Classification of the Weighted Network Traffic Approach Using an Optimized Deep Neural Network Algorithm

Rasool Altaee
Department of Information Networks, College of Information Technology, University of Babylon, Babylon, Iraq
altaeerasool@gmail.com (corresponding author)

Ghaidaa A. Al-Sultany
College of Information Technology Engineering, Alzahraa University for Women, Karbala, Iraq | Department of Information Networks, College of Information Technology, University of Babylon, Babylon, Iraq
ghaidaa.almulla@alzahraa.edu.iq

Received: 15 December 2024 | Revised: 21 February 2025 | Accepted: 6 March 2025

DOI: https://doi.org/10.48084/etasr.9949

ABSTRACT

In the context of increasing network traffic type complexity, traditional traffic classification methods face significant challenges due to the nature of resource constraints, which handle massive amounts of traffic with limited processing resources. The complexity of network services is also a contributing factor, as the diversity of applications leads to a failure to adapt to new services and applications. This, in turn, results in to inaccuracies in identifying traffic types. In this paper, Weighted Network Traffic (WNT) is proposed as a means of leveraging an optimized Deep Neural Network (DNN) algorithm to enhance classification accuracy and efficiency for the entire network performance. The proposed system integrates a robust preprocessing method based on feature engineering and class reduction processes applied on IP Network Traffic Flows Labeled with 75 Apps and CICIDS2017 datasets. In the proposed WNT approach, traffic is categorized based on bandwidth metrics into three weight categories: high, moderate, and low. The optimized DNN model was evaluated using three train-test splits: 60-40, 70-30 and 80-20. The best results were achieved with the IP Network Traffic Flows Labeled with 75 Apps dataset using the 60-40 split, with a classification accuracy of 99.89%, a low loss function of 0.0043, and a and a model build time of 1 hour and 11 minutes. This performance surpasses that of the CICIDS2017 dataset and other state-of-the-art methods.

Keywords-Deep Neural Network (DNN); network traffic; classification; Weighted Network Traffic (WNT)

I. INTRODUCTION

The internet offers a variety of resources and software platforms according to user needs. However, due to the broad and diverse nature of data generation across various domains, including daily communication, education, health management, business, industrial automation, it is imperative to monitor the generation and distribution of data [1]. Moreover, the extensive growth of the internet has resulted in a substantial increase in the challenges and responsibilities faced by network administrators. These challenges include cyber-attacks, network management issues, performance and safety concerns, and the need for general monitoring of the network [2, 3].

The core network infrastructure is widely implemented in organizations that provide users with network services and mechanisms [4]. Consequently, a substantial number of applications that monitor data traffic behavior and examine its network traffic requirements have been developed. In addition, the scalability of these networks, driven by high performance demands, often results in congestion issues and challenges in executing tasks within the network [5].

The transfer of data across networks is a complex process. This complexity increases with the volume data, as well as with the increasing number of data centers and their generation sources, such as the Internet of Things (IoT) or sensors. These factors affect the formation, management, expansion, and other aspects of networks [6], necessitating the development of flexible networks that can accommodate this complexity, such as the Software-Defined Networking (SDN). Classifying data traffic in the network is considered an essential task for managing the network and ensuring its quality by allocating network resources. There are several methods that contribute to the allocation of resources, including deep learning, machine learning and resource management algorithms that contribute to the flexibility of classifying network data traffic [7].

Recent advancements in Artificial Intelligence (AI) techniques have demonstrated remarkable efficacy in modeling complex domains, achieving accurate results with untrained data, and producing efficient outcomes regardless of the nature of the data employed [8]. The rapid integration of AI technologies with IoT has contributed to the development of many smart devices, which are often heterogeneous and require robust communication and data transmission networks to maintain a steady flow of data traffic [9]. Predicting the future environment of IoT users is crucial to alleviating congestion and data transmission problems in the network. Adopting deep machine learning techniques allows for predictions based on the user activity information that is generated on the network [10].

Deep Neural Networks (DNNs) are one of the most important types of artificial neural networks, consisting of multiple hidden layers for inputs and outputs. These neural networks are designed to automatically learn complex patterns and representations from various large data sets. This feature makes them effective for classification, feature extraction, and regression tasks. The neurons in these networks are interconnected cells that process data through weighted connections, learning algorithms, and backpropagation. DNNs rely on optimization techniques that increase learning levels, organize leakage, and use advanced gradient descent algorithms, such as Adam, to improve performance and prevent the overprocessing of data [11].

Deep learning models often exhibit overfitting when trained on imbalanced datasets, a common issue in network traffic scenarios where certain classes are predominant while others are underrepresented [11]. Therefore, the classifiers demonstrate a bias toward the dominant classes and are ineffective in detecting minority traffic cases. This contributes to challenges in network resource management, system reliability and efficiency, and increases network complexity. In addition, rebalancing strategies present additional challenges, such as determining the appropriate training, samples without loss of information. These issues underscore the necessity for a dynamic deep classifier that is compatible with the nature of the changing network traffic [12].

The present study proposes an approach to data traffic classification that utilizes the Weighted Network Traffic (WNT) model. This approach enables the analysis of data traffic volumes of varying sizes based on DNN by integrating optimization techniques for accuracy and computational efficiency. This makes the proposed system a powerful classification tool for data traffic in modern networks with a high volume of data flow. Several issues have been identified in the process of network traffic classification, including the design of a framework for the classification of network traffic based on bandwidth and the construction of a robust and accurate model. The objective of the present study is to address the aforementioned issues by developing a mathematical model to calculate the total bandwidth for each network traffic and by designing and implementing a network traffic management approach based on an optimized DNN model.

II. RELATED WORKS

Various techniques for network traffic monitoring and analysis using deep learning and machine learning have been proposed. One such technique, outlined in [13], is flow-based data classification, which utilizes DNNs trained to predict the pipeline flow rate of real-world network traffic and evaluate the flow rate of data relative to a reference data set collected from a university data center. The data rates in this case are divided into three categories and the data were refined based on pre-processing and anonymization to analyze and understand the characteristics of the data traffic. The proposed system demonstrated a maximum flow feature classification accuracy of 94.1%.

In [14], authors proposed a novel deep learning approach to address the challenges posed by imbalanced data in network traffic classification. The dataset was divided into different sections, and a cost matrix was used to organize the distribution of data for each section. The costs were also applied to mitigate classification errors by distinguishing misclassifications across different types of network traffic. The approach employed two deep learning classifiers: the stacked autoencoder and the convolutional neural networks. The results on the ISCX VPNnonVPN dataset demonstrated that the proposed approach yielded the highest classification for low-frequency classes compared to related approaches.

In [15], authors proposed an approach based on deep learning that utilizes decision trees as a method for non-linear mixing groups. This approach utilizes tree-based classifiers to increase the accuracy of generalization, employing a two-level structure comprising basic and super-classifiers. The models based on decision trees are used as basic classifiers at the first level, whereas deep learning techniques are used as a super-classifier that contributes to integrating the outputs of basic classifiers at the second level. The proposed system was trained on two distinct datasets: the VPN-nonVPN dataset and the real-world traffic dataset. The following algorithms were utilized: Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), k-Nearest Neighbors (KNN), Light Gradient Boosting Machine (LightGBM), Categorical Boosting (Catboost), and Extreme Gradient Boosting (XGBoost). The results demonstrated the efficacy of the proposed system in classifying network traffic.

In [16], authors proposed an integration of the SDN architecture with machine learning technology, utilizing three supervised learning models (SVM, Nearest Centroid, and Naïve Bayes (NB)), to classify network traffic. The results demonstrated that the SVM algorithm exhibited a classification accuracy of 92%, the NB algorithm exhibited an accuracy of 96%, and the Nearest Centroid algorithm exhibited an accuracy of 91%. The study also highlighted key challenges, including handling live network traffic, and accurately classifying application-specific traffic in SDN.

A machine learning-based data segmentation and allocation model is proposed in [17] to build a system that takes into account the quality of service and dynamic segmentation of data traffic in the network. In this model, resources are intelligently allocated and redistributed to all network segments according to the extent of the difference in virtual resource requirements over time. The results of the proposed system showed the superiority of the methods adopted employed in comparison to conventional, standard, and random methods. Additionally, the system enhanced the utilization of resources while ensuring the highest quality in coordinating data traffic within the network.

The network data were analyzed in [18], where classification was implemented using machine learning and integrating specific software models that contributed to data analysis and the allocation of resources in the network. The proposed system was based on the tree ensemble and RF algorithms. The system was trained by broadcasting a video on YouTube in order to verify the quality of the video and classify the traffic in the network. The findings indicated that the proposed system achieved the highest level of performance in classifying the broadcast over the network.

In [19], an efficient network system was designed using a hybrid learning algorithm based on three main stages. The first stage was based on data collection, the second stage was based on extracting the optimal weighted features, and the third stage was based on segmentation classification. The data were collected from a 5G network, incorporating key features such as user device type, connection duration, packet loss ratio, packet delay budget, bandwidth, data transfer delay rate, speed, fluctuation, and modification type. The weight function was enhanced through the hybridization of the Glowworm Swarm Optimization algorithm with the Dynamic Harris Hawk Optimization Algorithm (GSDHOA). The features in question were characterized by an accurate classification of data traffic in the network using deep belief and neural networks. The results showed that the proposed system provides precise 5G network slices, facilitating the classification of data traffic within the network.

In [20], real-time network data traffic was analyzed and classified to each appropriate slice. The proposed system was based on a robust feature expansion technique, instead of the traditional maximum and minimum normalization. The employment of the k-means clustering algorithm to determine the optimal number of slices has been demonstrated to facilitate high classification accuracy in data collected from the SDN control unit for real-time traffic flows. The classification accuracy achieved using a feed-forward artificial neural network was 98%, whereas the classification accuracy achieved using SVM was 96%.

In [21], an internet traffic classification system was introduced, designed to function both online and offline, using a deep learning mechanism for SDN. The proposed system was based on deep learning models in the SDN controller unit, relying on the MLP, Convolutional Neural Network (CNN), and Stacked Autoencoder (SAE) algorithms. The data were generated from seven common applications that contributed to the training and testing of the proposed system. In this system, the traffic data samples were analyzed considering the quality of service. The findings demonstrated the accuracy of the proposed algorithms in assessing the performance of data traffic within the network.

In [22], a data network traffic forecasting model was proposed, which categorized traffic according to the following: broadcast, messaging, search, and cloud. The forecasting model was based on a dataset containing internet traffic patterns over a period of six days. The proposed system was based on four deep learning algorithms: MLP, attention-based encoder decoder, Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The system was evaluated based on accuracy, recall, and F1-score and the results showed that the MLP and encoder decoder algorithms achieved average performance in predicting mobile data traffic. Conversely, GRU demonstrated a satisfactory performance, with the LSTM algorithm attaining optimal accuracy in traffic prediction.

In [23], network data were analyzed to define appropriate slices according to the traffic flow behaviors and with the objective of reducing the dimensions. The feature selection process was executed, resulting in the identification of 15 features out of a total of 87. This selection was conducted using a real dataset comprising over three million entries. The proposed system employed the k-means clustering algorithm to facilitate a deeper understanding, distinction, and analysis of the behaviors of data traffic. The results exhibited a strong correlation within clusters, indicating the method's effectiveness in real-world environments.

In [24], the authors conducted a comprehensive review of advancements in machine learning models and their role in the early detection and prediction of monkeypox. The study explored several strategies using clinical and imaging data, to provide information on how machine and deep learning processes contribute to the models' enhanced accuracy and sensitivity while concurrently assessing the reliability of the design. The results indicated the potential for diagnostic precision of diseases and for implementing preventive measures on a global scale in the future.

In [25], the authors evaluated the role of machine learning models in predicting the course of Lyme disease and improving diagnosis. They investigated the environmental, host, and human factors that contribute to the rise and fall of tick populations and disease outbreaks. They employed techniques such as neural networks and RF to accurately and efficiently survey the various risks related to tick location and distribution, contributing to the improvement of public health management issues. The results demonstrated the potential of machine learning in Lyme disease management and the implications of the findings for preparedness for public health emergencies on a larger scale.

In [26], the authors proposed a novel nature-inspired optimization algorithm, called Greylag Goose Optimization (GGO), which is modeled on the behavior of greylag geese flocks. These birds have been observed to cover thousands of kilometers in a single flight and to reduce air resistance on the back of the swarm. The GGO algorithm has demonstrated its efficacy in a variety of engineering criteria functions and case studies. Statistical analysis tests have demonstrated the algorithm's superiority over numerous other optimization algorithms.

III. PROPOSED METHODOLOGY

The proposed method uses the WNT approach to classify network traffic into three main classes: high, moderate, and low weight based on the bandwidth requirements for each traffic flow. The objective of this approach is to enhance the efficiency of network resource allocation by taking into account the data traffic flow requirements. In addition, it ensures the availability of dynamic network services during peak communication periods and allows data traffic management according to user needs. The system was trained and evaluated using an optimized DNN algorithm on two datasets: IP Network Traffic Flows Labeled with 75 Apps and CICIDS2017. A flowchart of the proposed WNT approach using the DNN algorithm is presented in Figure 1. The following sections describe the primary steps of the proposed methodology.

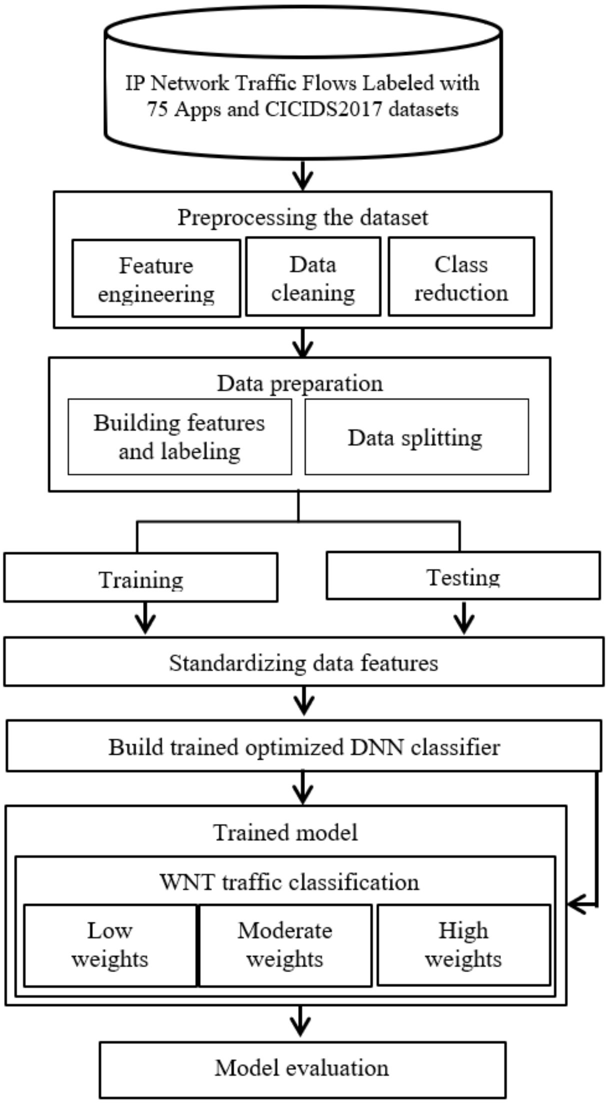

Fig. 1. The proposed WNT approach using the DNN algorithm.

A. Data Loading and Preprocessing

The proposed method based on two datasets. The first dataset, IP Network Traffic Flows Labeled with 75 Apps, was obtained from the Kaggle website and contains 87 attributes and 3,577,296 records generated from real network traffic [27]. The second dataset, CICIDS2017, was obtained from the official website of the University of New Brunswick and contains 5,950,088 records with 85 attributes [28]. Both datasets share common network traffic attributes facilitating the research. Table I presents the specifications of the datasets. Furthermore, the system is optimized by different preprocessing methods which enhance network traffic classification and decrease computations on irrelevant features in the dataset. These preprocessing methods are outlined below.

TABLE I. ATTRIBUTES OF THE DATASETS

Attribute	**Data type**	**Description**
Total_Length_of_Fwd_Packets	Numeric	Total length of the forward packets
Total_Length_of_Bwd_Packets		Total length of the backward direction packets
Fwd_IAT_Total		Total inter-arrival time of forward direction data traffic
Bwd_IAT_Total		Total inter-arrival time of backward direction data traffic

1) Feature Engineering

This technique is employed to enhance the performance of the learning model by extracting raw data, which are then converted into meaningful features that can be used in the subsequent step. It involves the calculation of total bandwidth, defined as the sum of forward and backward packet lengths divided by forward and backward inter-arrival time. This calculation provides a quantitative measure of the amount of data transmitted over a specific period, allowing for better classification of traffic types based on their bandwidth usage, as it showed in (1).

$$df[TB] = \frac{\frac{TLFP}{1000} + \frac{TLBP}{1000}}{\frac{FIATT}{1000000} + \frac{BIATT}{1000000}} \qquad (1)$$

Equation (1) represents the total bandwidth, $df[TB]$, which is calculated by dividing the total packet length by the total inter-arrival time. In particular, the total packet length consists of the total length of the forward packets, $TLFP$, and the total length of the backward packets, $TLBP$, both of which are converted from bytes to megabytes to align with the data traffic requirements of the proposed system. The total inter-arrival time in the forward direction, $FIATT$, and the backward direction, $BIATT$, represents the total arrival time converted to seconds to match the requirements of the transmitted packets.

2) Missing Data Handling

Maintaining model accuracy and reliability is essential in network traffic classification. The appropriate method depends on the extent and nature of missing data and specific requirements of the analysis. Analysts can mitigate the impact of missing data and improve the robustness of their traffic classification models by employing appropriate techniques. In the proposed system, this is implemented by replacing zeros with NaN values and removing rows with missing values in total_bandwidth.

3) Class Reduction

Class reduction is an important method that classifies rows into three weight categories (0 = low weight, 1 = moderate, 2 = high) based on the bandwidth threshold. The maximum threshold values for each category are shown in Table II.

TABLE II. TRAFFIC CLASS TYPES AND THEIR MAX BANDWIDTH

Weight	**Max bandwidth (Mbps)**
Low weight	20
Moderate weight	40
High weight	80

B. Data Preparation for the Optimized Deep Neural Network Model

This step involves the preparation of the dataset used in the proposed model, which includes the following steps:

- Features and labels: Feature X is determined as the total bandwidth and feature Y as the class.
- Train/test split: The data are split into training and testing sets using a ratio 60% - 40%.
- Standardize data: Feature data in training and testing datasets are standardized to improve model performance. This step ensures the data are on the same scale, which is essential for accurate model evaluation. In addition, it prevents data leakage, leads to faster convergence during training, and improves accuracy.

C. Building the Deep Neural Network

The DNN algorithm is used to classify network traffic based on feature engineering and class reduction processes applied to the IP Network Traffic Flows Labeled with 75 Apps and CICIDS2017 datasets. This dataset contains features such as source and destination IP addresses, ports, and inter-arrival times. The methodology focuses on analyzing the bandwidth metric to categorize network traffic into three distinct classes: high, moderate, and low-weight traffic. Leveraging these metrics enhances the WNT approach's accuracy in classifying traffic, enabling more effective management and optimization of network resources. This classification framework not only helps identify specific application traffic, but also contributes to a deeper understanding of network performance dynamics.

The DNN algorithm is applied with the following architecture:

- Three hidden layers with 64, 32, and 16 neurons, all of which use the ReLU activation function.
- The output layer has three neurons with a softmax activation function for multi-class classification.

D. Compiling the Model

The model is compiled using two methods: sparse categorical cross entropy loss and the Adam optimizer. Since the target labels are integers, sparse categorical cross-entropy loss is appropriate because it enables the model to handle class labels without requiring one-hot encoding. The Adam optimizer is used for its ability to dynamically learn and adapt for each parameter based on the first and second moments of the gradients. This leads to rapid convergence between estimates and improved performance. This model is ideal for effective training, allowing the DNN algorithm to learn from high-quality data while reducing the loss during optimization.

E. Model Training

The proposed system is trained using 10 epochs, which represents the number of complete passes through the dataset during the training phase. This phase includes a series of data processing periods for smaller groups, known as batches. Each batch contains 10 samples, meaning the repetition process updates the model weights using gradients from ten samples at a time. The goal of this training is to stabilize the learning process, use memory efficiently, and increase the speed of comparison between samples in the complete dataset throughout the training phase. It also contributes to continuously adjusting the model's parameters to reduce the loss function and ensure accurate predictions. This training improves the model's understanding of basic data patterns and distinguishes unseen data during evaluation to build an integrated, trained model that absorbs all the characteristics of the samples in the dataset.

F. Model Evaluation

The model's performance is evaluated based on its accuracy in the test dataset. The proposed system is evaluated based on the following metrics: precision, recall, F1-score, support, accuracy, time per step, and total training time.

IV. IMPLEMENTATION OF THE PROPOSED SYSTEM

The WNT system, which is optimized for the DNN algorithm, is designed to leverage a robust environment that maximizes performance and efficiency. The system is implemented using Python 3 and executed within Google Colab, a platform that offers a user-friendly interface for developing and running machine learning models. The chosen runtime type is the TPU V2-8, a powerful hardware accelerator optimized for machine learning workloads that enables faster computations and improved training times. With substantial system RAM of 334.6 GB, the environment can handle large datasets and complex models without significant memory constraints. The disk space of 225.3 GB ensures ample storage for datasets, model checkpoints, and other necessary files during the development process.

This configuration allows for a seamless workflow, facilitating efficient experimentation and iteration in training deep learning models while taking full advantage of the capabilities offered by TPU technology. Table III shows the system specifications.

TABLE III. SPECIFICATIONS OF THE PROPOSED SYSTEM

Requirement	Google Colab	Workstation computer
Language	Python 3	
CPU	Core(TM) i7-4700CPU @ 2.40GHz	
Runtime type	TPU V2-8	-
System RAM	334.6 GB	8 GB
Disk	225.3 GB	1 TB

V. RESULTS

The performance of the WNT classification approach was evaluated using a training dataset and validated using 60-40%, 70-30%, and 80-20% of both datasets.

A. *Results of 60-40 Dataset Splitting*

The results presented in Table IV demonstrate the efficiency of the data processing and the effectiveness of the proposed classification method.

TABLE IV. EVALUATION METRICS FOR 60-40 DATASET SPLITTING

Evaluation metrics	IP Network Traffic Flows Labeled with 75 Apps dataset	CICIDS2017 dataset
Time of loading dataset	39 s	47 s
Time of data preparation	32 s	39 s
Time of model training	1 h 11 m 31 s	1 h 25 m 47 s
Accuracy	99.89%	99.25%
Loss	0.0043	0.00437
Validation score	0.99893	0.99255
Training score	0.99889	0.99252
Total runtime of testing	82.0161 m	100.027 m
Total runtime of training	82.0299 m	100.032 m
Runtime of one row testing	0.0001 s	0.00017 s
Runtime of one row training	6.87696 ms	8.49 ms
Parameters / size	8,363 / 32.67 KB	8,363 / 32.67 KB
Trainable parameters / size	2,787 / 10.89 KB	2,787 / 10.89 KB
Non-trainable parameters / size	0 / 0.00 B	0 / 0.00 B
No. of epochs	10	10

The data loading process took approximately 39 s, the data preparation process took approximately 32 s, and the model training process took 1 h 11 m 44 s, reflecting the computational complexity involved in processing the dataset. The model achieved outstanding accuracy of 99.89%, demonstrating its ability to classify network traffic accurately. The loss value was 0.0043, indicating effective learning during training. The validation score reached 0.99893, suggesting excellent performance on unseen data. The training score was slightly lower, at 0.99889, but still indicative of strong model performance. The total runtime for testing was approximately 82.0161 m and training took 82.0299 m. Testing a single row took approximately 0.00010 s. Training a single row required around 6.87696 ms. The model comprised 8,363 parameters, with a total size of approximately 32.67 KB, indicating a relatively compact model. All parameters were trainable, with the same count and size as above, as shown in Table IV.

Tables V and VI present various evaluation metrics that demonstrate the high accuracy of the proposed method, especially for the first dataset with the 60–40 split. These results suggest that the WNT approach effectively balances training efficiency and classification accuracy across multiple epochs.

TABLE V. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 60-40 DATASET SPLIT OF THE IP NETWORK TRAFFIC FLOWS LABELED WITH 75 APPS DATSET.

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	1024850
1	0.99	0.99	0.99	56264
2	0.99	1.00	1.00	111708

TABLE VI. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 60-40 DATASET SPLIT OF THE CICIDS2017 DATASET

Class	Precision	Recall	F1-score	Support
0	0.995	0.995	0.995	1240015
1	0.985	0.985	0.985	68074
2	0.985	0.995	0.995	135027

B. *Results of 70-30 Dataset Splitting*

The performance of the proposed WNT classification approach was also evaluated using datasets that were split into training and validation sets at a ratio of 70-30. The results across ten epochs are summarized as follows: The time taken for each epoch remained consistent, ranging from 334 to 416 s, and the average processing time per step was stable at approximately 2 ms across all epochs. The training loss demonstrated a significant decrease over the epochs, dropping from 0.0402 in the first epoch to 0.0084 in the tenth, indicating the model's effective learning and convergence. The model achieved an impressive accuracy of 99.59% throughout the training process. Although validation loss fluctuated, it generally decreased, starting from 0.0098 and dropping to around 0.0037. This reflects the model's ability to generalize well on unseen data. Validation accuracy was consistently high, reaching 99.96% in several epochs and remaining above 99%, confirming the WNT approach's robustness in accurately classifying network traffic. Tables VII-IX show the evaluation metrics of the 70-30 dataset split.

C. *Results of 80-20 Dataset Splitting*

The WNT classification approach was also evaluated using an 80–20 train-test split. Data loading took 39 s and data preparation took 32 s, indicating the efficiency with which the data can be processed. Training the model took an estimated 1 h 4 m 6 s, achieving a high accuracy of 99.60%. The loss was measured at 0.0089, demonstrating the model's ability to effectively learn during the training phase. The validation accuracy was 0.995975, and the training score was 0.995940. This reflects the efficient performance in both the training and validation datasets. The total test run time was 41.03576 s, whereas the training took 97.88842 s. This required testing a single row at 0.000103 ms, and training a single row at 6.154843 ms. The model was based on 8,363 parameters used for training, with a total data size of 32.67 KB and over 10 epochs sizes. These results highlight the efficiency and accuracy of the WNT approach in classifying network traffic effectively and efficiently, making it the best-case study in terms of time taken compared to other studies. Tables X-XII present the results of splitting the data into 80-20. The results demonstrate the ideal accuracy of the proposed model, which achieved 100% accuracy for both the test and training cases. This confirms the model's reliability in classifying data traffic in the network. Figure 2 shows a comparison of the accuracy results and the time required to train the model system for the used datasets.

TABLE VII. EVALUATION METRICS FOR 70-30 DATASET SPLITTING

Evaluation metrics	IP Network Traffic Flows Labeled with 75 Apps dataset	CICIDS2017 dataset
Time of loading dataset	39 s	47 s
Time of data preparation	32 s	39 s
Time of model training	1 h 2 m 49 s	1 h 15 m 12 s
Accuracy	99.59%	99.12%
Loss	0.0074	0.012
Validation score	0.99586	0.99122
Training score	0.99575	0.99112
Total runtime of testing	21.18037367 m	25.5874 m
Total runtime of training	142.014153 m	160.4712
Runtime of one row testing	5.32694 ms	5.5782 ms
Runtime of one row training	8.92929 ms	9.2153 ms
Parameters / size	8,363 / 32.67 KB	8,363 / 32.67 KB
Trainable parameters / size	2,787 / 10.89 KB	2,787 / 10.89 KB
Non-trainable parameters / size	0 / 0.00 B	0 / 0.00 B
No. of epochs	10	10

TABLE VIII. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 70-30 DATASET SPLIT OF THE IP NETWORK TRAFFIC FLOWS LABELED WITH 75 APPS DATSET.

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	1366325
1	0.94	0.97	0.96	75170
2	0.99	1.00	0.99	148934

TABLE IX. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 70-30 DATASET SPLIT OF THE CICIDS2017 DATASET

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	1653102
1	0.93	0.96	0.95	90906
2	0.98	0.99	0.98	181014

TABLE X. EVALUATION METRICS FOR 80-20 DATASET SPLITTING

Evaluation metrics	IP Network Traffic Flows Labeled with 75 Apps dataset	CICIDS2017 dataset
Time of loading dataset	39 s	47 s
Time of data preparation	32 s	39 s
Time of model training	1 h 4 m 6 s	1 h 20 m
Accuracy	99.60%	99.17%
Loss	0.0089	0.011
Validation score	0.995975	0.99172
Training score	0.995940	0.99171
Total runtime of testing	41.03576 m	50.952 m
Total runtime of training	97.88842 m	120.3451 m
Runtime of one row testing	0.000103 s	0.00012 s
Runtime of one row training	6.154843 ms	7.24751 ms
Parameters / size	8,363 / 32.67 KB	8,363 / 32.67 KB
Trainable parameters / size	2,787 / 10.89 KB	2,787 / 10.89 KB
Non-trainable parameters / size	0 / 0.00 B	0 / 0.00 B
No. of epochs	10	10

TABLE XI. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 80-20 DATASET SPLIT OF THE IP NETWORK TRAFFIC FLOWS LABELED WITH 75 APPS DATSET.

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	1366325
1	0.96	0.96	0.96	75170
2	1.00	0.98	0.99	148934

TABLE XII. PRECISION, RECALL, F1-SCORE, AND SUPPORT FOR THE 80-20 DATASET SPLIT OF THE CICIDS2017 DATASET

Class	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	1653215
1	0.95	0.95	0.95	90973
2	0.99	0.97	0.98	184358

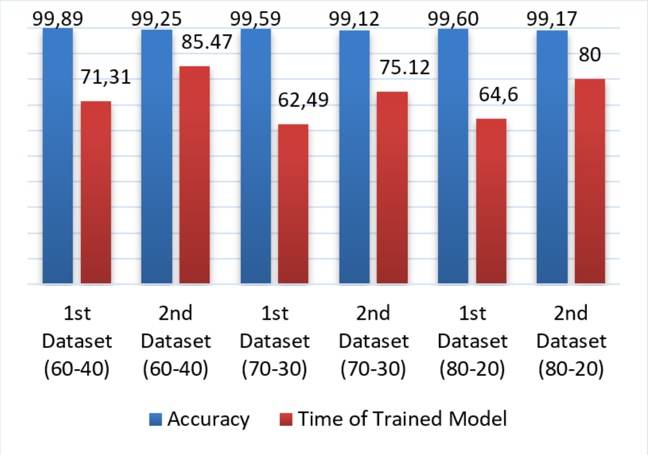

Fig. 2. Acuracy and training time comparison for the proposed system.

Furthermore, Table XIII shows the computation cost of the datasets in terms of average RAM usage and disk usage. The first dataset has a better computational cost than the second dataset due to the fewer processes required and the shorter time needed to build the classification model. Furthermore, Table XIV presents the comparison results between the proposed approach and other resent studies.

TABLE XIII. COMPUTATION COST (AVERAGE RAM AND DISK USAGE) FOR THE USED DATASETS

Resource	**IP Network Traffic Flows Labeled with 75 Apps Dataset**	**CICIDS2017 Dataset**
Average RAM usage	15.1 / 334.6 GB	18.3 / 334.6 GB
Average Disk usage	17.4 / 225.3 GB	21.1 / 225.3 GB

TABLE XIV. A COMPREHENSIVE COMPARISON OF THE PERFORMANCE OF THE PROPOSED SYSTEM AGAINST A VARIETY OF RELEVANT EXISTING SYSTEMS.

Reference	**Methodology**	**Tool, environment**	**Dataset**	**Acc. (%)**
[14] 2021	Deep learning model	Python, general network	ISCX VPNnonVPN dataset	98.6
[15] 2022	Decision tree-based and deep learning	Python, general network	VPN-nonVPN dataset	97.51
[18] 2021	SVM (linear)	Python, Mininet, SDN	IP Network Traffic Flows, Labeled with 75 Apps	96.37
[20] 2022	Feed-forward artificial neural network	Python, Mininet, SDN	IP Network Traffic Flows, Labeled with 75 Apps	98.2
Proposed system	Optimized DNN	Python, Mininet, SDN	IP Network Traffic Flows, Labeled with 75 Apps	99.89

VI. CONCLUSION

The evaluation of the Weighted Network Traffic (WNT) system's ability to classify network data traffic using the Deep Neural Network (DNN) algorithm demonstrated efficient performance at various scales and with different dataset divisions. These results highlight the model's strength and reliability in accurately classifying complex, high- volume data traffic while reducing training loss and increasing verification accuracy. For the first dataset, the accuracy rate reached almost 100%. The model also exhibited a low number of false positives and a high ability to identify relevant cases for several dataset divisions at ratios of 60-40, 70-30, and 80-20. Despite dealing with varying amounts of data, the results showed the DNN algorithm's efficiency in maintaining high performance with reasonable training and operating times.

Future improvements to the proposed system will include using genetic algorithms to systematically explore the information space, thereby improving the model's ability to classify data traffic within the network more accurately and efficiently. Additionally, real-time traffic processing will classify incoming network traffic requests based on the trained model, balancing network traffic and enhancing overall network performance. Furthermore, an optimized DNN will be developed for implementation in an edge computing environment. This DNN will optimize the management of network traffic and the classification of incoming Internet of Things (IoT) requests without exhausting the resources of constrained devices or overloading the network. This will be achieved by classifying network traffic in a flexible, scalable controller node.

REFERENCES

[1] S. Zahoor and R. N. Mir, "Resource management in pervasive Internet of Things: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 8, pp. 921–935, Oct. 2021, https://doi.org/10.1016/j.jksuci.2018.08.014.

[2] S. Jagtap, G. Garcia-Garcia, and S. Rahimifard, "Optimisation of the resource efficiency of food manufacturing via the Internet of Things," *Computers in Industry*, vol. 127, May 2021, Art. no. 103397, https://doi.org/10.1016/j.compind.2021.103397.

[3] W. A. Jabbar, T. Subramaniam, A. E. Ong, M. I. Shu'Ib, W. Wu, and M. A. de Oliveira, "LoRaWAN-Based IoT System Implementation for Long-Range Outdoor Air Quality Monitoring," *Internet of Things*, vol. 19, Aug. 2022, Art. no. 100540, https://doi.org/10.1016/j.iot.2022.100540.

[4] S. S. Gill *et al.*, "AI for next generation computing: Emerging trends and future directions," *Internet of Things*, vol. 19, Aug. 2022, Art. no. 100514, https://doi.org/10.1016/j.iot.2022.100514.

[5] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of Internet of Things in smart agriculture: A survey," *Future Generation Computer Systems*, vol. 126, pp. 169–184, Jan. 2022, https://doi.org/10.1016/j.future.2021.08.006.

[6] F. Tang, X. Chen, M. Zhao, and N. Kato, "The Roadmap of Communication and Networking in 6G for the Metaverse," *IEEE Wireless Communications*, vol. 30, no. 4, pp. 72–81, Aug. 2023, https://doi.org/10.1109/MWC.019.2100721.

[7] S. Garg, S. Sinha, A. K. Kar, and M. Mani, "A review of machine learning applications in human resource management," *International Journal of Productivity and Performance Management*, vol. 71, no. 5, pp. 1590–1610, Feb. 2021, https://doi.org/10.1108/IJPPM-08-2020-0427.

[8] M. I. Ali and A. A. Allawi, "An Artificial Neural Network Prediction Model of GFRP Residual Tensile Strength," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18277–18282, Dec. 2024, https://doi.org/10.48084/etasr.9107.

[9] Y. Fouad, N. E. Abdelaziz, and A. M. Elshewey, "IoT Traffic Parameter Classification based on Optimized BPSO for Enabling Green Wireless Networks," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18929–18934, Dec. 2024, https://doi.org/10.48084/etasr.9230.

[10] J. Nagalapuram and S. Samundeeswari, "A Framework for Smart City Traffic Management utilizing BDA and IoT," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18989–18993, Dec. 2024, https://doi.org/10.48084/etasr.8003.

[11] N. Naderializadeh, J. Sydir, M. Simsek, and H. Nikopour, "Resource Management in Wireless Networks via Multi-Agent Deep Reinforcement Learning," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications*, Atlanta, GA, USA, 2020, pp. 1–5, https://doi.org/10.1109/SPAWC48557.2020.9154250.

[12] Y. Zhang, B. Liu, Y. Gong, J. Huang, J. Xu, and W. Wan, "Application of Machine Learning Optimization in Cloud Computing Resource Scheduling and Management," in *Proceedings of the 5th International Conference on Computer Information and Big Data Applications*, Wuhan, China, 2024, pp. 171–175, https://doi.org/10.1145/3671151.3671183.

[13] C. Hardegen, B. Pfülb, S. Rieger, A. Gepperth, and S. Reißmann, "Flow-based Throughput Prediction using Deep Learning and Real-World

Network Traffic," in *2019 15th International Conference on Network and Service Management*, Halifax, Canada, 2019, pp. 1–9, https://doi.org/10.23919/CNSM46954.2019.9012716.

[14] A. Telikani, A. H. Gandomi, K.-K. R. Choo, and J. Shen, "A Cost-Sensitive Deep Learning-Based Approach for Network Traffic Classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 661–670, Mar. 2022, https://doi.org/10.1109/TNSM.2021.3112283.

[15] O. Aouedi, K. Piamrat, and B. Parrein, "Ensemble-Based Deep Learning Model for Network Traffic Classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4124–4135, Dec. 2022, https://doi.org/10.1109/TNSM.2022.3193748.

[16] M. M. Raikar, S. M. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning," *Procedia Computer Science*, vol. 171, pp. 2750–2759, Jan. 2020, https://doi.org/10.1016/j.procs.2020.04.299.

[17] N. Kumar and A. Ahmad, "Machine learning-based QoS and traffic-aware prediction-assisted dynamic network slicing," *International Journal of Communication Networks and Distributed Systems*, vol. 28, no. 1, Jan. 2022, Art. no. 27, https://doi.org/10.1504/IJCNDS.2022.120298.

[18] M. P. J. Kuranage, K. Piamrat, and S. Hamma, "Network Traffic Classification Using Machine Learning for Software Defined Networks," in *Machine Learning for Networking: Second IFIP TC 6 International Conference*, Paris, France, 2019, pp. 28–39, https://doi.org/10.1007/978-3-030-45778-5_3.

[19] M. H. Abidi *et al.*, "Optimal 5G network slicing using machine learning and deep learning concepts," *Computer Standards & Interfaces*, vol. 76, Jun. 2021, Art. no. 103518, https://doi.org/10.1016/j.csi.2021.103518.

[20] A. A. El-serwy, E. AbdElhalim, and M. A. Mohamed, "Network Slicing Based on Real-Time Traffic Classification in Software Defined Network (SDN) using Machine Learning," *Mansoura Engineering Journal*, vol. 47, no. 3, Sep. 2022, Art. no. 9, https://doi.org/10.21608/bfemu.2022.261455.

[21] L.-H. Chang, T.-H. Lee, H.-C. Chu, and C.-W. Su, "Application-Based Online Traffic Classification with Deep Learning Models on SDN Networks," *Advances in Technology Innovation*, vol. 5, no. 4, pp. 216–229, Sep. 2020, https://doi.org/10.46604/aiti.2020.4286.

[22] R. K. Gupta, A. Ranjan, M. A. Moid, and R. Misra, "Deep-Learning Based Mobile-Traffic Forecasting for Resource Utilization in 5G Network Slicing," in *Internet of Things and Connected Technologies: Conference Proceedings on 5th International Conference on Internet of Things and Connected Technologies*, Patna, India, 2020, pp. 410–424, https://doi.org/10.1007/978-3-030-76736-5_38.

[23] O. Aouedi, K. Piamrat, S. Hamma, and J. K. M. Perera, "Network traffic analysis using machine learning: an unsupervised approach to understand and slice your network," *Annals of Telecommunications*, vol. 77, no. 5, pp. 297–309, Jun. 2022, https://doi.org/10.1007/s12243-021-00889-1.

[24] A. E.-S. Saqr, A. M. Elshewey, S. K. Raju, and M. M. Eid, "A Comprehensive Review on Optimizing Machine Learning Models for Early Detection and Forecasting of Monkeypox Outbreaks," *Journal of Artificial Intelligence and Metaheuristics*, vol. 8, no. 1, pp. 09–20, Aug. 2024, https://doi.org/10.54216/JAIM.080102.

[25] A. E.-S. Saqr and A. M. Elshewey, "Utilizing Machine Learning for Predicting Lyme disease Trends and Enhancing Diagnostic Accuracy," *Metaheuristic Optimization Review*, vol. 1, no. 2, pp. 48–58, Dec. 2024, https://doi.org/10.54216/MOR.010205.

[26] E.-S. M. El-kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, and A. Ibrahim, "Greylag Goose Optimization: Nature-inspired optimization algorithm," *Expert Systems with Applications*, vol. 238, no. E, Mar. 2024, Art. no. 122147, https://doi.org/10.1016/j.eswa.2023.122147.

[27] "IP Network Traffic Flows Labeled with 75 Apps." Kaggle. [Online]. Available: https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps.

[28] "Intrusion detection evaluation dataset (CIC-IDS2017)." University of New Brunswick. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html.