Received: July 10, 2024. Revised: August 15, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Internet Traffic Classification Model Based on A-DBSCAN Algorithm

Samah Adil Mohsin[1] **Ali Saeed Alfoudi**[1,2]*

[1]*College of Computer Science and Information, Technology, University of Al-Qadisiyah, Al-Qadisiyah, Iraq*
[2]*College of Computer Science, Liverpool John Moores University, Liverpool, UK*
* Corresponding author's Email: a.s.alfoudi@qu.edu.iq

Abstract: Network traffic classification has become more important with the rapid growth of the Internet and online applications. The rapid development of the Internet has enabled explosive growth of various network traffic. The challenge lies in how to classify and identify different categories of network traffic among these huge network traffic. The classification with the massive data network traffic suffers from noise and imbalanced data. Traditional classification algorithms are becoming less effective in handling these issues of the large number of traffic generated by these technologies. This paper proposes an advanced clustering model to enhance network traffic classification and improve the quality of services based on Advanced Density-Based Spatial Clustering of Applications with Noise (A-DBSCAN) with similarity and probability distance. A-DBSCAN with adaptive parameters are applied to identify clusters. The similarity distance is utilized to distinguish between clusters to identify the quality of clusters, where the value of similarity between (-1,1). Moreover, the cluster with a value similarity of more than 0 is identified as a high-quality cluster. The probability distance is used to re-evolve the instances of negative clusters to suitable positive clusters. This stage results in consolidated optimal clusters to overcome the problem of imbalances data in the dynamic network efficiently. Additionally, the standard classifiers, such as the Random Forest (RF), K Nearest Neighbours (KNN), Decision Trees (DT), and Naïve Bayes (NB) classifier are utilized to classify data network traffic. Finally, the ISCX VPN-nonVPN dataset remarks as a benchmark to evaluate the proposed solution. The experiment results show that the performance evaluation achieves higher accuracy 81.9% compared to the standard classifiers and related works.

Keywords: Network traffic classification, Machine learning, Traffic prediction, Clustering, DBSCAN, QoS.

1. Introduction

Demand for the Internet has risen dramatically in recent years [1, 2]. In addition to the increase in internet services and the development of smart devices, internet networks are being developed to provide these required services better [3, 4].

The strong expansion in the use of Internet networks means an increase in data traffic on the networks, which makes the management of these networks a major challenge to ensure the optimal use of the network devices in addition to increasing the quality of service and the optimal use of the resources available on the network [5, 6].

The use of traditional methods in network management is currently considered a major challenge [7, 8]. Traditional methods cannot optimally exploit network resources due to their inability to analyze traffic accurately [9, 10]. The precise knowledge and analysis of the users' requirements of these networks lead to utilizing artificial intelligence techniques to analyze traffic and provide the best possible service [11].

Machine Learning (ML) methods can examine large data sets, derive patterns from them, and accurately predict data classifications [12, 13]. However, the effectiveness of these techniques depends on the integrity of the input data [11]. In addition, machine learning algorithms must be trained on balanced data sets to ensure the accuracy of their predictions [14-16].

In the real world of dynamic network traffic, there is an inherent imbalance due to the intrinsic characteristics of traffic, which is exacerbated by large differences between different traffic types [17]. Therefore, the machine learning prediction results are biased in favor of the majority classes since the minority classes are not present in the training data [9]. The basic requirement of a balanced, high-quality data input is, therefore, crucial for the optimal performance of these algorithms, making implementing these algorithms a real challenge, especially in inherently imbalanced network traffic scenarios [11].

Many techniques help provide balance in data to provide balanced training data to train machine learning algorithms to achieve high results, including up-sampling the data with minority classes or down-sampling the data with majority classes, but these techniques are ineffective in scenarios that contain dynamic network traffic because The massive imbalance in these data leads to the generation of a huge number of data the data with minority classes or reducing huge instances of the data with majority classes [18, 19]. In both cases, it leads to distortion of the data and inefficient performance [20, 21].

This paper proposes un advance of clustering model to perform balancing and removing noise of the data traffic without changing the data originality. Moreover, this paper utilizes unsupervised DBSCAN machine learning algorithm to group data with similar characteristics into clusters, where the data in one cluster is very similar and very different from the data in other clusters. By employing A-DBSCAN with similarity and probability distance, it is possible to create data that can be treated separately and balanced training data that supervised machine learning algorithms can handle. In addition, this model proposes a new approach by re-evolving the points in negative cluster to the most suitable positive cluster. Moreover, this allows to select a suitable classifier to train on the sub-data resulting from the clustering step to predict a higher accurate classifying traffic. In addition, the proposed solution has a simplified and efficient mathematical model, so it can be applied in real time. Next, the summarization of the main contributions illustrates follow:

1. This paper proposes an advanced clustering model based on A-DBSCAN to enhance network traffic analysis and improve the balancing of network traffic data.
2. The proposes solution utilizes the similarity and probability distance to remove the data noise by re-evolving the points in the negative cluster to the positive cluster. As a result with efficient classifier, this well enhance the classification and quality of services in the network traffic.

Finally, the ISCX VPN-nonVPN dataset remarks as a benchmark to evaluate the proposed solution. The experiment results show that the performance evaluation achieves higher accuracy 81.9% compared to the standard classifiers and related works.

This paper is structured as follows: Section 2 provides a comprehensive summary of previous research on Traffic classification. Section 3 describes the system model and proposed solution. Section 4 provides detailed results and a discussion. The paper is concluded in Section 5.

2. Related work

This section provides an overview of the most significant network traffic classification methods.

Wei et al. [22] propose a method for extracting features based on spike periods designed to mitigate the adverse effects of background traffic in network sessions and obtain a maximum number of traffic flow features. Furthermore, it provides a framework for identifying unfamiliar Internet traffic using JigClu, a method of training unlabelled datasets using self-supervised learning. It ultimately integrates with the clustering technique and automatically recognizes unidentified Internet traffic. The algorithm has exhibited a minimum accuracy of 74%, inaccurately recognizing unfamiliar Internet traffic using the publicly available ISCXVPN2016 dataset across several circumstances.

Baek et al. [23] proposed a new deep learning model applied to the ISCXVPN2016 row data of the ISCXVPN2016 benchmark dataset after pre-processing and cleaning it to analyze their network traffic. The proposed model achieved a low accuracy of 80% and unhandled the imbalance problem.

Al-Fayoumi et al. [24] proposed a new Association Classification (AC) algorithm that utilizes the Harmonic Mean measure instead of the traditional support and confidence measures to solve estimation issues and uncover hidden patterns that other AC algorithms may miss. The proposed model achieved an accuracy of 78% when applied to the ISCXVPN2016 benchmark dataset.

Lotfollahi et al. [25] proposed (C4.5, KNN) to analyze the network traffic characterization to study the effectiveness of flow-based time-related features to detect VPN traffic and to characterize encrypted traffic into different categories. The proposed solution achieved low accuracy when applied to the ISCXVPN2016 benchmark dataset.

Iliyasu & Deng, [26] proposed a semi-supervised learning approach using Deep Convolutional Generative Adversarial Network (DCGAN) by

Table 1. Systematic analysis of the related work

Study	The methodology	Datasets	Proposed solution	Drawbacks
[22]	Surge period-based method	ISCXVPN2016	Remove the negative impact of background traffic in network sessions and capture as many traffic flow features as possible.	This model has issue with label data because it works efficiently with structure data
[23]	an ensemble technique with deep learning algorithms	ISCXVPN2016	analyze network traffic. in addition to unhandled the imbalance problem.	Relies heavily on the behaviour of applications and specific protocols so this approach does not generalize well to datasets with different or new protocol
[24]	(PC) (GA)	ISCXVPN2016	Extracts essential features and effectively removes various features of network traffic.	Using the SMOTE (synthetic minority over-sampling technique) algorithm creates a synthetic example based on existing minority class examples which leads to Overfitting
[25]	Deep Packet (SAE, CNN)	ISCXVPN2016	integrates feature extraction and classification for network traffic analysis, enabling the differentiation between VPN and non-VPN traffic.	Limited Subspace of Hyperparameter which means the optimal configuration may not have been found
[26]	DCGAN	ISCXVPN2016	unlabelled data to improve the performance of a classifier trained on a few labelled samples.	The performance improvement in PIM is minimal beyond 20 sample packets and increasing the number of packet of packets could lead to Overfitting
[27]	KMEAN	ISCXVPN2016	The PCA was utilized to reduce the data dimensionality, whereas the KMEAN was utilized to eliminate the need to label data, which can be cumbersome, error-prone, and time-consuming. Finally, the Hellinger distance is utilized to merge similar clusters toward identifying the optimal number of clusters	Arbitrary selection number of clusters(k) in Kmean clustering, which does not accurately reflect the optimal cluster for dataset

utilize the samples generated by DCGAN generators as well as unlabeled data to improve the performance of a classifier trained on a few labelled samples. The ISCXVPN2016 was utilized as a benchmark dataset.

Min et al. [27] proposed a Principal Component Analysis (PCA) pipeline, with KMeans clustering and Hellinger distance to provision 5G network slices across the application mix. The PCA was utilized to reduce the data dimensionality, whereas the KMEAN was utilized to eliminate the need to label data, which can be cumbersome, error-prone, and time-consuming. Finally, the Hellinger distance is utilized to merge similar clusters and identify the optimal number of clusters. The ISCXVPN2016 was utilized as a benchmark dataset.

3. The proposed methodology

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a non-parametric [17] that automatically filters the noise from samples and identifies clusters with arbitrary shapes, making it suitable for uncertain data distributions. DBSCAN generates inconsistent cluster sizes, with the largest cluster often containing almost all samples, hindering the refinement of the data space [18]. Furthermore, one significant benefit of DBSCAN is that it does not require the cluster data category information for the cluster. It is becoming a more and more popular clustering technique because of these benefits. DBSCAN clusters samples based on their distance, ensuring reliable results for a given collection [19]. The DBSCAN algorithm possesses distinctive and sophisticated characteristics that are advantageous for identifying items, classes, patterns, and structures of varying shapes and sizes, in contrast to clustering algorithms not based on density. DBSCAN is a highly effective method for identifying natural clusters and their distribution in the data space, particularly when these clusters have similar densities. It does not require prior knowledge about the groups present in the dataset [20].

The proposed A-DBSCAN algorithm uses a new evolving method to reduce the number of clusters into more compatible ones, as shown in Fig. 3. Moreover, the evolving phase creates highly balanced training data by creating balanced sub-datasets in each cluster. Each sub-dataset can be introduced as sub-training data. Moreover, each sub-training dataset has its trained classifier. In predicting the test data, the label of this point on the classifier of the most suitable cluster is predicted for each point in the test data.

3.1 Preprocessing

Pre-processing data is a critical step in preparing it for analysis by machine learning algorithms. One of the essential procedures within this phase is ensuring the data is suitably adapted for the specific requirements of the experiment. A pivotal aspect of this adaptation is the normalization process. Normalization involves transforming numerical values to fall within a standardized range, typically between zero and one. This transformation is achieved according to the Eq. (1) [28]:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where (x) represents the original data value, ($x_{\min}$) and ($x_{\max}$) are the minimum and maximum values of the dataset, respectively, and (x') is the normalized value. This process ensures that all features contribute equally to the analysis, mitigating the potential for features with larger scales to influence the learning algorithms disproportionately.

After that, the data is split into training and test data. The training data is used to train the machine learning model, which is then tested in the test data to evaluate model performance.

3.2 The proposed model

The proposed model uses clustering techniques to produce highly balanced training data without generating new, unrealistic data for low-frequency labels or pruning data for high-frequency labels. This balance is achieved by creating self-balanced sub-training data using the DBSCAN algorithm. A classifier is then trained for each balanced sub-data, resulting in a classifier being trained for each sub-data that is trained and prepared for prediction. The appropriate classifier is then selected for each classification point to be classified.

Fig. 1 illustrates the main proposed solution steps as enumerated blocks, with preprocessing being the first process. Preprocessing is a critical step for machine learning techniques that produce preprocessed training and testing data.

Process 2.0 applies the DBSCAN algorithm on training data to groping instances in clusters, each point in the same cluster is highly like others in the same cluster and different from instances in other clusters, adaptive parameters are introduced in DBSCAN. Adaptive Epsilon (ε) is adjusted based on the density of points in the local neighborhood, Adaptive minimum point (MinPts) is also adjusted based on local data distribution.

The clusters that resulted from process 2.0 are not optimal, so we applied the Evolving Clustering algorithm, as shown in algorithm 3-1, to enhance the cluster's quality and ensure that the sub-datasets are highly balanced. the algorithm firstly evaluates clustering quality by calculating the Silhouette score for each cluster, according to Eq. (2), as shown in process 3.0.

$$SIL = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{x_j - \mu_k}{max\{x_j, \mu_k\}} \tag{2}$$

Where k refer to number of clusters and m refer to size of cluster i, xj is the point ∈ cluster i , $\mu_k$is the nearest cluster center in clusters pool that has the smallest distance to point xj.

Observing the clusters evaluation using the Silhouette Score, the algorithm identifies and isolates the positive clusters into a positive pool, and the

Received: July 10, 2024. Revised: August 15, 2024.

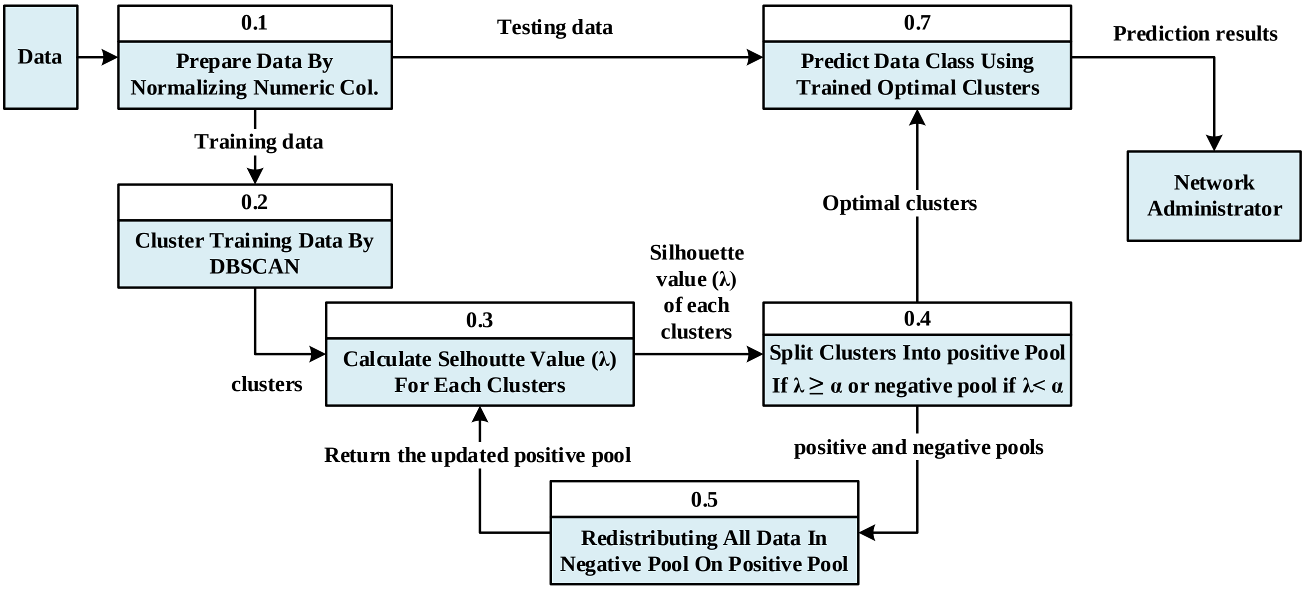

Data
0.1
Prepare Data By Normalizing Numeric Col.
Testing data
0.7
Predict Data Class Using Trained Optimal Clusters
Prediction results
Network Administrator
Training data
0.2
Cluster Training Data By DBSCAN
Optimal clusters
Silhouette value (λ) of each clusters
0.3
Calculate Selhoutte Value (λ) For Each Clusters
clusters
0.4
Split Clusters Into positive Pool If λ ≥ α or negative pool if λ< α
Return the updated positive pool
positive and negative pools
0.5
Redistributing All Data In Negative Pool On Positive Pool

Figure. 1 Main steps of the proposed solution

negative clusters into another pool named negative clusters, according to Eq. (3). This process is illustrated in Operation 4.0 in the main diagram. The beta value is a parameter that determines the degree of segregation between positive and negative clusters. Operation 4.0 refers to a specific step in the algorithm where the segregation of clusters occurs. The main diagram provides a visual representation of the entire process.

$$S = \begin{cases} s \geq \theta & add\ to\ positive \\ s < \theta & add\ to\ negative \end{cases} \tag{3}$$

Observing the clusters evaluation using the Silhouette Score, the algorithm identifies and isolates the positive clusters into a positive pool, and the negative clusters into another pool named negative clusters, according to Eq. (3). The main diagram illustrates Operation 4.0, a specific step in the algorithm where cluster segregation occurs. The beta value is a parameter that determines the degree of segregation between positive and negative clusters. The main diagram provides a visual representation of the entire process.

Process 5.0 illustrates the redistributed instances of clusters to their suitable cluster in the negative pools. If the cluster instance in the negative pool is a vector have d features: $u = (u_1, u_2, \ldots, u_d)$ and a single positive cluster have n vectors have d features: $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,d}) i \in \{1,2,\ldots,n\}$, then the fuzzy value can be calculated according to Eq. (4). This process is still repeated until no cluster has been evaluated as a negative cluster.

$$\min_{\mathrm{i}\in[1,2,\ldots,\mathrm{n}]\mathrm{c}} \mathrm{H}\,(\mathbf{u}, \mathbf{v_i}) = \min_{\mathrm{i}\in[1,2,\ldots,\mathrm{n}]} \frac{1}{\sqrt{2}} \sqrt{\textstyle\sum_{\mathrm{j}=1}^{\mathrm{d}} \left(\sqrt{\mathrm{u_j}} - \sqrt{v_{i,j}}\right)^2} \tag{4}$$

Where u_j is the jth feature of the vector $\boldsymbol{u}$ and $v_{i,j}$ is the jth feature of the vector ith instance of the cluster in the positive pool.

Regarding the predicting classes process, as shown in process 7.0, this is done by choosing the appropriate classifier for each testing instance.

ALGORITHM OF EVOLVING CLUSTERING IN A-DBSCAN

```
Input: clusters
Output: $clusters_{evolved}$
Initializing θ, α
While len(indexs_negative) = 0 or len(indexs_positive) = 1 do:
    Calculate Selh for each cluster
    Initializing $Pool_{negative}$, $pool_{positive}$
    for i:1 to len(clusters) do:
    //spliting clusters into pools
        if Selh[i] < θ:
            forEach point in clusters[i] do:
                $Pool_{negative}$ ← $Pool_{negative}$ + point
        if θ <= Selh[i]:
            $pool_{positive}$ ← $pool_{positive}$ + clusters[i]
    end
    cluster = fuzzy_func($Pool_{negative}$, $pool_{positive}$) // select suitable cluster
```

```
13. End
14. clusters_evolved = cluster
15. Return clusters_evolve
```

It is done by calculating the membership function between the instance point needed to predict all clusters according to Eq. (5).

$$k = \arg\min_{k} \left(\frac{1}{n_k} \sum_{i=1}^{n_k} \sqrt{\sum_{j=1}^{m} \left(v_j - u_{kij}\right)^2} \right) \quad (5)$$

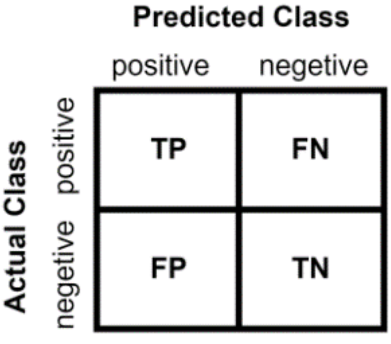
Where k is the number of clusters, n_k is the number of points belong to k^{th} cluster, and m is the number of features the j^{th} component of vector v, the u_{kij} is the j^{th} component of i^{th} element in k^{th}clusters.

To evaluate the classification performance of the proposed model with standard classifiers, in addition to related works, the Accuracy (ACC), Precision (P), Recall (RC) and F1 Score (F1) will be used. The Confusion Matrix is used to calculate these evaluation metrics after predicting labels of test data then compared with the actual labels, as shown in Fig. 2.

1. Accuracy (ACC) refers to the proportion of correct predictions made by the classifier measured as a percentage. According to Eq. (6), accuracy is the ratio of accurate predictions to the overall number of predictions. Accuracy is a suitable measure when the distribution for a balanced target class is not a good metric for evaluating performance in cases where the class is unbalanced.

$$ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

2. Precision (P): evaluate the proportion of instances that were accurately predicted but were actually negative. It is particularly useful in scenarios where the risk of a false positive is more concerning than a false negative. Precision is defined as a proportion of true positives to the total number of positive predictions, according to Eq. (7).



		Predicted Class	
		positive	negetive
Actual Class	positive	TP	FN
	negetive	FP	TN

Figure. 2 Confusion Matrix

$$P = \frac{TP}{(TP+FP)} \quad (7)$$

3. Recall (RC): measures the proportion of the actual positive instances that our model accurately identified. This metric is particularly useful in situations where a false negative is more significant than a false positive. Recall is defined as the proportion of true positives to the total number of actual positives, according to Eq. (8).

$$RC = \frac{TP}{(TP+FN)} \quad (8)$$

4. F1 score (F1): it provides a balanced measure that considers both Precision and Recall, reaching its peak When Precision and Recall are equal, according to Eq. (9). It is computed as a harmonic means of precision and recall, which imposes a higher cost on extreme values. F1-Score is particularly useful in i) if false positive (FP) and false negative (FN)are equal cost. ii) if adding more data does not significantly alter the outcome. iii) if the number of true negatives is high

$$F1 = 2 \times \frac{(P \times RC)}{(P+RC)} \quad (9)$$

4. Experiment results and discussion

This section presents the experiment's results and explains the dataset utilized in order to evaluate the performance of the proposed model:

4.1 Dataset

The ISCX VPN-non-VPN traffic dataset contains captured traffic of several apps in pcap format files, which are systematically organized according to the application that generated the packets (such as Skype and Hangouts) and the specific activity performed by the application during the capture session (such as voice call, chat, file transfer, or video call), [25]as illustrated in Table 2.

The dataset comprises packets that were captured during connection to a Virtual Private Network (VPN). Multiple sites connected by VPN route traffic over public communication networks. IP packet tunneling enables secure remote access to servers and services (2010). Similar to regular non-VPN traffic, VPN traffic can be intercepted for several applications, including Skype, during chats, voice calls, and video calls.

Utilizing ISCX VPN-nonVPN, a benchmark dataset to evaluate the proposed Enhanced DBSCAN

Table 2. The definition of all abbreviations

Abbreviation	Description
Antigen	Pre-processed vehicle objects
x	A single value in a specific feature
x_{min}	Minimum value in the dataset
x_{max}	Maximum value in the dataset
x'	Normalized data value
SIL	Silhouette Score
n	Number of points
k	Number of clusters
m	Number of features
x_j	Data point
μ_k	Mean of cluster k
θ	Threshold for determining negative and positive cluster
s	Silhouette Score value for a data point
β	Parameter determining the degree of segregation between positive and negative cluster
u	A vector in the negative pool with d feature
u_j	The $j-th$ features of vector u
v_i	A vector in the positive pool with d features
$v_{i,j}$	The $j-th$ feature of the $i-th$ vector in the positive pool
H	Hellinger distance
k^*	Optimal cluster index
n_k	Number of points in cluster k
v_j	Feature vector
u_{kij}	Feature value in the cluster

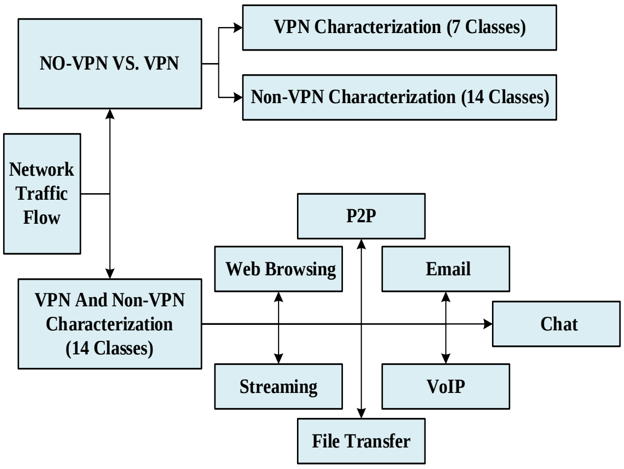

NO-VPN VS. VPN
VPN Characterization (7 Classes)
Non-VPN Characterization (14 Classes)
Network Traffic Flow
VPN And Non-VPN Characterization (14 Classes)
P2P
Web Browsing
Email
Chat
Streaming
VoIP
File Transfer

Figure. 3 Dataset content

(EDBSCAN) model for traffic classification, utilizing two distinct scenarios, A and B, illustrated in Fig. 3.

Scenario A is a two-tiered strategy separating traffic into broad categories (VPN, non-VPN). It comprises 14 types of traffic, including 7 regular types of encrypted traffic and 7 VPN types of traffic.

Scenario B is a one-tiered strategy where traffic combines VPN and non-VPN traffic into one of 14 classes.

Table 3 shows the accuracy for each class present in the data using standard classifiers compared to each other and the classification accuracy of the proposed classifier. The Naif Bayes classifier achieved the lowest accuracy compared to all classifiers used in this experiment. In contrast, the KNN classifier performed better than the first classifier in some classes and did not outperform the NB classifier in other classes. It is generally assumed that the KNN classifier can handle data better than the NB classifier. The DT classifier was superior to all previous classifiers and standard classifiers used in this experiment.

Moreover, the RF classifier achieved the best classification results on all classes appearing in the data with an apparent distinction.

On the other hand, the proposed model achieved better classification accuracy than all the standard classifiers used in this experiment. It outperformed most classes many times, except for the email class, where the difference was minimal. In the VPN_EMAIL class, the proposed model also achieved lower results with a slight difference illustrated in Fig 4.

Moreover, Table 4 illustrates the recall score per each class present in the data using standard classifiers in comparison to each other and the classification accuracy of the proposed classifier. In this experiment, the classifier scored the lowest for all classes except for the VPN Mail class, outperforming all classifiers. The KNN classifier scored better than the NB classifier in some classes. In other classes, it slightly outperformed all other classifiers used in this experiment, such as VOIP and FT. while the DT classifier achieved mixed results, better than the KNN classifier but less than the RF classifier, as the RF classifier outperformed all standard classifiers in most classes with a clear and significant difference. This emphasizes the importance of this classifier and its efficiency in classifying this data.

On the other hand, the proposed model achieved better classification recall than all the standard classifiers used in this experiment. It beat most classes several times, except for the P2P class, where

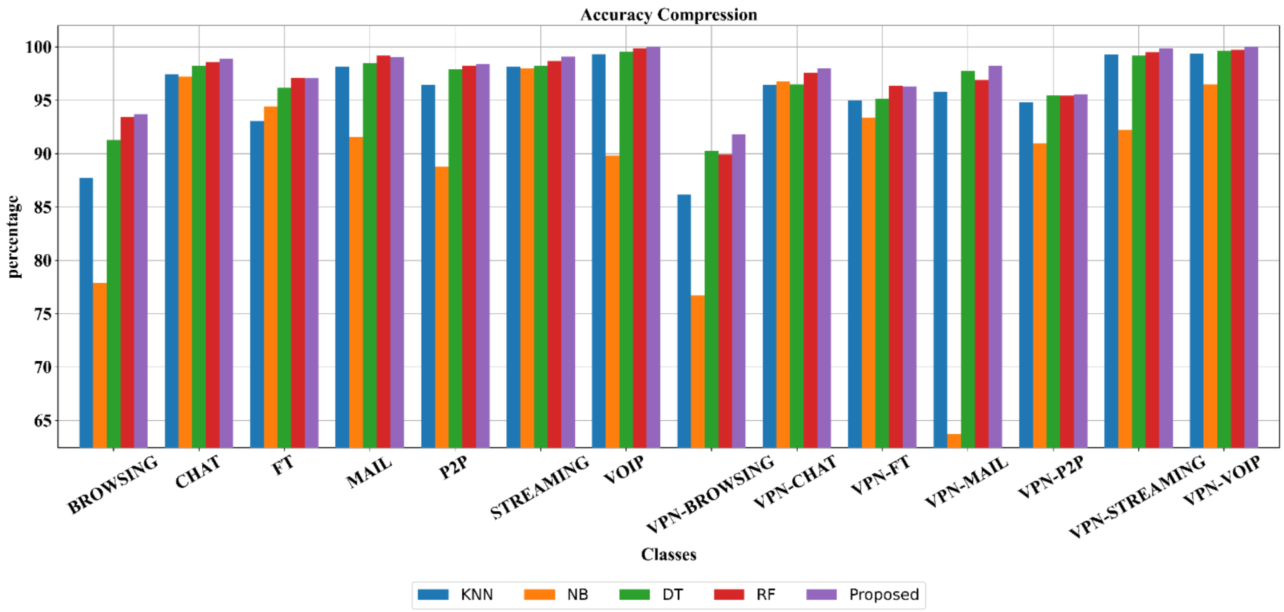


Figure. 4 Accuracy Comparison

Table 3. Accuracy Comparison

Accuracy	KNN	NB	DT	RF	Proposed
BROWSING	87.73	77.87	91.28	93.42	93.72
CHAT	97.43	97.22	98.21	98.58	98.92
FT	93.04	94.44	96.17	97.09	97.06
MAIL	98.11	91.53	98.49	99.2	99.04
P2P	96.45	88.78	97.9	98.21	98.42
STREAMING	98.15	97.99	98.24	98.67	99.07
VOIP	99.32	89.8	99.57	99.88	100
VPN-BROWSING	86.15	76.72	90.26	89.89	91.81
VPN-CHAT	96.45	96.75	96.48	97.56	97.99
VPN-FT	94.99	93.38	95.15	96.35	96.29
VPN-MAIL	95.8	63.71	97.77	96.91	98.21
VPN-P2P	94.78	90.94	95.46	95.43	95.55
VPN-STREAMING	99.26	92.24	99.2	99.51	99.87
VPN-VOIP	99.35	96.48	99.66	99.75	100

Table 4. Recall Comparison

Recall	KNN	NB	DT	RF	Proposed
BROWSING	81.33	43.07	80.27	89.6	90.53
CHAT	45.21	19.18	60.27	61.64	59.3
FT	68.45	4.28	57.22	60.43	60.29
MAIL	60	3.64	63.64	60	67.67
P2P	84.33	33	90	91	90.4
STREAMING	35.48	6.45	51.61	46.77	48.79
VOIP	95.76	25.42	97.46	99.15	100
VPN-BROWSING	67.6	0	79.47	83.87	80.93
VPN-CHAT	30.48	0	41.9	48.57	49.92
VPN-FT	41.4	1.86	54.88	58.14	61.8
VPN-MAIL	67.28	92.63	85.25	74.65	83.81
VPN-P2P	47.54	2.05	81.97	81.97	81.55
VPN-STREAMING	69.77	72.09	67.44	83.72	86.45
VPN-VOIP	89.66	45.69	95.69	94.83	96.95

there was a slight difference. The proposed model also achieved lower results in the CHAT class with a minimal difference as illustrated in Fig. 5.

Table 5 shows the precision for each class present in the data using standard classifiers compared to each other and the classification accuracy of the proposed classifier.

The Naif Bayes classifier obtained the lowest accuracy compared to all the classifiers used in this experiment, except the Mail-FT classifiers, which achieved very high results compared to all the standard classifiers used and the proposed classifier. While the KNN seed obtained higher results than the first seed in some classes, it failed to outperform the

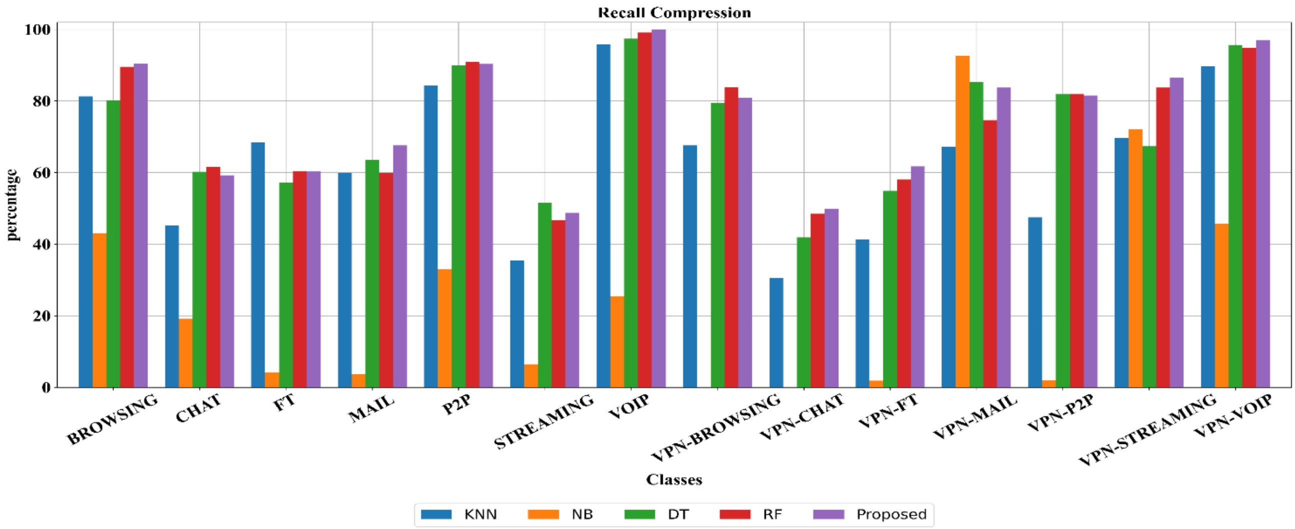


Figure. 5 Recall Comparison

Table 5. Precision Comparison

Precision	KNN	NB	DT	RF	Proposed
BROWSING	70.36	52.78	81.79	83.27	83.04
CHAT	43.42	31.11	60.27	71.43	70.89
FT	43.54	88.89	70.86	84.96	77.64
MAIL	45.83	0.9	54.69	89.19	59.13
P2P	78.82	37.93	87.66	89.8	89.22
STREAMING	52.38	36.36	54.24	74.36	73.57
VOIP	86.92	11.03	91.27	97.5	97.12
VPN-BROWSING	71.21	0	78.73	75.33	82.47
VPN-CHAT	43.24	0	45.36	67.11	67.93
VPN-FT	71.2	57.14	66.29	81.7	72.93
VPN-MAIL	69.19	14.79	82.22	78.26	84.2
VPN-P2P	73.89	8.47	66.01	65.79	64.48
VPN-STREAMING	73.17	11.48	70.73	80	77.48
VPN-VOIP	92.04	50.96	94.87	98.21	99.52

Table 6. Miss Rate Comparison

Miss Rate	KNN	NB	DT	RF	Proposed
BROWSING	18.67	56.93	19.73	10.4	8.57
CHAT	54.79	80.82	39.73	38.36	39.8
FT	31.55	95.72	42.78	39.57	38.81
MAIL	40	96.36	36.36	40	31.43
P2P	15.67	67	10	9	8.7
STREAMING	64.52	93.55	48.39	53.23	50.31
VOIP	4.24	74.58	2.54	0.85	0
VPN-BROWSING	32.4	100	20.53	16.13	18.17
VPN-CHAT	69.52	100	58.1	51.43	49.18
VPN-FT	58.6	98.14	45.12	41.86	37.3
VPN-MAIL	32.72	7.37	14.75	25.35	15.29
VPN-P2P	52.46	97.95	18.03	18.03	17.55
VPN-STREAMING	30.23	27.91	32.56	16.28	12.65
VPN-VOIP	10.34	54.31	4.31	5.17	2.15

NB seed in other classes. While it is generally considered that the KNN classifier is better for dealing with data than the nb classifier. While the DT classifier was superior to all previous classifiers and significantly in all previous standard classifiers, with a great convergence in results with the rf classifier, it obtained slightly higher results in all classes shown in the data.

On the other hand, the proposed model achieved results comparable to all standard classifiers used in

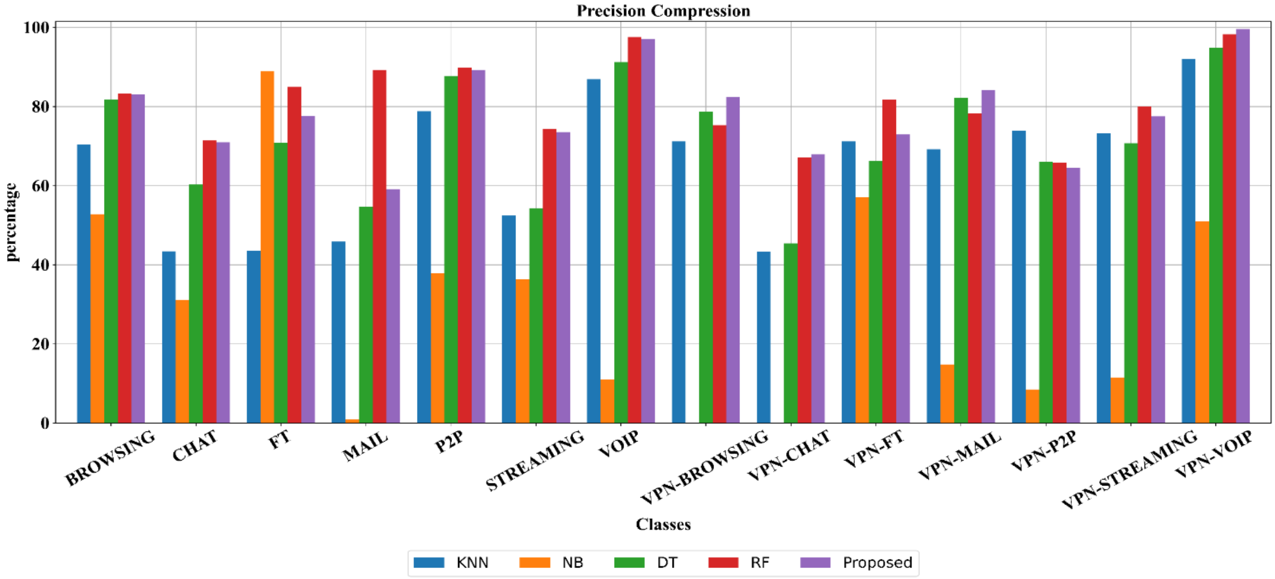


Figure. 6 Precision Comparison

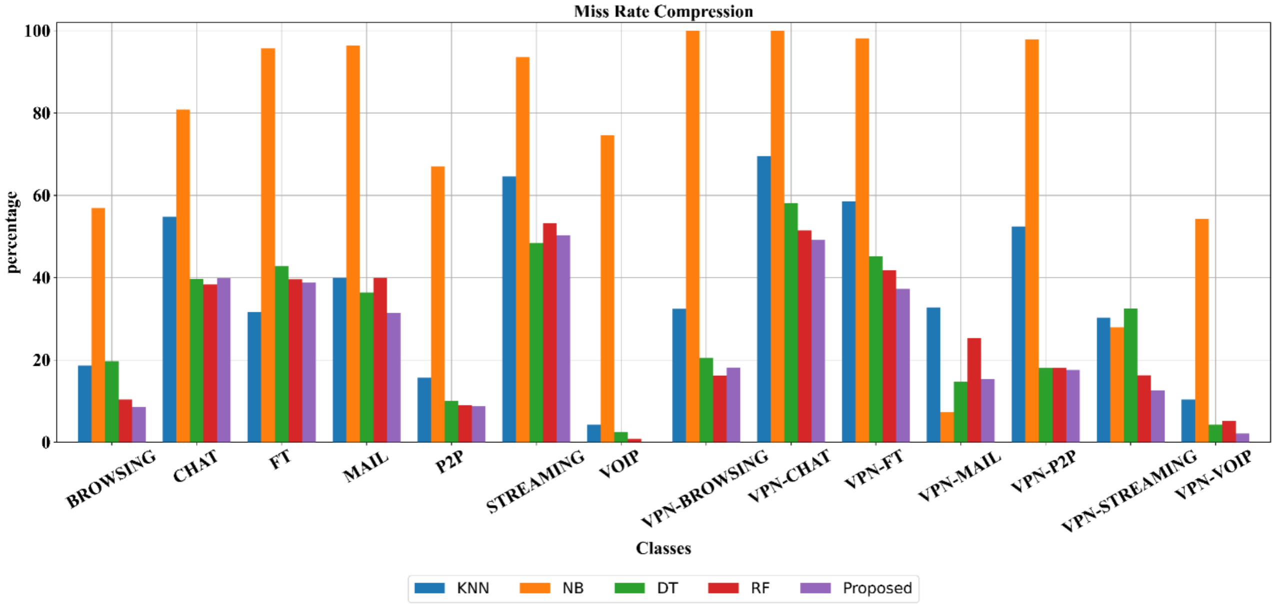
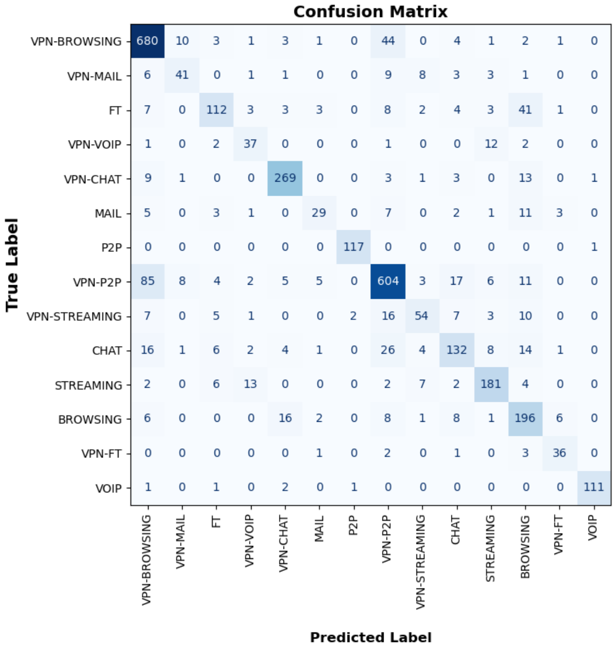


Figure. 7 Miss Rate Comparison

this experiment in some classes. It outperformed most of the classes by a significant margin but failed to achieve high results in favor of the RF classifier as illustrated in Fig. 6.

Table 6 shows the miss rate value for each class present in the data using standard classifiers compared to each other and the classification accuracy of the proposed classifier. This measure is the probability that the model predicts negative values while they are positive. In this experiment, the NB classifier achieved the highest miss rate value with all classes except for the FT class, where it outperformed all classifiers with this class, achieving the lowest value of 31.55. The KNN classifier achieved lower results than the NB classifier in some classes and did not outperform any other classifier except the NB classifier. On the other hand, the DT classifier achieved better results than the KNN classifier. Still, they varied compared to the RF classifier, as it outperformed the classifier in some classes and failed in others with slight differences. The RF classifier outperformed all standard classifiers in most classes by a significant and significant difference due to its ability to predict better than the other standard classifiers used in the experiment.

On the other hand, the proposed model achieved a miss rate lower than all standard classifiers used in



True Label \ Predicted Label	VPN-BROWSING	VPN-MAIL	FT	VPN-VOIP	VPN-CHAT	MAIL	P2P	VPN-P2P	VPN-STREAMING	CHAT	STREAMING	BROWSING	VPN-FT	VOIP
VPN-BROWSING	680	10	3	1	3	1	0	44	0	4	1	2	1	0
VPN-MAIL	6	41	0	1	1	0	0	9	8	3	3	1	0	0
FT	7	0	112	3	3	3	0	8	2	4	3	41	1	0
VPN-VOIP	1	0	2	37	0	0	0	1	0	0	12	2	0	0
VPN-CHAT	9	1	0	0	269	0	0	3	1	3	0	13	0	1
MAIL	5	0	3	1	0	29	0	7	0	2	1	11	3	0
P2P	0	0	0	0	0	0	117	0	0	0	0	0	0	1
VPN-P2P	85	8	4	2	5	5	0	604	3	17	6	11	0	0
VPN-STREAMING	7	0	5	1	0	0	2	16	54	7	3	10	0	0
CHAT	16	1	6	2	4	1	0	26	4	132	8	14	1	0
STREAMING	2	0	6	13	0	0	0	2	7	2	181	4	0	0
BROWSING	6	0	0	0	16	2	0	8	1	8	1	196	6	0
VPN-FT	0	0	0	0	0	1	0	2	0	1	0	3	36	0
VOIP	1	0	1	0	2	0	1	0	0	0	0	0	0	111

Figure. 8 Confusion Matrix

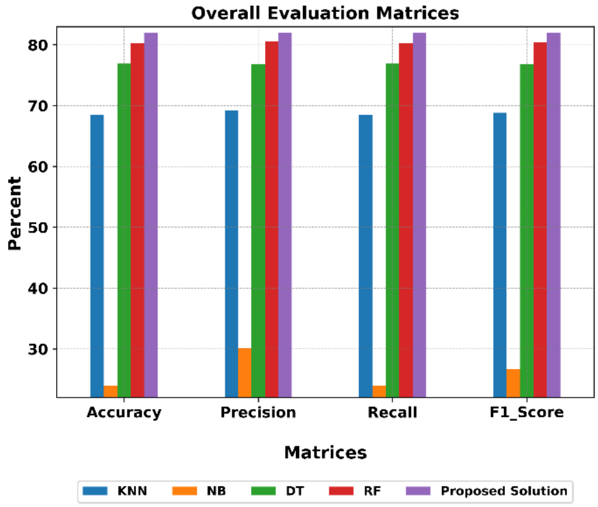


Figure. 9 Overall Evaluation Matrices

this experiment. It beat most of the classes several times, except for the CHAT and VPN-BROWSING classes, where the difference was very small. This shows the proposed model's ability to deal with unbalanced and similar data Fig. 7.

Moreover, the Fig. 8 shows the confusion matrix of the proposed model when applied to this benchmark dataset.

Table 7 shows the comparison of all classes present in the data when applying the standard classifiers and the proposed classifier. The proposed model shows high results in terms of Accuracy, Precision, Recall, compared to the standard classifiers, i.e., the ability of the proposed model to efficiently analyze and classify the data, despite the strong imbalance you suffer from. The NB achieves the worst results in all metrics, while the KNN classifier achieves better results than the NB. On the other hand, the DT achieves better results than KNN and NB but cannot outperform the RF. The RF achieves the best results compared to all standard classifiers in this experiment, as shown Fig. 9.

The proposal outlined in this document represents a significant advance in the methods previously discussed in this scientific discourse, as illustrated in Table 8. This is particularly noteworthy given that our methodology uses identical standard datasets and adheres closely to the same procedural framework for the classification process as our predecessors. This convergence in approach and the differences in results emphasize the robustness of our proposed method. It raises questions about the optimization and efficiency of existing classification methods in this area. I think it is worth noting that the comparative analysis presented here rigorously follows established scientific norms and methods, ensuring that the superiority of the proposed solution is demonstrably evidence-based and grounded in methodological precision.

Table 7. Overall Evaluation Matrices

Metrics	KNN	NB	DT	RF	Proposed
Accuracy	68.5	23.93	76.91	80.22	81.93
Precision	69.2	30.12	76.81	80.52	81.95
Recall	68.5	23.93	76.91	80.22	81.93
F1 Score	68.85	26.67	76.86	80.37	81.94

Table 8. Rrelated work comparison

ref	The methodology	Datasets	*Precision*	Recall	Acc
[22]	Surge period-based method	ISCXVPN2016	74.3%	74.3%	74%
[26]	DCGAN	ISCXVPN2016	78%	79%	80%
[15]	(PC)/(GA)	ISCX-VPN2016	-	-	78%
Proposed solution	A-DBSCAN	ISCXVPN2016	81.95%	81.95%	81.9

The proposed solution archives a precision, recall, and core of 81.95% and an accuracy of 81.9% on the ICSXVPN2016 dataset. This surpasses the precision, recall,74.3%, 74.3% and 74.1% respectively, and the accuracy of 74% achieved by the surge period-based method [22], also exceeds the precision, recall of 78% and 79%, respectively and the accuracy of 78% achieved by the DCGAN methods [26], as well as the accuracy 78% achieved by the (PC)/(PA) method [15]. This comparative analysis highlights the enhanced performance and effectiveness of our approach.

5. Conclusion

The rapid expansion of the Internet and the proliferation of online applications have necessitated more sophisticated methods of classifying network traffic. Traditional network management systems are increasingly inadequate to cope with the huge and diverse volumes of data. In this study, an advanced clustering model based on DBSCAN was introduced to address these challenges and improve network traffic analysis and quality of service. The model's effectiveness lies in its ability to manage imbalances in dynamic network data through the distribution clustering of DBSCAN. By incorporating similarity distance and probability distance measures along with an ensemble of classifiers — Random Forest, KNN, Decision Trees, and XGBoost — the proposed approach provides a robust solution to classify network traffic. Our experimental results obtained on the ISCX VPN-nonVPN benchmark dataset show that the proposed model significantly outperforms the standard classifiers and related works in terms of accuracy; where it achieves a higher accuracy with 81.9%. This confirms the feasibility and superiority of our approach in real-world scenarios in teams of mitigating the negative impact of imbalance and noise data. Future research could further explore integrating additional machine-learning techniques or optimization techniques and applying this model to other types of network datasets to solidify its utility and adaptability in different network environments.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

In this paper, the contribution distributed between the authors as follows: Conceptualization and methodology, S. Mohsin and A. Alfoudi; software, and validation, S. Mohsin and A. Alfoudi; formal analysis and investigation, S. Mohsin and A. Alfoudi; writing—original draft preparation, S. Mohsin; writing—review and editing, A. Alfoudi.

References

[1] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms", In: *Proc. of 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2451-2455, 2016.

[2] M. Reza, M. Javad, S. Raouf, and R. Javidan, "Network Traffic Classification using Machine Learning Techniques over Software Defined Networks", *International Journal of Advanced Computer Science and Applications,* Vol. 8, No. 7, 2017.

[3] H. Alizadeh, A. Khoshrou, and A. Zúquete, "Traffic classification and verification using unsupervised learning of Gaussian Mixture Models", In: *Proc. of 2015 IEEE international workshop on measurements & networking (M&N)*, pp. 1-6, 2015.

[4] C. Wang, W. Zhang, H. Hao, and H. Shi, "Network Traffic Classification Model Based on Spatio-Temporal Feature Extraction", *Electronics (Switzerland),* Vol. 13, No. 7, pp. 1-17, 2024.

[5] S. K. Singh, M. M. Salim, J. Cha, Y. Pan, and J. H. Park, "Machine learning-based network sub-slicing framework in a sustainable 5G environment", *Sustainability (Switzerland),* Vol. 12, No. 15, pp. 1-22, 2020.

[6] M. Chen, X. Wang, M. He, L. Jin, and K. Javeed, "A Network Traffic Classification Model Based on Metric Learning", Vol. 64, No. 2, pp. 941-959, 2020.

[7] Y. Su, D. Xiong, K. Qian, and Y. Wang, "A Comprehensive Survey of Distributed Denial of Service Detection and Mitigation Technologies in Software-Defined Network", *Electronics (Switzerland),* Vol. 13, No. 4, 2024.

[8] A. S. Alfoudi, M. R. Aziz, Z. A. A. Alyasseri, A. H. Alsaeedi, R. R. Nuiaa, M. A. Mohammed, K. H. Abdulkareem, and M. M. Jaber, "Hyper clustering model for dynamic network intrusion detection", *IET Communications,* vol. n/a, no. n/a.

[9] S. Buzura, A. Peculea, B. Iancu, E. Cebuc, V. Dadarlat, and R. Kovacs, "A Hybrid Software and Hardware SDN Simulation Testbed", *Sensors,* Vol. 23, No. 1, 2023.

[10] A. Blenk, A. Basta, M. Reisslein, W. Kellerer, and S. Member, "Survey on Network

Virtualization Hypervisors for Software Defined Networking", *IEEE Commun. Surv. Tutorials*, Vol. 18, No. 1, pp. 655–685, 2015.

[11] M. S. Towhid and N. Shahriar, "Encrypted Network Traffic Classification using Self-supervised Learning", In: *Proc. of the 2022 IEEE International Conference on Network Softwarization: Network Softwarization Coming of Age: New Challenges and Opportunities, NetSoft 2022*, pp. 366-374, 2022.

[12] A. K. Singh, S. Mittal, P. Malhotra, and Y. V. Srivastava, "Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means", In: *Proc. of Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, No. Iccmc, pp. 306-310, 2020.

[13] D. Al-Shammary, M. Radhi, A. H. AlSaeedi, A. M. Mahdi, A. Ibaida, and K. Ahmed, "Efficient ECG classification based on the probabilistic Kullback-Leibler divergence", *Informatics in Medicine Unlocked*, Vol. 47, p. 101510, 2024/01/01/ 2024.

[14] D. A. Popescu and A. W. Moore, "Reproducing network experiments in a time-controlled emulation environment", In: *Proc. of Traffic Monitoring and Analysis - 8th International Workshop, TMA 2016*, No. April, 2016.

[15] B. Pradhan, G. Srivastava, D. S. Roy, K. H. K. Reddy, and J. C. W. Lin, "Traffic Classification in Underwater Networks Using SDN and Data-Driven Hybrid Metaheuristics", *ACM Transactions on Sensor Networks*, Vol. 18, No. 3, 2022.

[16] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey", *IEEE Communications Surveys and Tutorials*, Vol. 22, No. 3, pp. 1761-1804, 2020.

[17] S. Soleymanpour, H. Sadr, and H. Beheshti, "An Efficient Deep Learning Method for Encrypted Traffic Classification on the Web", In: *Proc. of 2020 6th International Conference on Web Research, ICWR 2020*, pp. 209-216, 2020.

[18] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms", *Front. Comput. Sci.*, Vol. 15, pp. 1-27, 2021.

[19] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges", *Computer Networks*, Vol. 146, pp. 65-84, 2018.

[20] K. Shingare, R. Nandurkar, P. Shrivastav, and S. Bendale, "Intrusion Dataset Over Network Traffic of SDN and TCP/IP Network", *International Journal of Advanced Research in Science, Communication and Technology*, Vol. 6, No. 1, pp. 694-701, 2021.

[21] A. Shirmarz and A. Ghaffari, "Automatic Software Defined Network (SDN) Performance Management Using TOPSIS Decision-Making Algorithm", *Journal of Grid Computing*, Vol. 19, No. 2, 2021.

[22] D. Wei, F. Shi, and S. Dhelim, "A Self-Supervised Learning Model for Unknown Internet Traffic Identification Based on Surge Period", *Future Internet*, Vol. 14, No. 10, pp. 1-16, 2022.

[23] U.-J. Baek, M.-S. Lee, J.-T. Park, J.-W. Choi, C.-Y. Shin, and M.-S. Kim, "Preprocessing and Analysis of an Open Dataset in Application Traffic Classification", In: *Proc. of 2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 227-230, 2023.

[24] M. Al-Fayoumi, J. Alwidian, and M. Abusaif, "Intelligent association classification technique for phishing website detection", *International Arab Journal of Information Technology*, Vol. 17, No. 4, pp. 488-496, 2020.

[25] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: a novel approach for encrypted traffic classification using deep learning", *Soft Computing*, Vol. 24, No. 3, pp. 1999-2012, 2020.

[26] A. S. Iliyasu and H. Deng, "Semi-Supervised Encrypted Traffic Classification With Deep Convolutional Generative Adversarial Networks", *IEEE Access*, Vol. 8, pp. 118-126, 2020.

[27] Z. Min, S. Gokhale, S. Shekhar, C. Mahmoudi, Z. Kang, Y. Barve, and A. Gokhale, "A Classification Framework for IoT Network Traffic Data for Provisioning 5G Network Slices in Smart Computing Applications", In: *Proc. of 2023 IEEE International Conference on Smart Computing, SMARTCOMP 2023*, pp. 133-140, 2023.

[28] M. R. Aziz and A. S. Alfoudi, "Feature Selection of The Anomaly Network Intrusion Detection Based on Restoration Particle Swarm Optimization", *International Journal of Intelligent Engineering & Systems*, Vol. 15, No. 5, 2022, doi: 10.22266/ijies2022.1031.51.