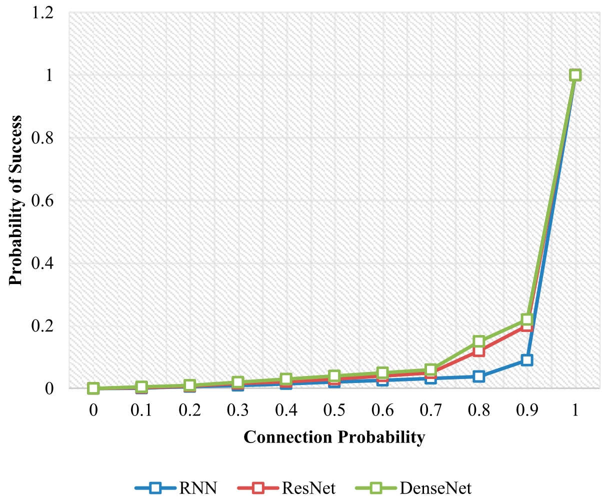

Figure 7. Successful connection probability with 300 nodes.

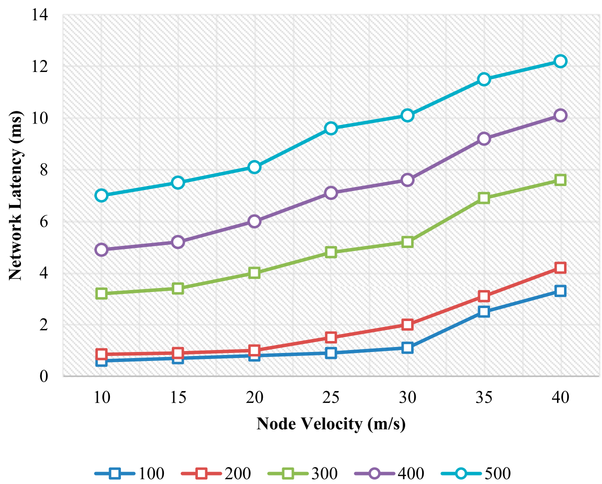

Figure 8. Average latency.

DenseNet and ResNet training, nodes with the lowest processing speed are assigned to increase the probability of packets reaching their destinations. Because of this, lower vehicle density has lower delay, and vice versa. This is what was discovered: Increasing velocity increases the packet delivery latency.

With variable vehicle density (Figure 9), ResNet and DenseNet offer lower average end-to-end delay. Packet delivery ratio strongly influences communication delays. The rise in traffic density results in growing data packet redundancy. Since the delay minimum and maximum values are calculated based on the node density of 50 and 300, respectively, As previously mentioned, the ResNet training methods outperform the RNN technique.

ResNet and DenseNet hops received are lower than those of other approaches in Figure 10. When the number of nodes is high, the lowest hop count is recorded in the RNN. On the other side, when the node density is reduced, the RNN will hop over more connections. Specifically, along the periphery, the density is often high, thus we record few hops there.

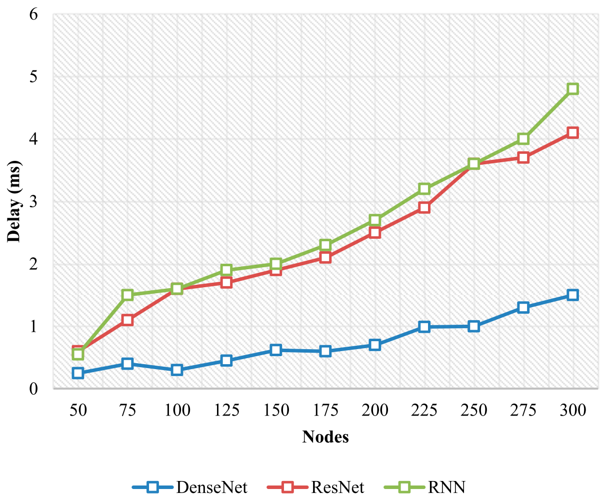

Figure 9. Average end-to-end delay.

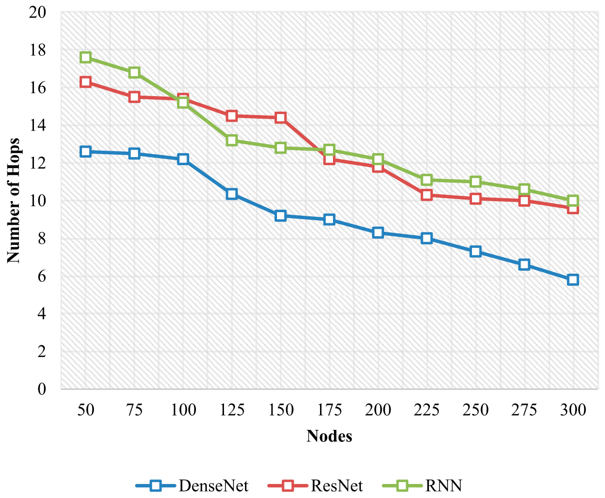

Figure 10. Reduced hop with varying nodes.

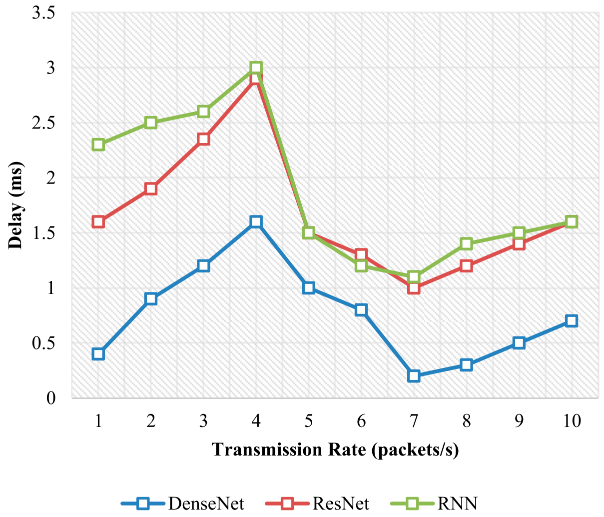

Figure 11. Average end-to-end delay with varying transmission rate.

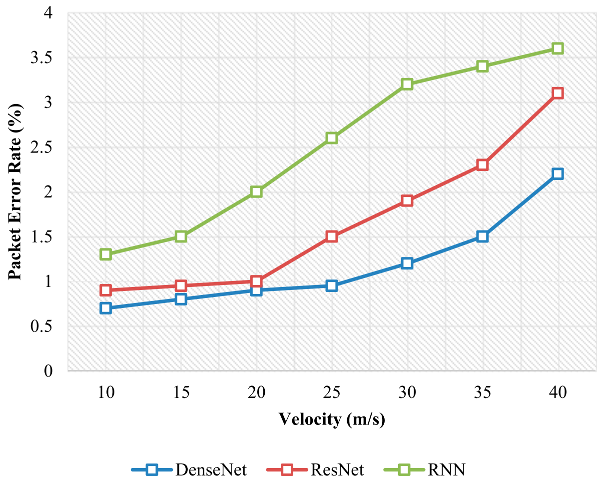

Figure 12. Packet error rate.

It is assessed against the various previous techniques in terms of average end-to-end latency (Figure 11). This experiment shows that RNN technology reduces delay while other technologies do not. However, the ever-growing number of nodes in the network contributes to increasing the time it takes for a two-to-six packet data rate to propagate. The DenseNet and ResNet lowest delay is calculated to be 7 packets per second.

The ResNet and DenseNet approaches reduce packet errors as the velocity increases, as shown in Figure 12. In terms of its capabilities, however, the ResNet and DenseNet perform better than the RNN. RNN gets the highest error rates while maintaining near proximity to ResNet and DenseNet. This is beneficial for increasing the overall sustainability of vehicle connectivity in neighbourhoods because vehicles can be driven while using the Internet. Collisions of data packets are mostly responsible for the losses. When speeding, vehicle performance may decrease because of higher packet loss or error rate, which is why the RNN algorithm prefers vehicles that are less fast.

The data delivery ratio between RNN and conventional approaches is shown in Figure 13. The data delivery ratio increased for RNN, whereas velocity decreased with increasing velocities for all techniques. Vehicles moving at a higher speed have a lower delivery ratio, whereas vehicles travelling at a lower speed have a higher delivery ratio. Packet loss increases as the vehicle's velocity increases. There are more collisions in the network now, and this causes the data transfer to drop to the bare minimum.

According to the results of Figure 13, ResNet and DenseNet have better PDR and throughput than RNN. There is no consistency in the data rates for both metrics, and they vary depending on the network density, with the highest throughput typically occurring at the base station and the lowest at the network edges. The packet delivery ratio is improved by using high-quality segments with appropriate forwarding paths for data. Selecting the next hop can increase the number of packets sent to the destination node while decreasing the packet error rate. To ensure reliability, it is important to

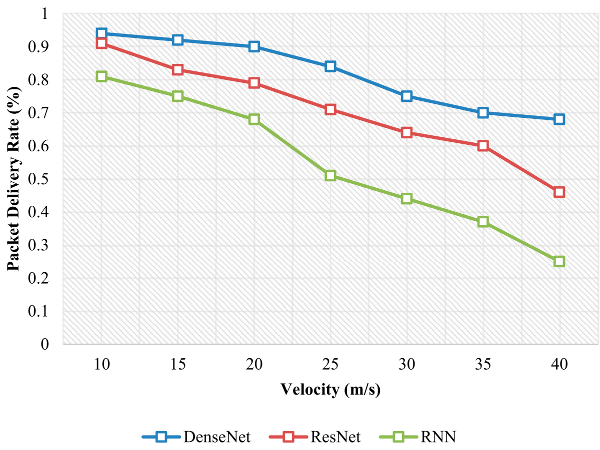

Figure 13. Packet delivery ratio.

monitor the packet delivery rate. However, increasing the packet rate may lead to decreased performance.

5. Conclusion

In this paper, a deep learning analytical model that reads the entire network traffic is developed that learns and provides prediction of future network traffic. The deep learning model utilizes traffic parameters from different scenarios to study the traffic condition. The traffic analysis in wireless networks is investigated in terms of routing and traffic scheduling that includes queuing, prioritization and network capacity. The study takes into account the location of primary and secondary users and the handover rate. The deep learning model is designed for transport planning and optimization of network traffic in such environment. The results of the simulation show that the proposed method achieves 93% success probability.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] D'Alconzo A, Drago I, Morichetta A, et al. A survey on big data for network traffic monitoring and analysis. IEEE Trans Netw Serv Manage. 2019;16(3):800–813.
[2] So-In C. A survey of network traffic monitoring and analysis tools. CSE 576 m Computer System Analysis Project, Washington University in St. Louis. 2009.
[3] Cecil A. A summary of network traffic monitoring and analysis techniques. Comput Syst Anal. 2006:4–7.
[4] Sun P, Aljeri N, Boukerche A. Machine learning-based models for real-time traffic flow prediction in vehicular networks. IEEE Network. 2020;34(3):178–185.
[5] Miglani A, Kumar N. Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions, and challenges. Veh Commun. 2019;20:100184–100198.
[6] Wang W, Bai Y, Yu C, et al. A network traffic flow prediction with deep learning approach for large-scale metropolitan area network. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, April. Taipei. IEEE; 2018. p. 1–9.
[7] Han SH, Kim MS, Ju HT, et al. The architecture of NG-MON: a passive network monitoring system for high-speed IP networks. In International Workshop on Distributed Systems: Operations and Management, October. Berlin: Springer; 2002. p. 16–27.
[8] Ehrlich M, Biendarra A, Trsek H, et al. Passive flow monitoring of hybrid network connections regarding quality of service parameters for the industrial automation. 8. Jahreskolloquium "Kommunikation in der Automation–KommA". 2017.
[9] Boutaba R, Salahuddin MA, Limam N, et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J Internet Serv Appl. 2018;9(1):1–99.
[10] Labrinidis A, Jagadish HV. Challenges and opportunities with big data. Proc VLDB Endow. 2012;5(12):2032–2033.
[11] Sivarajah U, Kamal MM, Irani Z, et al. Critical analysis of big data challenges and analytical methods. J Bus Res. 2017;70:263–286.
[12] Kandasamy M, Yuvaraj N, Raja RA, et al. QoS design using Mmwave backhaul solution for utilising underutilised 5G bandwidth in GHz transmission. In 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), February, Coimbatore, India. IEEE; 2023. p. 1615–1620.
[13] Indhumathi R, Amuthabala K, Kiruthiga G, et al. Design of task scheduling and fault tolerance mechanism based on GWO algorithm for attaining better QoS in cloud system. Wirel Pers Commun. 2023;128(4):2811–2829.
[14] Shahraki A, Taherkordi A, Haugen Ø, et al. A survey and future directions on clustering: from WSNs to IoT and modern networking paradigms. IEEE Trans Netw Serv Manage. 2020;18(2):2242–2274.
[15] Masala E, Servetti A, Basso S, et al. Challenges and issues on collecting and analyzing large volumes of network data measurements. In: M Wojciechowski, M Pechenizkiy, editors. New trends in databases and information systems. Cham: Springer; 2014. p. 203–212.
[16] Basso S, Servetti A, De Martin JC. Rationale, design, and implementation of the network neutrality bot. In Congresso AICA 2010 (L'Aquila), 2010.
[17] Sundaresan S, De Donato W, Feamster N, et al. Broadband internet performance: a view from the gateway. ACM SIGCOMM Comput Commun Rev. 2011;41(4):134–145.
[18] Zhou D, Yan Z, Fu Y, et al. A survey on network data collection. J Netw Comput Appl. 2018;116:9–23.
[19] Morariu C, Stiller B. DiCAP: distributed packet capturing architecture for high-speed network links. In 2008 33rd IEEE Conference on Local Computer Networks (LCN), October, Montreal. IEEE; 2008. p. 168–175.
[20] Oliner A, Ganapathi A, Xu W. Advances and challenges in log analysis: logs contain a wealth of information for help in managing systems. Queue. 2011;9(12):30–40.
[21] Demchenko Y, De Laat C, Membrey P. Defining architecture components of the big data ecosystem. In 2014 International Conference on Collaboration Technologies and Systems (CTS), May, Minneapolis, MN. IEEE; 2014. p. 104–112.
[22] Li M, Wang Y, Wang Z, et al. A deep learning method based on an attention mechanism for wireless network traffic prediction. Ad Hoc Netw. 2020;107:102258–102273.

[23] Rathore MM, Paul A, Hong WH, et al. Exploiting IoT and big data analytics: defining smart digital city using real-time urban data. Sustainable Cities and Society. 2018;40:600–610.

[24] Gheisari M, Wang G, Bhuiyan MZA. A survey on deep learning in big data. In 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), July, Guangzhou. IEEE; 2017. Vol. 2, p. 173–180.

[25] Suganthi ST, Ayoobkhan MUA, Bacanin N, et al. Deep learning model for deep fake face recognition and detection. PeerJ Computer Science. 2022;8:1–20.

[26] Rezaei S, Liu X. Deep learning for encrypted traffic classification: an overview. IEEE Commun Mag. 2019;57(5):76–81.

[27] Ucci D, Aniello L, Baldoni R. Survey of machine learning techniques for malware analysis. Comput Secur. 2019;81:123–147.