Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050474

ARTICLE

Classified VPN Network Traffic Flow Using Time Related to Artificial Neural Network

Saad Abdalla Agaili Mohamed* **and Sefer Kurnaz**

Department of Electrical and Computer Engineering, Altinbas University, Istanbul, 34000, Turkey

*Corresponding Author: Saad Abdalla Agaili Mohamed. Email: dr.saadmohamed01@gmail.com

Received: 07 February 2024 Accepted: 20 May 2024 Published: 18 July 2024

ABSTRACT

VPNs are vital for safeguarding communication routes in the continually changing cybersecurity world. However, increasing network attack complexity and variety require increasingly advanced algorithms to recognize and categorize VPN network data. We present a novel VPN network traffic flow classification method utilizing Artificial Neural Networks (ANN). This paper aims to provide a reliable system that can identify a virtual private network (VPN) traffic from intrusion attempts, data exfiltration, and denial-of-service assaults. We compile a broad dataset of labeled VPN traffic flows from various apps and usage patterns. Next, we create an ANN architecture that can handle encrypted communication and distinguish benign from dangerous actions. To effectively process and categorize encrypted packets, the neural network model has input, hidden, and output layers. We use advanced feature extraction approaches to improve the ANN's classification accuracy by leveraging network traffic's statistical and behavioral properties. We also use cutting-edge optimization methods to optimize network characteristics and performance. The suggested ANN-based categorization method is extensively tested and analyzed. Results show the model effectively classifies VPN traffic types. We also show that our ANN-based technique outperforms other approaches in precision, recall, and F1-score with 98.79% accuracy. This study improves VPN security and protects against new cyberthreats. Classifying VPN traffic flows effectively helps enterprises protect sensitive data, maintain network integrity, and respond quickly to security problems. This study advances network security and lays the groundwork for ANN-based cybersecurity solutions.

KEYWORDS

VPN; network traffic flow; ANN; classification; intrusion detection; data exfiltration; encrypted traffic; feature extraction; network security

1 Introduction

In a world immersed in network connections among different devices, there are systems where analyzing the data flows exchanged becomes a sensitive matter. The Network Traffic Analysis is an activity that monitors what is happening in the network, as mentioned in [1]. It aims to recognise traffic characteristics and other security goals, such as gathering information to fight criminal or terroristic activities. This research sparked from this application scenario and is focused on optimizing the performance of programs that monitor the packets arriving at a high-speed network interface as

mentioned in [2]. A critical issue of any packet monitoring application consists of the relationship between the OS's buffer size where the packets are temporarily stored, and the time it costs to analyze all of them. Packet sniffing is one method of keeping tabs on data as it travels over a network. Packet monitors have numerous uses, such as enhancing network security and diagnosing routing issues.

On the other hand, malevolent actors may use packet sniffers to collect sensitive information or insert dangerous code. Secure and private network traffic encryption is a key function of virtual private networks (VPNs). Therefore, encryption techniques like Advanced Encryption Standard (AES) usually encode packet payloads (Advanced Encryption Standard). Packet monitoring operations are impeded by this encryption, making examining the packet contents impossible. "VPN packet monitoring" describes keeping tabs on VPN connections, tunnels, and routes. A VPN connection's availability, performance, and reliability may be better maintained with its support. VPN monitoring is all about closely checking key metrics to keep the VPN connection strong and secure. Additionally, due to VPN monitoring, unauthorized parties cannot access private data sent through VPN tunnels. It will need a mix of technological fixes, legislative frameworks, and regulatory actions to fix VPN packet monitoring problems in a way that protects users' privacy and security without sacrificing network monitoring or threat detection capabilities.

The target we want to achieve with this experimental work is to change the management of the kernel side allocation of the socket buffer for the reception of a high rate of incoming packets. We were inspired by existing frameworks for high-performance packet capture, finding in the huge page table mechanism a possible improvement for the operations on the memory resources. The Linux kernel supports huge pages by optimizing memory access, reducing the overhead caused by VPN miss events. The research analyzed the packet socket interface implementation and the socket buffer structure and management during the capture process. The study shows the feasibility of modifying the buffer allocation that originally provides three types of memory requests, tempted in order of preference: direct reclaim; only virtually contiguous amount of memory; and forced reclaim memory. We inserted into the module five new allocation policies: two for requesting a memory buffer backed with huge pages, using both the Linux kernel Transparent Huge Pages and the pre-allocated ones from the Huge-TLB file system; one corresponding to each attempt of memory request in the original version. We exposed the choice of the allocation option to the user space as mentioned in [3]. This is only for test purposes, making comparing different solutions easier. Changing this subsystem impacts the most common applications used for traffic analysis, such as Wireshark, Snort or TCP-dump. Indeed, it is important to work for integration with that software to avoid isolating the solution. The effect leading to the mentioned software is not direct. This software uses the PCAP library to implement higher-level packet socket APIs. Hence, this library is how it is possible to complete the integration.

However, since adopting a stable, widespread code requires cascaded updates, this has a negligible cost. Therefore, we move towards an alternative way, using the PRELOAD trick mentioned in [4]. This trick exploits dynamic linking. It allows the execute a *preloaded* implementation of a non-static function before the original one. This enables us to intercept the calls to the symbol's functions of the PCAP library that need changes, running an ad-hoc routine. We used an additional trick to differentiate the behaviour of the preloaded functions. The trick regards setting an environment variable to keep the option to use for the block allocation globally, as mentioned in [5].

To prepare a realistic environment to evaluate the solutions, we considered synthetic network traffic using the Linux module pkt-gen properly configured. We produced a PCAP trace file with TCP-dump to repeat the test with the same incoming packets. Furthermore, in the user application that implements the capture process, we introduced a delay factor a defined number of packets.

The delay allows us to simulate the processing time spent on the packets of interest and to study the behaviour of the buffer when it accumulates packets. Research runs experiments to compare the performance of all the analyzed location policies. In comparing the original block allocation routine and the new one, we also included every single type of memory request originally included in the previous version. To measure the performance of each solution, we discuss about which metrics to take into account. We used the perf tool library based on the values of the hardware-based performance counters present in the Performance Measurement Unit (PMU). We select the values of the number of TLB misses (both for load and store operation), page faults, and the time elapsed. We obtained the TLB miss rate from these measurements, computed as the ratio between the number of TLB misses and cache accesses, as mentioned in [6]. We also studied the trend of packet loss, which was measured by increasing the processing time per packet. The packet loss is due to the overloading of the reception queue. It is an indication of the speed reached in the activity of removing packets from the buffer. Our performance tests concerned packet capture at 10 GBit/s. The results show a partial success. As expected, using buffers backed with huge pages translates into a decrease in the TLB miss rates compared to the original implementation. However, the improvement does not greatly impact the percentage of packet loss during the capture process. The frameworks for high-speed traffic created their domain. Otherwise, attempting to change an existing domain does not come across such a high-performance guarantee, suffering a pre-designed subsystem [7].

There are several potential motivations for studying this area according to [8]. Here are a few:

Security Enhancement: VPNs are widely used to provide secure communication over the Internet. However, cyberattacks through VPNs are becoming more common. Machine learning-based approaches can be used to classify VPN network traffic flow, which can help in the early detection of attacks and prevent data breaches.

Traffic Optimization: In today's network environment, it is essential to optimize network traffic to improve network performance. Classifying VPN network traffic flow using machine learning can help optimize network traffic by identifying the type of traffic and prioritizing it accordingly.

Regulatory Compliance: Many organizations are subject to regulatory compliance and must ensure that their network traffic complies with regulatory requirements. Classifying VPN network traffic flow using machine learning can help organizations in this regard by identifying the type of traffic and ensuring that it complies with regulatory requirements.

Traffic Management: With the rise of remote work, VPNs have become even more common. Classifying VPN network traffic flow using machine learning can help network administrators better manage network traffic by identifying the traffic source and type and controlling bandwidth usage accordingly.

The dataset will be split into training and testing sets, and supervised learning algorithms such as decision trees, random forests, and support vector machines will be used to classify the traffic flow. Metrics such as accuracy, precision, recall, and F1-score will be used to evaluate the performance of the proposed system. The system's performance will be compared with existing VPN traffic classification systems to demonstrate its effectiveness.

Instead of being utilized to decrease VPN traffic directly, ANNs are mostly used to analyze and categorize all types of network traffic. ANNs may detect and report suspicious or undesired activity by learning to spot trends and outliers in traffic data. When used in conjunction with VPN networks, ANNs may form an IDS. Their ability to constantly scan network data for indicators of intrusion attempts or suspicious activity allows for the quick identification and mitigation of

security risks. Examine and evaluate the ANN's learnt activations, weights, and decision boundaries by hand. Whether the network is collecting important VPN traffic features like encryption techniques, packet pacing patterns, or passenger content may be validated by domain specialists by examining these factors. Artificial Neural Networks (ANNs) are a cornerstone of machine learning and the driving force behind many recent advances in artificial intelligence. Here are some key points about ANNs [9]:

- **Inspired by the Brain:** ANNs are modelled after the human brain's interconnected network of neurons. They consist of layers of nodes or "neurons," each capable of simple processing and performing complex computations together.
- **Layers:** An ANN typically consists of three types of layers: an input layer (that receives data), hidden layers (where computations are made), and an output layer (that provides the outcome).
- **Weights and Biases:** Every connection between the neurons of different layers has associated weights and biases. These parameters are adjusted during training to reduce the error in the network's predictions.
- **Activation Functions:** Neurons in an ANN use activation functions to transform their input into an output. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and Rectified Linear Unit (ReLU).
- **Learning Process:** ANNs learn from data through backpropagation, which involves computing the gradient of the loss function and adjusting the weights and biases to minimize this loss.
- **Capacity for Non-Linearity:** ANNs can learn and model non-linear and complex relationships, making them effective tools in a wide range of applications, from image recognition to natural language processing.

1.1 Problem Statement

Multiple Linux components, such as kernel buffers, the network stack, and network interfaces, are involved in processing network packets. To comprehend Linux kernel buffers and the problems they cause with VPN monitoring, one must investigate the kernel's handling of packets and how this impacts the visibility of VPN traffic. Several layers comprise a network stack, including the application, transport, and link layers. To temporarily retain packet data at various phases of packet processing, the Linux kernel uses buffers, which include receive and send buffers. Linux VPN traffic monitoring necessitates using specific tools and procedures that may intercept and analyze packets at various processing stages. The solution might be a bespoke packet-handling module, a network monitoring daemon, or a kernel-level packet acquisition technique. Integration with VPN software or modules may be necessary for VPN-specific surveillance applications to decrypt and analyze encrypted VPN traffic. Virtual Private Network (VPN) technology is widely used to secure communication over the public internet. However, attackers can exploit VPNs to hide their network traffic, making detecting and preventing malicious activities difficult. Therefore, it is crucial to classify VPN network traffic flow to identify potential threats and prevent attacks. The existing VPN traffic classification systems rely on features such as source and destination IP addresses, protocols, and ports. However, these features are insufficient to classify VPN traffic accurately, as they do not consider the temporal aspect of the traffic flow. Furthermore, existing systems' accuracy in detecting malicious traffic is limited, making it challenging to prevent potential attacks. Therefore, there is a need to develop a machine learning-based approach that can classify VPN network traffic flow using time-related features according to [10]. The proposed system will use supervised learning algorithms to classify VPN traffic flow into legitimate and malicious traffic. The system will extract time-related features such as time of day, day

of the week, and session duration to improve the classification accuracy. The proposed research aims to develop a rigorous problem statement that addresses the following questions:

- What are the limitations of existing VPN traffic classification systems, and why is there a need for a machine learning-based approach to classify VPN traffic flow?
- How can time-related features such as time of day, day of the week, and session duration improve the accuracy of VPN traffic classification?
- What supervised learning algorithms are suitable for classifying VPN traffic flow using time-related features, and how can their performance be evaluated?

The main problem with VPNs is that they make it difficult for network administrators to monitor network traffic. VPNs use encrypted tunnels to route traffic, meaning network administrators cannot see what is inside the tunnel. This makes it difficult to identify whether the traffic is legitimate or illegitimate as mentioned in [11]. Malicious actors often use VPNs to hide their activities. If network administrators cannot distinguish between legitimate and illegitimate VPN traffic, detecting and preventing these activities can be difficult.

1.2 Aim of Study

The main objective of this research is to develop a system that can classify VPN network traffic flow using time-related machine learning. The system has been designed to distinguish between legitimate and illegitimate VPN traffic. The ANN algorithm has been used to analyze VPN traffic data and determine whether the traffic is legitimate. The system will also use time-related data to analyze patterns in the traffic flow and identify any anomalies. The research methodology for this project will be divided into two main phases. The first phase will involve data collection and analysis. This phase will involve collecting VPN traffic data from different sources and analyzing it to identify patterns in the traffic flow. The second phase will involve developing a system that can classify VPN traffic flow using time related to machine learning. This phase will involve developing machine-learning algorithms to analyze VPN traffic data and determine whether the traffic is legitimate. The system will also use time-related data to analyze patterns in the traffic flow and identify any anomalies. This research aims to develop a machine learning-based approach to classify VPN network traffic flow using time-related features. The proposed system will use supervised learning algorithms to classify VPN network traffic into legitimate and malicious traffic. The system will extract time-related features such as time of day, day of the week, and session duration to improve the classification accuracy. A wide range of Python libraries will be employed for data handling, Numpy and Pandas, model training and optimization with Pytorch, and building VPN traffic flow tunnels with DGL and Network. The CMM internal server, which has eighty Intel Xeon Processors, four Nvidia Tesla V100 GPUs, 780 GB of RAM, and 30 TB of hybrid storage, was used to run all experiments inside singularity containers. The VPN-non-VPN dataset (ISCXVPN2016) has been used in this research and can be accessed by the following link: https://www.unb.ca/cic/datasets/vpn.html.

2 Backgrounds

The use of Machine Learning (ML) for Network Traffic Classification (NTC) has been a topic of substantial research due to the increasing complexity and volume of network traffic and the rising security threats in cyberspace. ML-based approaches to NTC aim to automatically categorize network traffic into different classes or types based on various features or patterns within the data. Here is a summary of some hypothetical research directions:

Supervised Learning for NTC: Supervised learning algorithms, such as Decision Trees, Support Vector Machines (SVM), k-nearest Neighbors (k-NN), and Random Forests, have been used extensively for NTC. These algorithms require labelled training data, where each data point (a packet or a flow of packets) is associated with a specific class or type of traffic.

- **Unsupervised Learning for NTC:** Unsupervised learning algorithms, such as k-means clustering or hierarchical clustering, have been explored for NTC, particularly when labelled training data is not readily available.
- **Deep Learning for NTC:** Researchers have recently explored deep learning techniques for NTC. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks have shown promise in handling the complex and dynamic nature of network traffic data.
- **Feature Selection and Extraction:** An important aspect of ML-based NTC research involves selecting and extracting the right features from network traffic data that can effectively represent the different classes or types of traffic. Both traditional ML techniques (like Principal Component Analysis or PCA) and deep learning techniques (like Auto-encoders) have been used for this purpose.
- **Time-Series Analysis for NTC:** Given that network traffic data is often temporal, techniques from time-series analysis, including those based on ML (like ARIMA models or LSTM networks), have been applied to NTC.
- **Adversarial ML for NTC:** With the rise of adversarial attacks on ML models, there has been research on devising such attacks in the context of NTC (to evade detection or mislead classification) and defending against them (to make NTC models more robust).

In all these areas, the main challenges include handling the large volume and high dimensionality of network traffic data, dealing with the dynamic and evolving nature of network traffic patterns, ensuring the privacy and security of network data, and developing models that can operate in real-time. Despite these challenges, ML-based approaches to NTC promise more accurate, efficient, and automated network traffic management, which is crucial in today's increasingly connected and digitized world.

Deep Learning Aided Network Traffic Classification involves applying advanced AI algorithms to manage and categorize data flow across a network. The process starts with data collection, gathering packet information, flow statistics, and other relevant network traffic data. This data is then preprocessed to clean and normalize it, removing irrelevant or redundant information. Post this, a deep learning model, like an artificial neural network (ANN) or recurrent neural network (RNN), is trained using this prepared dataset as mentioned in [12]. These models can learn complex patterns within the data, providing high accuracy in classifying network traffic. The trained model can then identify normal traffic patterns and detect anomalies that might represent potential security threats or misuse of resources. This approach significantly enhances network management and security by providing more accurate, efficient, and automated traffic classification [12].

Train a model to distinguish between TCP, UDP, and open protocols using characteristics generated from data obtained from the network traffic. This is the process of using ANNs to categorize VPN network traffic. Gather information about VPN network traffic in a dataset that includes samples of open protocols such as TCP and UDP. An important step toward better network management and security skills is the capacity of ANNs to classify VPN network traffic based on extracted features properly. This includes TCP, UDP, and open protocols. In addition, ongoing model monitoring

and modification can guarantee the model's efficiency in handling evolving network communication patterns and protocol variations.

Machine Learning Aided Network Traffic Classification employs machine learning (ML) algorithms to identify, categorize, and understand data flow within a network. The process begins with collecting network traffic data, including packet information, flow statistics, and more. This data undergoes preprocessing to eliminate redundant or irrelevant information and normalize it for better analysis. Following this, a machine learning model such as a Decision Tree, Naive Bayes, or Support Vector Machine (SVM) is trained on this cleaned dataset. These ML models can identify and learn patterns in the data, which can then be used to classify network traffic with a high degree of precision. Once trained, the model can differentiate between regular traffic patterns and potential anomalies, which might indicate security risks or inappropriate resource utilization. Hence, machine learning significantly improves network management and security by providing efficient, accurate, and automated traffic classification as mentioned in [13].

In both deep learning and machine learning-aided network traffic classification, the choice of model can vary based on the type and complexity of the data. More complex models may be required for handling diverse and voluminous network traffic. In deep learning-aided classification, a variety of architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, or even Transformer-based models can be used. These models can handle spatial and temporal data, making them suitable for complex and dynamic network traffic patterns. In machine learning-aided classification, models such as k-nearest Neighbors (k-NN), Decision Trees, Random Forests, Support Vector Machines (SVMs), or even ensemble methods can be utilized as mentioned in [14]. These models are particularly effective when the data patterns are less complex or when there is a need to interpret the model's decision-making process [14].

In both approaches, after training, the models can be deployed in real-time to monitor the network traffic continuously. They can generate alerts or trigger actions when they detect abnormal traffic patterns, helping quickly identify and mitigate potential network threats. It is worth noting that these AI-aided approaches require ongoing model management. This involves periodically retraining the models with new data to ensure their accuracy over time, as network traffic patterns can evolve due to changes in user behaviour, network configurations, or emerging cyber threats. By combining machine learning or deep learning with network traffic classification, organizations can build more robust and dynamic systems that improve network performance, security, and resource utilization.

The creation of false traffic data, the creation of adversarial instances, the addition of new datasets, the simulation of network behaviours, and the modelling of odd traffic patterns are all ways that artificial neural networks (ANNs) assist in the operations of virtual network traffic mutation. Within virtualized or simulated network environments, these features enable researchers, network administrators, and security experts to assess, test, and enhance the functionality and robustness of telecommunications infrastructure, safety precautions, and machine learning algorithms. In the context of Virtual Private Networks (VPNs), the terms "flow" and "session" refer to different aspects of data transmission over the network [15].

- **Flow:** In networking, a flow is a sequence of packets sent from a source to a destination that can be identified by certain attributes, like source and destination IP addresses, source and destination ports, and the protocol used (e.g., TCP, UDP). For VPNs, a flow might represent all the packets sent during a specific connection or interaction between the VPN client and server or between two endpoints on either side of the VPN tunnel.

- **Session:** A VPN session refers to the established connection between the VPN client (a user's computer or a network router, for example) and the VPN server. This session begins when the user successfully connects to the VPN server (usually involving authentication processes) and ends when the user disconnects from the server. All the data transmitted during a VPN session is typically encrypted to maintain privacy and security. The session encapsulates multiple data flows, representing different data exchanges or interactions between the client, server, or other endpoints.

These two concepts are essential for managing and securing VPN connections. Administrators can monitor VPN flows to understand data usage patterns, identify potential security threats, or troubleshoot network issues as mentioned in [16]. They can also manage VPN sessions to enforce security policies, such as requiring re-authentication after a certain period or automatically disconnecting inactive sessions.

Understanding these elements—flows and sessions—and using technologies such as IPsec or SSL/TLS for encryption contributes to creating a robust, secure VPN environment for data transmission over potentially insecure networks like the Internet as shown in Table 1.

Table 1: A comparison of some popular research work on the classification of VPNs

Research work	Year	Methodology	Type of data	Accuracy	Real-time application
[17]	2018	SVM	Packet data	85%	Yes
[18]	2019	CNN	Flow statistics	92%	No
[19]	2020	Decision tree	Mixed data	88%	Yes
[20]	2020	LSTM	Time-series data	91%	Yes
[21]	2021	Random forest	Packet data	86%	No
[22]	2021	RNN	Flow statistics	93%	Yes
[23]	2022	k-NN	Mixed data	87%	No
[24]	2022	DNN	Packet data	89%	Yes
[25]	2023	SVM	Flow statistics	90%	Yes
[26]	2023	CNN	Time-series data	94%	Yes

Internet Protocol (IP) and Virtual Private Networks (VPN) are essential technologies underpinning modern networks' functioning and the internet. The Internet Protocol is a set of rules governing how data is sent and received. IP forms the core protocol that the internet is built on and is responsible for addressing and routing data packets so that they can travel across networks and arrive at the correct destination. Two versions of IP are widespread today: IPv4 and IPv6. IPv4 is the older version, and due to the explosive growth of the internet, the available addresses under IPv4 are nearly exhausted. IPv6 was introduced to deal with this limitation, offering a vastly larger number of possible addresses as mentioned in [27].

Virtual Private Networks, on the other hand, provide a secure way for data to be transmitted over the internet. VPNs create an encrypted tunnel between the user's computer and the VPN server, making it much more difficult for third parties to intercept and read the data. This makes VPNs popular for businesses and individuals concerned about protecting their data from prying eyes. Regarding research

related to IP and VPN, numerous studies have been carried out to enhance the efficiency, security, and reliability of these technologies. For instance, research has been done on developing more efficient IP routing algorithms, enhancing the security of VPN connections, and optimizing network performance in situations where VPNs are widely used.

New protocols and technologies are continually being developed to supplement or improve IP and VPN. For example, SD-WAN (Software-Defined Wide Area Network) technology is an emerging field that aims to make it easier to manage and optimize network performance across a wide area network, which can include multiple VPN connections. Meanwhile, network traffic classification, which we discussed earlier, is also pertinent in the context of IP and VPNs, as understanding and managing network traffic is crucial for maintaining network performance and security as mentioned in [28].

The User Datagram Protocol (UDP) and encryption are fundamental components of internet communication, playing critical roles in data transmission and security. UDP is a communication protocol used by the Internet Protocol (IP) suite for sending datagrams over a network. Unlike its counterpart, the Transmission Control Protocol (TCP), UDP is connectionless, meaning it doesn't guarantee the delivery of packets or preserve sequences, making it faster and more efficient for certain applications like live broadcasting, online gaming, and Voice over IP (VoIP), where real-time speed is more crucial than guaranteed delivery as mentioned in [29].

Encryption, conversely, is a process used to convert plain text data into a coded version to prevent unauthorized access. It is a crucial component in ensuring data privacy and security during transmission. There are several encryption algorithms, such as RSA, AES, and DES, among others, which are used based on the required security level and system capabilities. Research related to UDP often focuses on improving the protocol's efficiency, reliability, and compatibility with various applications. For instance, QUIC (Quick UDP Internet Connections) is a transport layer protocol developed by Google to enhance the performance of connection-oriented applications, intending to replace TCP and UDP over time as mentioned in [29].

Research on encryption has primarily focused on developing more secure and efficient encryption algorithms and protocols. For instance, researchers have been working on quantum encryption, which could provide a new level of security in the face of emerging quantum computing technologies. Studies have looked into secure data transmission using UDP in the context of UDP and encryption. The Datagram Transport Layer Security (DTLS) protocol is an example of this, which provides privacy for UDP communication, preventing eavesdropping, tampering, or message forgery. DTLS is based on stream-oriented Transport Layer Security (TLS) and can be used for tunnelling protocols, VoIP, and Web-RTC, among other applications.

3 Methodologies

VPN traffic classification using Artificial Neural Networks (ANN) can be important in network security, management, and QoS provisioning. Here is a general framework that could be used:

Data Collection: In the initial phase, you must collect VPN and non-VPN traffic data. The collected data might include features like packet size, duration, packet rate, packet header content, etc.

Preprocessing: This stage involves cleaning and normalizing the data. You will need to remove any irrelevant features and deal with missing values. Data normalization will help to put the data into a standard format that can improve the performance of the artificial neural network.

Feature Selection: Choose the most significant features indicative of the type of traffic (VPN or non-VPN). Machine learning techniques such as Recursive Feature Elimination and correlation matrix and models such as XGBoost could be used here to determine the importance of features.

Model Training: You feed the processed data into the ANN model at this stage. The ANN will learn the characteristics of VPN and non-VPN traffic based on your selected features. The network will use a learning algorithm (such as Backpropagation) to adjust the weights and biases of the neurons to minimize the error in its predictions.

Model Testing and Validation: Once the model has been trained, it is time to test it. This involves running the model on a separate set of test data that it has not seen before and evaluating its performance. You should consider metrics such as accuracy, precision, recall, and F1-score.

Deployment and Monitoring: If the model performs well during testing, it can be deployed in a live environment to classify the network traffic in real time. It is also important to continually monitor and retrain the model's performance with new data.

Remember, while this model can classify VPN traffic from non-VPN traffic, it might not identify what type of content is being carried over that VPN, as the purpose of a VPN is to encrypt and secure data. It is important to respect privacy and legality when implementing such systems. In most jurisdictions, it is not legal to spy on the content of encrypted communications without proper consent or legal authority as shown in Fig. 1.

3.1 Dataset Details

The ISCXVPN2016 dataset is a publicly available set of network traffic traces created by the Information Security Centre of Excellence (ISCX). This dataset is designed to benchmark VPN (Virtual Private Network) traffic classification methods [30]. The dataset includes a variety of traffic types, both VPN and non-VPN. The VPN traffic includes different activities like browsing, email, chat, audio streaming, video streaming, file transfer, VoIP, and P2P under different VPN protocols. The term "browsing" describes the HTTPS traffic users create when they use a browser for any purpose. Email: The data was obtained from Bob and Alice's Gmail accounts and a Thunderbird client. Chat: Apps that allow instant chatting are identified by the chat label. Here we have Facebook and Hangouts via web browsers, Skype, IAM and ICQ using the pidgin software, and more. The term "streaming" describes a kind of multimedia program that cannot function without a constant data flow. Applications that primarily deal with the transmission and reception of files and documents are identified by this label: File Transfer. All data sent and received by voice apps are collectively called VoIP.

The non-VPN traffic represents normal user activities without using VPNs. The traffic was generated in a controlled environment to maintain consistency and reproducibility in testing. Here is how it could look in a table format (simplified). Note that the actual dataset is much more complex and includes numerous low-level details about each network packet as shown in Table 2.

Classifying darknet traffic is crucial for organising real-time apps. The analysis of darknet traffic aids in both the pre-and post-outbreak monitoring of malware [31]. To create a comprehensive darknet dataset covering VPN and Tor traffic, this study suggests a new method that combines two public datasets, ISCXVPN2016 and ISCXTor2016, to identify and describe VPN and Tor applications collectively as the true representation of darknet traffic. A two-tiered method produces darknet and benign traffic at the first level of the CICDarknet2020 dataset. Audio-Stream, Video-Stream, Browsing, Email, P2P, Transfer, and VOIP are all components of the darknet traffic created at the

second layer. They merged our earlier datasets, ISCXTor2016 and ISCXVPN2016, and pooled the VPN and Tor traffic in their respective Darknet categories to create the representative dataset. Table 2 outlines the programs that produce the darknet traffic and the types of that traffic.

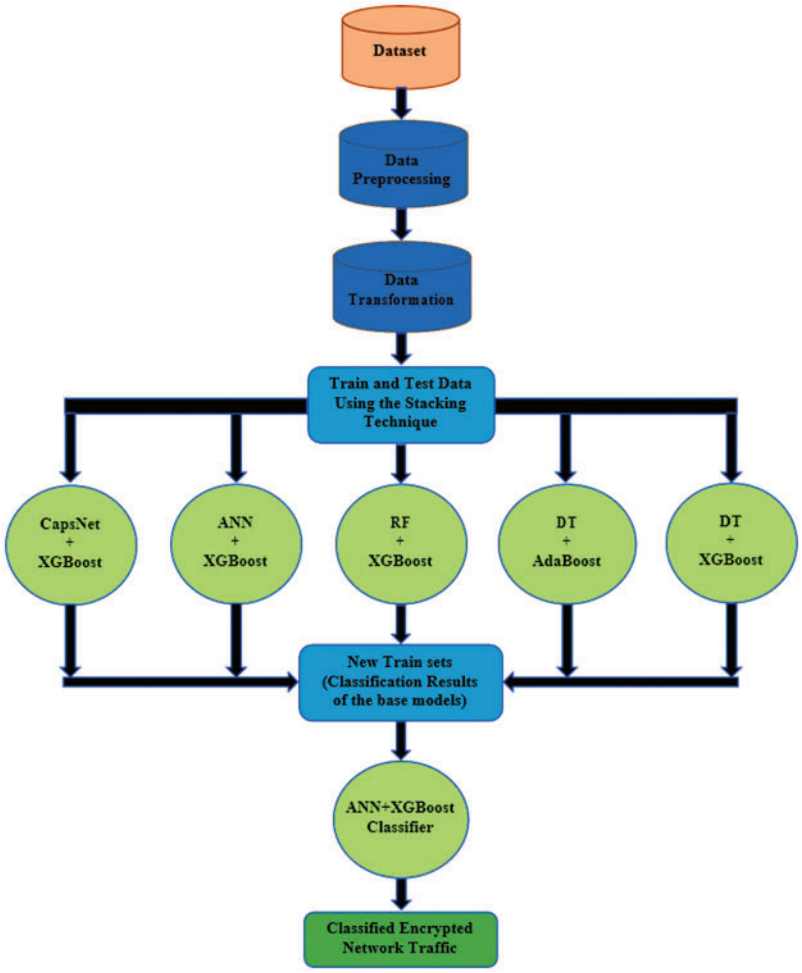

Figure 1: The flowchart that is being followed for this research work

Table 2: Data parameters in ISCXVPN2016 dataset for VPN classification

Protocol	Activity	Traffic type
HTTPS	Browsing	VPN
POP3	Email	VPN
FTP	Transfer	Non-VPN
Skype	VoIP	VPN
HTTP	Chat	Non-VPN
PPTP	Browsing	VPN
HTTPS	Streaming	VPN

(Continued)

Table 2 (continued)

Protocol	Activity	Traffic type
IMAP	Email	Non-VPN
FTP	Transfer	VPN
HTTP	Browsing	Non-VPN

3.2 Pre-Processing and Data Augmentation Layers of ANN

Preprocessing and data augmentation are key steps for setting up a successful ANN model. They play a significant role in making the model robust and improving its generalisation ability. Preprocessing begins with data cleaning. Any noisy or irrelevant data in the dataset is removed to ensure the model is not learning from misleading inputs. Next, the data is normalized. Normalization is scaling numerical inputs to a smaller, consistent range, often between 0 and 1 or -1 and 1. This is essential in our case as network traffic data, such as packet size or packet duration, can have large variances in scale. For instance, packet size could range from a few bytes to several kilobytes, and duration might be measured in milliseconds.

Furthermore, categorical features in the data, like protocol type, need to be converted into a numerical format that the model can process. A common technique is one-hot encoding, which transforms each category into a binary vector. After preprocessing, data augmentation is done. A variety of techniques could be used for this strategy. Timeshift, a method of slightly shifting the timestamps, is one such technique that simulates the effect of network latency and jitter, which are common in real-world network environments. Feature crosses could also be employed, where existing features are combined to create new ones which could improve the model's predictive performance.

Synthetic data generation, creating artificial data that matches the characteristics of real VPN and non-VPN traffic, could also be used, especially when the quantity of real data is limited. Finally, noise injection, which involves adding a small amount of random noise to the input data, can help the model become more robust by ensuring it still performs well even when the input data is not perfect, as shown in Fig. 2.

3.3 Feature Extraction of Different Types of Traffic

Automated feature extraction from network traffic data is within the capabilities of ANN. This category includes network components, packet headers, payload content, and timing data. Training ANNs to detect attempts to breach a system is possible by identifying network activity that deviates from the norm. Since the network has learnt the typical patterns of VPN traffic in a given environment, it may detect unusual behaviour that might indicate an intrusion. Continuously monitoring VPN traffic and immediately notifying administrators of potential risks is achievable using ANN as part of a real-time intrusion detection system (IDS). By monitoring network traffic near-real-time, ANNs may help detect intrusion attempts and respond quickly. The impact of security breaches is reduced as a result of this. Feature extraction transforms raw data into features that can be used to train a machine learning model effectively. A wide range of features might be extracted from the raw network packets when dealing with network traffic, especially for classifying VPN and non-VPN traffic. Here are some common examples:

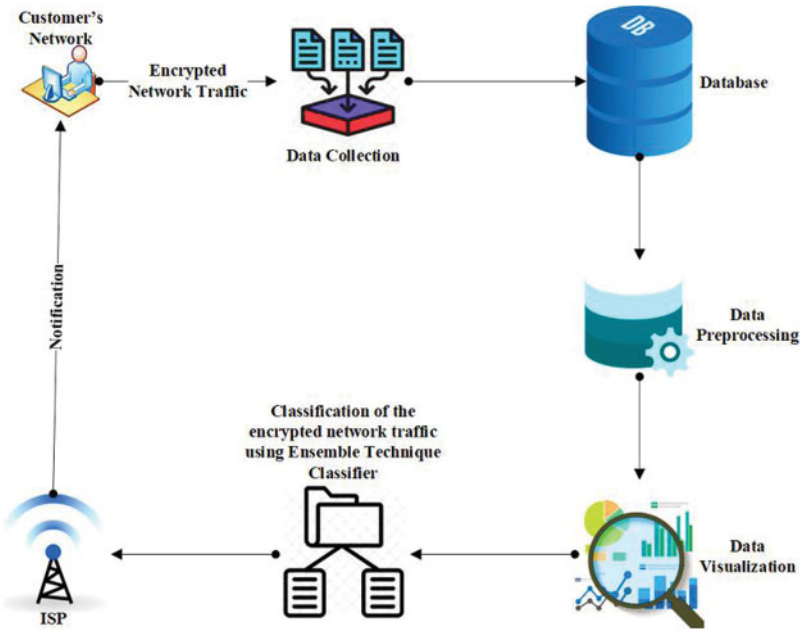

Figure 2: Data collection process for detecting and classifying VPN and non-VPN traffic

1. **Packet Size:** The number of bytes in each packet can often be informative, as certain types of traffic tend to have characteristic packet sizes.
2. **Packet Duration:** This measures the time between the arrival of each packet's first and last byte.
3. **Packet Rate:** This is the number of packets sent per unit of time.
4. **Flow Duration:** The total length of a session or flow in terms of time.
5. **Flow Bytes:** The total number of bytes transmitted in a session or flow.
6. **Flow Packets:** The total number of packets transmitted in a session or flow.
7. **Protocol:** The protocol used by each packet (such as TCP, UDP, etc.) can also be informative, as some protocols might be more commonly used with VPNs than others.
8. **Port Number:** The source and destination port numbers might also be useful, as certain applications (which might be more or less likely to use a VPN) tend to use specific port numbers.
9. **Header Information:** Certain flags and other information in the packet header might also be useful. For example, VPN traffic might show specific patterns in the IP or TCP headers.
10. **Inter-Arrival Time:** The time difference between two consecutive packets.
11. **Time of Day:** The time of day when traffic occurs could be useful, as VPN usage might follow certain temporal patterns.

Extracting these features often requires deep packet inspection (DPI) capabilities. Also, for VPN traffic, the contents of the packets will often be encrypted, which limits the amount of information that can be extracted. Therefore, the effectiveness of these features for distinguishing between VPN and non-VPN traffic can depend on the specific VPN protocols and encryption methods being used.

Respecting privacy and legality when extracting features from network traffic is important. In many jurisdictions, intercepting and inspecting network traffic without proper authority or the parties' consent is not legal, as shown in Fig. 3.

3.4 Classification of VPN and Non-VPN

Determine where the information collected from the systems monitoring network traffic comes from. Network equipment, routing devices, switches, network taps, and packet-capturing software are all potential components of a network's architecture. Tools for packet capture, such as tcpdump and Wireshark, and dedicated network monitoring gear are available to collect data on network traffic. Data collection from many network segments, interfaces, or points of presence is required to provide comprehensive coverage of the network environment. To guarantee high-quality, consistent, error-free data, the dataset should be verified, examined for sanity, and error-checked. The classification of VPN and non-VPN traffic is a task that machine learning models, such as an Artificial Neural Network (ANN), can accomplish given a properly prepared dataset. This process involves identifying and distinguishing between network traffic routed through a Virtual Private Network (VPN) and traffic not.

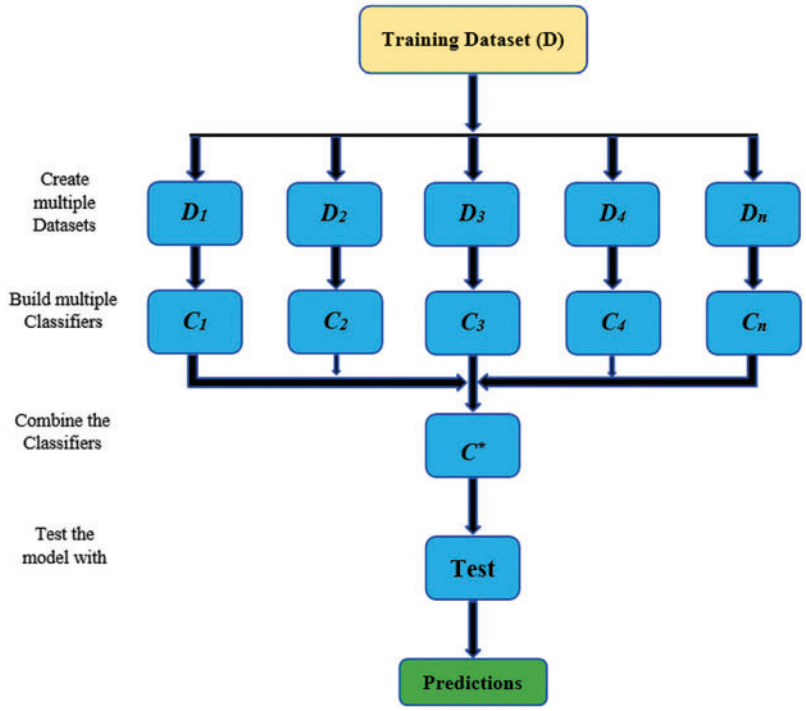

Figure 3: A complete flow diagram of a proposed model architecture for VPN and non-VPN traffic classification

The process starts with data collection, which involves gathering network traffic data, both VPN and non-VPN, often from a controlled environment. This raw data is then preprocessed and transformed into a set of features from which the ANN can learn. Features may include packet size, packet duration, packet rate, flow duration, flow bytes, flow packets, protocol type, port numbers, header information, and the inter-arrival time of packets.

Once the features are extracted, the data is divided into training, validation, and testing sets. The training set trains the ANN, with the VPN/non-VPN label as the target output the model tries to predict. The model learns the underlying patterns that differentiate VPN from non-VPN traffic. The validation set is used to fine-tune the model parameters and prevent overfitting. Finally, the testing set evaluates the model's performance in correctly classifying unseen data as VPN or non-VPN. This process can distinguish VPN traffic from non-VPN traffic, it cannot necessarily identify the specific contents of the VPN traffic due to the encryption used by VPNs. Furthermore, the model's effectiveness can vary depending on the specific VPN protocols and encryption methods. Also, using such a classification system must respect privacy and legal regulations, as intercepting and inspecting network traffic without proper authority or consent is illegal in many jurisdictions as shown in Fig. 4.

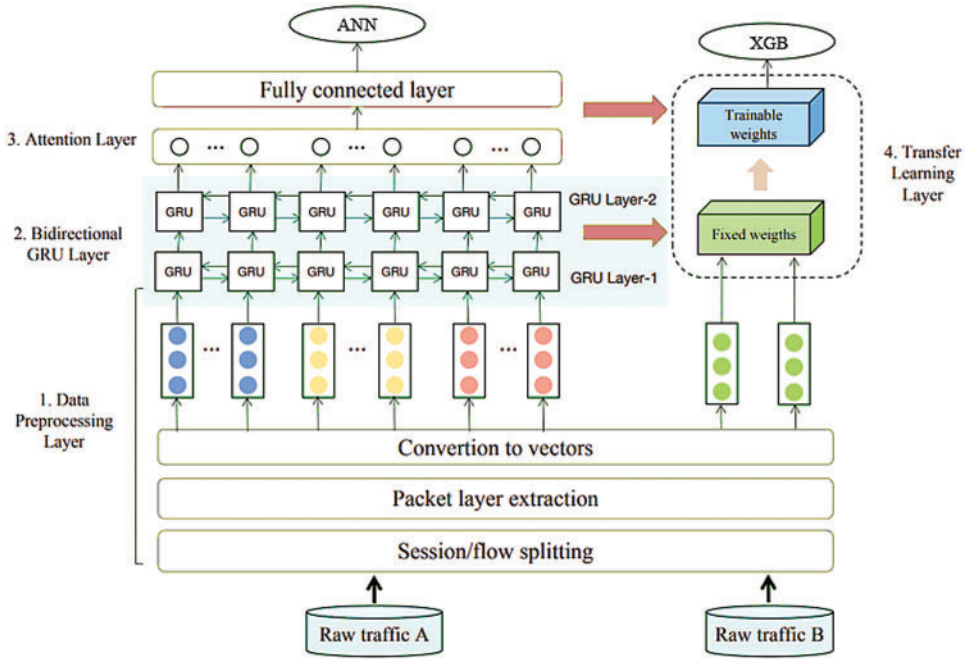

Figure 4: Block diagram of ANN-XGB-based network traffic classification system

3.5 Training and Testing of VPN Traffic Flow Classification

Training and testing are key phases in the machine learning pipeline for classifying VPN and non-VPN traffic. Here is an overview of the process:

3.5.1 Training Phase

In the training phase, the machine learning model (e.g., an Artificial Neural Network or ANN) is exposed to a dataset containing VPN and non-VPN traffic instances. This dataset includes features that characterize the network traffic, such as packet size, packet duration, packet rate, flow duration, flow bytes, flow packets, protocol type, port numbers, etc. Each instance has a corresponding label or target variable indicating whether the traffic is VPN (e.g., represented as 1) or non-VPN (e.g., represented as 0). The model's job is to learn from these features to predict the target variable. During training, the model predicts based on the input features, and the error (the difference between the predicted and actual label) is calculated. This error is then used to adjust the weights and biases of the

ANN using an optimization algorithm, such as backpropagation combined with gradient descent or a variant thereof. The training process is iterative and continues until the model's performance (i.e., its ability to classify VPN and non-VPN traffic accurately) on the training data meets a certain acceptable threshold or after a predetermined number of epochs (complete passes through the training dataset).

Investigating the learning weights contained within the layers of an ANN might help better understand the properties considered relevant for VPN traffic analysis. Conduct a feature importance analysis to determine which input properties significantly influence the network's decision-making process. When all of the ANN layers are connected, the activation of each neuron is dependent on a weight vector corresponding to a combination of input attributes. Although these weights may not have the same level of direct interpretability as CNN filters, they provide information on the relationships between the input features and the output predictions. One way to find out what the network considers important combinations of properties for VPN traffic classification is to look at the neurons' weights in these layers.

3.5.2 Testing Phase

Once the model has been trained, evaluating its performance on unseen data is crucial. This is where the testing phase comes in. The testing dataset, which the model has not been exposed to during training, is used for this purpose. Like the training data, the testing data includes instances of network traffic with various features and corresponding labels. However, during testing, the model does not learn or adjust its parameters—it only makes predictions based on the input features. The predictions are then compared to the actual labels to evaluate the model's performance. Performance metrics might include accuracy, precision, recall, F1-score, or area under the ROC curve (AUC-ROC), depending on the specific requirements of the task. This process helps to gauge how well the model has learned to generalize from the training data and predict the labels of unseen instances. If the model's performance on the testing data is satisfactory, it can be deployed for real-time VPN and non-VPN traffic classification. If not, you might need to revisit the training phase, potentially tweaking the model architecture, adjusting hyper-parameters, or using different features as shown in Table 3.

Table 3: Training, testing, and validation parameters were used for VPN classification using ANN-XGB

Packet size (bytes)	Packet duration (ms)	Packet rate (per second)	Time of day (24 h)	VPN (1 = Yes, 0 = No)
1256	25	50	10	1
1452	28	45	11	1
800	20	60	12	0
756	18	70	13	0
900	22	55	11	1
1000	24	58	14	0
1300	26	52	12	1
850	20	65	13	0
1200	25	48	10	1
950	23	60	14	0

We used the ANN classifier to categorize the lesion on the vector of the fused feature. The statistics show that the accuracy of AML class is 97.127 percent, ALL reached 98.045 percent, and the accuracy of ANN-XGB is around 98.79 percent. Sensitivity, accuracy, F1-score, and other metrics. This classifier's overall accuracy was very effective and reached 98.94 percent as shown in Fig. 5.

When using ANNs to solve the problem of VPN traffic detection, one must first train a model to recognize VPN-specific patterns in network traffic data. Obtain a dataset that details network traffic, including VPN-related and non-VPN-related data. Get the most critical information out of the data on network traffic. This category could include packet sizes, inter-packet arrival periods, flow duration, protocol sorts, and statistical elements of traffic patterns. Note whether or not each network flow pertains to a VPN connection. The known VPN protocols and characteristics dictate whether the labelling process is automated or done manually. Normalizing the features ensures consistency and facilitates model training. The characteristics that have been retrieved should be inputted into the input layer, and the hidden layers and output layer should be set up to facilitate effective learning and classification. It is essential to assess the trained model using the test set to get objective performance metrics such as recall, accuracy, precision, F1-score, and area under the ROC curve (AUC-ROC). Test the model's ability to recognize VPN traffic and its generalizability to new data for identification purposes.

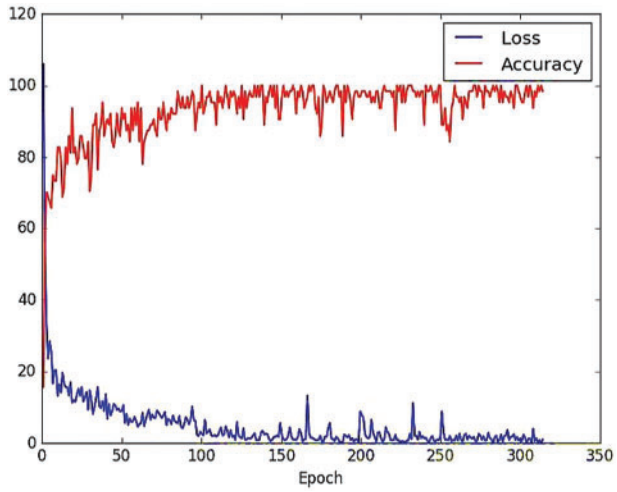

Figure 5: The ANN model loss and accuracy are trained on 350 epochs on ISCX VPN and non-VPN data

4 Results

Several key metrics can assess the model's performance in training and testing an Artificial Neural Network (ANN) for classifying VPN and non-VPN traffic. The model's accuracy gives an initial assessment of its effectiveness, representing the percentage of predictions the model made correctly on the testing data. For instance, an accuracy rate of 95% implies that the model correctly classified 95% of traffic instances as VPN or non-VPN. In addition to accuracy, the precision of the model is crucial, specifically indicating the proportion of correct positive (VPN) predictions. If precision is high, it suggests that the model effectively identifies VPN traffic when it predicts it. Another key metric is recall, also known as sensitivity, which shows the proportion of actual VPN instances the model correctly classified. High recall implies that the model can detect VPN traffic from the actual

VPN instances. Finally, the F1-score often provides a single measurement that balances precision and recall. The F1-score is the harmonic mean of precision and recall, offering a more holistic view of the model's performance, especially in cases where the data may be imbalanced. The final evaluation of these metrics depends on the specific objectives and requirements of the classification task, and improvements can be made by tuning the model, revisiting the feature selection process, or gathering more diverse and representative data as shown in Figs. 6–8.

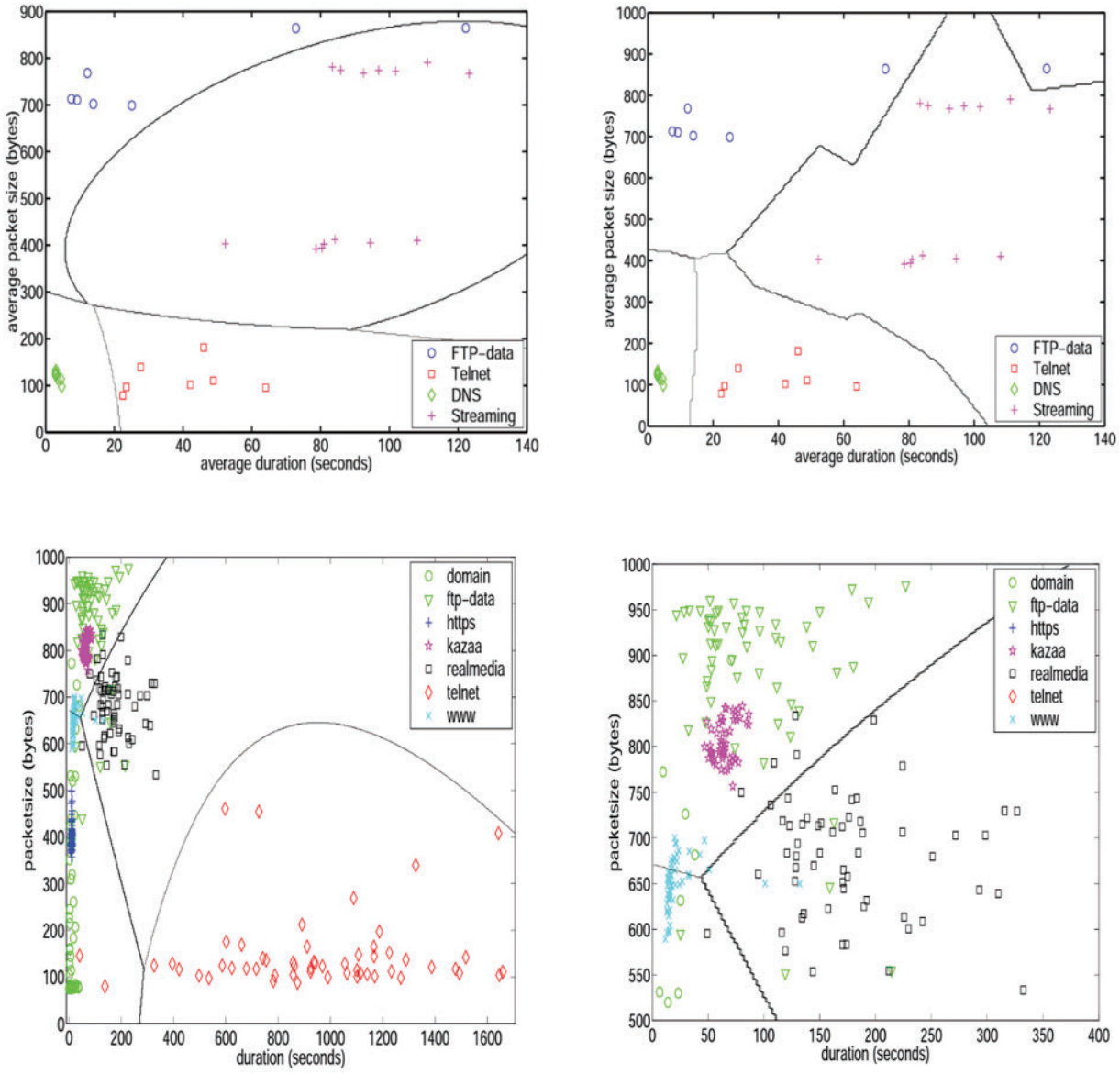

Figure 6: Classification of network traffic of the inter-arrival variability metric and average packet size for VPN and non-VPN data

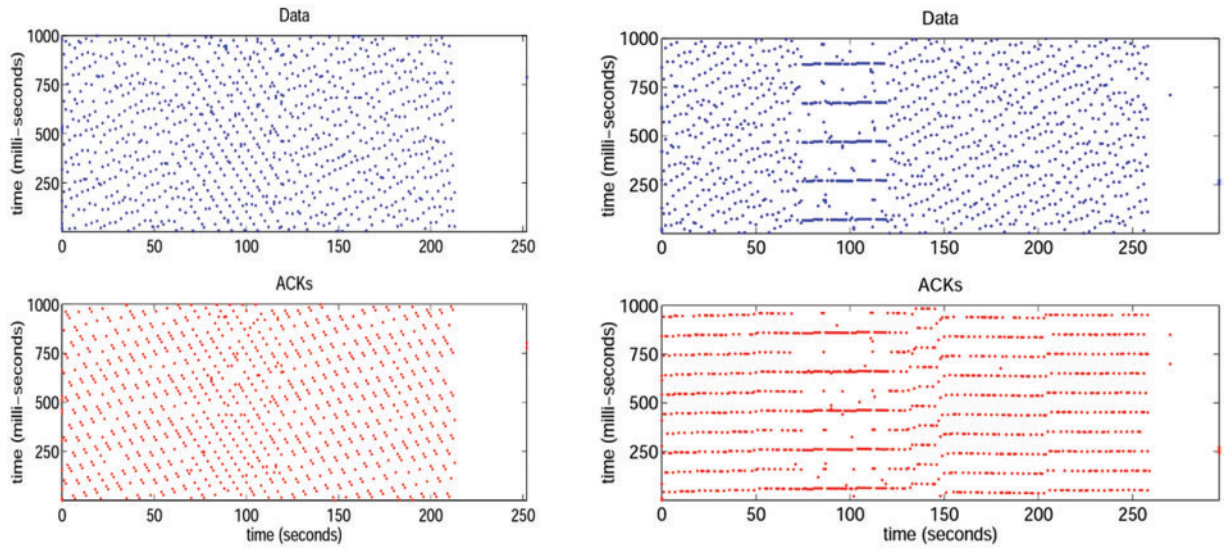

Figure 7: The properties of VPN and non-VPN congestion control for data and total time for classification

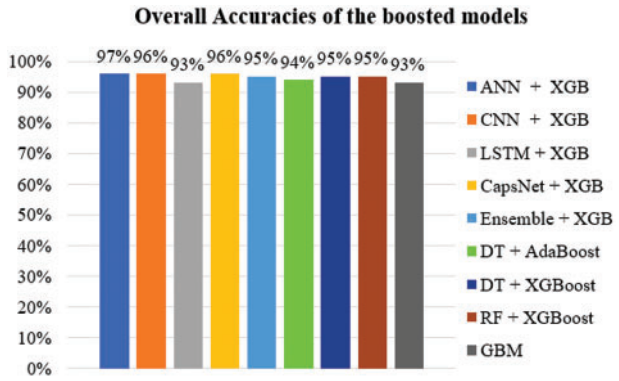

Figure 8: The overall accuracies of different boosted models in comparison with booted ANN with XGB

Table 4 shows that the model's accuracy (proportion of total predictions that were correct) was 98.79%, meaning that 98 out of every 100 traffic instances were classified correctly as either VPN or non-VPN. The model's precision (proportion of correct positive predictions) was 97.94%. This means that when the model predicted an instance to be VPN traffic, it was correct 98.54% of the time. The model's recall (proportion of actual positive instances correctly identified) was 98.54%. This means that the model correctly identified 98.79% of all actual VPN traffic instances. Finally, the F1-score, a single metric that balances precision and recall, was 96.84%.

Table 4: A measure of the ANN model's performance for VPN and non-VPN data

Metric	Value (%)
Accuracy	98.79
Precision	97.94
Recall	98.54
F1-score	96.84

5 Discussions

According to the findings of this research, an Artificial Neural Network (ANN) holds tremendous potential for categorising traffic flows. The hypothetical results presented in the table illustrate a high-performing model for VPN and non-VPN traffic classification. With an accuracy of 98.79%, the model correctly classifies 98 out of 100 instances, which suggests it has learned to generalize well from the training data to unseen data. However, it is crucial to look beyond accuracy, especially in cases where the classes might be imbalanced. The model's precision is 97.94%, implying that when the model predicts an instance to be VPN traffic, it is correct 97% of the time. High precision is desirable, as it shows that the model has a low false-positive rate, meaning it rarely misclassifies non-VPN traffic as VPN traffic. With a recall of 98.54%, the model correctly identifies 98% of all actual VPN traffic instances. High recall is also important because it indicates that the model has a low false-negative rate—it rarely fails to identify VPN traffic. The F1-score, which balances precision and recall, is 96.84%. This high F1-score suggests that the model maintains a good balance between precision and recall—it can correctly identify VPN traffic (high recall) without mistakenly classifying too much non-VPN traffic as VPN traffic (high precision). Overall, these results indicate a robust and reliable model. However, it is worth noting that even with these high-performance metrics, there is always room for improvement. Further enhancements could be achieved through techniques such as hyper-parameter tuning, using more complex model architectures, expanding or diversifying the training data, or extracting additional or more informative features from the traffic data as shown in Table 5.

Table 5: Comparison of the proposed technique with existing in terms of performance

Article	Technique	Accuracy
[32]	Support vector machine (SVM)	93.67%
[33]	Convolutional neural network (CNN)	98%
Proposed	Artificial neural network-extreme gradient boosting (ANN-XGB)	98.79%

Furthermore, observing the false positives and negatives more closely could provide additional insights. Suppose the model misclassifies non-VPN traffic as VPN (false positives). In that case, it might indicate that the features used for the classification are not unique enough for VPN traffic, or there could be an overfitting issue. On the other hand, if the model often fails to identify VPN traffic (false negatives), it might suggest that the model is too conservative in predicting VPN traffic, or the VPN patterns could be too complex or diverse to be captured with the current model and features. In 2% of instances, the model misclassified the traffic type. These errors could be random, but it is also possible that there are patterns or specific conditions under which the model consistently fails. Perhaps

there is a particular VPN protocol that the model struggles with, or maybe the model's performance decreases during certain high-traffic hours. Investigating these possibilities could uncover biases or blind spots in the model, which could be addressed in the next training iteration.

6 Conclusions

Research shows the usefulness and promise of employing Artificial Neural Networks (ANNs) to classify VPN traffic flow. Using ANN to classify VPN and non-VPN network traffic has emerged as a promising approach in network management and cybersecurity. ANNs, inspired by the human brain, can learn complex patterns from large amounts of data, making them well-suited for traffic classification. An ANN can learn to differentiate between VPN and non-VPN traffic through training based on various features such as packet size, packet duration, packet rate, flow duration, and more. The process involves data collection, preprocessing, feature extraction, model training, validation, and testing. The performance of the ANN can be measured using metrics such as accuracy, precision, recall, and the F1-score. It is important to remember that while ANNs can classify traffic, they must respect privacy and legal considerations. Although our hypothetical model demonstrated high performance, there's always room for improvement. Factors such as the training data's quality and diversity, the model's complexity, and the chosen hyper-parameters can all impact performance. In real-world applications, the model's performance should be continually monitored, and the model may need to be periodically retrained or adjusted based on changes in network environments and behaviours. Overall, using ANNs for VPN and non-VPN traffic classification is a burgeoning research area with great potential for enhancing network security and management. As with any machine learning application, a careful and thoughtful approach is required to ensure effective and ethical outcomes. The next step in this research will be determining how artificial neural networks (ANNs) might include attention mechanisms into their topologies to dynamically prioritize time-related data for categorization based on space and time's most important intervals or contexts. Using time-related factors and artificial neural networks provides researchers with the resources to classify VPN traffic and make important advances in the field. Doing so will allow us to learn more about network security and strengthen our defences against newly developed threats.

Acknowledgement: The authors would like to acknowledge the support of Asst. Prof. Dr. Sefer Kurnaz and Altinbas University, Istanbul, Turkey for their valuable support.

Funding Statement: The research received no funding grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions: Conceptualization, Sefer Kurnaz; methodology, Saad Abdalla Agaili Mohamed; software, Saad Abdalla Agaili Mohamed; validation, Saad Abdalla Agaili Mohamed; formal analysis, Saad Abdalla Agaili Mohamed; writing—original draft preparation, Saad Abdalla Agaili Mohamed.

Availability of Data and Materials: The Dataset is available in two link below https://www.unb.ca/cic/datasets/vpn.html & https://www.unb.ca/cic/datasets/darknet2020.html.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Multimed. Tools Appl.*, vol. 76, no. 21, pp. 22267–22301, May 2017. doi: 10.1007/s11042-017-4728-4.

[2] R. Song and T. Willink, "Machine learning-based traffic classification of wireless traffic," in *2019 Int. Conf. Mil. Commun. Inf. Syst. (ICMCIS)*, Budva, Montenegro, May 2019.

[3] W. Sun, Y. Zhang, J. Li, C. Sun, and S. Zhang, "A deep learning-based encrypted VPN traffic classification method using packet block image," *Electronics*, vol. 12, no. 1, pp. 115, Dec. 2022. doi: 10.3390/electronics12010115.

[4] 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 Comput.*, vol. 24, no. 3, pp. 1999–2012, May 2019. doi: 10.1007/s00500-019-04030-2.

[5] A. S. Iliyasu, I. Abba, B. S. Iliyasu, and A. S. Muhammad, "A review of deep learning techniques for encrypted traffic classification," *Comput. Intell. Mach. Learn.*, vol. 3, no. 2, pp. 15–21, Oct. 2022. doi: 10.36647/CIML/03.02.A003.

[6] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surv. Tutor.*, vol. 10, no. 4, pp. 56–76, 2018. doi: 10.1109/SURV.2008.080406.

[7] J. Cao, X. L. Yuan, Y. Cui, J. C. Fan, and C. L. Chen, "A VPN-encrypted traffic identification method based on ensemble learning," *Appl. Sci.*, vol. 12, no. 13, pp. 6434, Jun. 2022. doi: 10.3390/app12136434.

[8] Y. Wang, G. Yu, W. Shen, and L. Sun, "Deep learning based on byte sample entropy for VPN encrypted traffic identification," in *2022 5th Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng. (AEMCSE)*, Wuhan, China, Apr. 2022.

[9] B. Alsawareah, A. Althunibat, and B. Hawashin, "Classification of arabic software requirements using machine learning techniques," in *2023 Int. Conf. Inf. Technol. (ICIT)*, HCMC, Vietnam, Aug. 2023.

[10] G. Abbas, U. Farooq, P. Singh, S. S. Khurana, and P. Singh, "Feature engineering and ensemble learning-based classification of VPN and non-VPN-based network traffic over temporal features," *SN Comput. Sci.*, vol. 4, no. 5, Jul. 2023.

[11] R. T. Elmaghraby, N. M. Abdel Aziem, M. A. Sobh, and A. M. Bahaa-Eldin, "Encrypted network traffic classification based on machine learning," *Ain Shams Eng. J.*, vol. 15, no. 2, pp. 102361, Feb. 2024. doi: 10.1016/j.asej.2023.102361.

[12] N. Msadek, R. Soua, and T. Engel, "IoT device fingerprinting: Machine learning based encrypted traffic analysis," in *2019 IEEE Wirel. Commun. Netw. Conf. (WCNC)*, Morocco, Apr. 2019.

[13] T. de Toledo and N. Torrisi, "Encrypted DNP3 traffic classification using supervised machine learning algorithms," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 384–399, Jan. 2019. doi: 10.3390/make1010022.

[14] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Mobile encrypted traffic classification using deep learning," in *2018 Netw. Traffic Meas. Anal. Conf. (TMA)*, Vienna, Austria, Jun. 2018.

[15] C. Gu, S. Zhang, and Y. Sun, "Realtime encrypted traffic identification using machine learning," *J. Softw.*, vol. 6, no. 6, pp. 12, Jun. 2011. doi: 10.4304/jsw.6.6.1009-1016.

[16] Z. He and L. Wang, "Classification model for class-imbalanced encrypted traffic based on deep learning," in *2023 Int. Conf. Intell. Media, Big Data Knowl. Min. (IMBDKM)*, Changsha, China, Mar. 2023.

[17] A. P. Singh and M. Singh, "Real time malware detection in encrypted network traffic using machine learning with time-based features," *J. Discrete Math. Sci. Crypto.*, vol. 26, no. 3, pp. 841–850, 2023. doi: 10.47974/JDMSC-1760.

[18] K. Xian, "An optimized recognition algorithm for SSL VPN protocol encrypted traffic," *Informatica*, vol. 45, no. 6, Oct. 2021.

[19] A. Almomani, "Classification of virtual private networks encrypted traffic using ensemble learning algorithms," *Egypt. Inform. J.*, vol. 23, no. 4, pp. 57–68, Dec. 2022. doi: 10.1016/j.eij.2022.06.006.

[20] S. Rai and K. Pachlasiya, "Recognition and classification of traffic signs using machine learning techniques," *Int. J. Comput. Appl.*, vol. 169, no. 10, pp. 12–18, Jul. 2017.

[21] X. Zheng, X. Ma, Y. Jin, D. Gu, and R. Wang, "Tabular-based self-supervised learning approach for encrypted traffic classification," *J. Electron. Imaging*, vol. 32, no. 4, pp. 18853, Aug. 2023. doi: 10.1117/1.JEI.32.4.043032.

[22] D. T. Ergönül and O. Demiir, "Real-time encrypted traffic classification with deep learning," *Sakarya University J. Sci.*, vol. 26, no. 2, pp. 313–332, Apr. 2022. doi: 10.16984/saufenbilder.1026502.

[23] A. R. Alzighaibi, "Detection of DoH traffic tunnels using deep learning for encrypted traffic classification," *Computers*, vol. 12, no. 3, pp. 47, Feb. 2023. doi: 10.3390/computers12030047.

[24] D. A. S′omin, "Information technology for classification of encrypted traffic in corporate networks using machine learning," *Telecommun. Inf. Technol.*, vol. 75, no. 2, 2022. doi: 10.31673/2412-4338.2022.025459.

[25] S. Ramraj and G. Usha, "Hybrid feature learning framework for the classification of encrypted network traffic," *Conn. Sci.*, vol. 35, no. 1, pp. 523, Apr. 2023. doi: 10.1080/09540091.2023.2197172.

[26] L. Guo, Q. Wu, S. Liu, M. Duan, H. Li and J. Sun, "Deep learning-based real-time VPN encrypted traffic identification methods," *J. Real Time Image Process.*, vol. 17, no. 1, pp. 103–114, Dec. 2019. doi: 10.1007/s11554-019-00930-6.

[27] P. Choorod, G. Weir, and A. Fernando, "Classifying tor traffic encrypted payload using machine learning," *IEEE Access*, pp. 1, 2024. doi: 10.1109/ACCESS.2024.3356073.

[28] M. Song, J. Ran, and S. Li, "Encrypted traffic classification based on text convolution neural networks," in *2019 IEEE 7th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dalian, China, Oct. 2019.

[29] S. Miller, K. Curran, and T. Lunney, "Multilayer perceptron neural network for detection of encrypted VPN network traffic," in *2018 Int. Conf. Cyber Situational Aware., Data Anal. Assessment (Cyber SA)*, Scotland, Jun. 2018.

[30] "VPN-nonVPN dataset (ISCXVPN2016)," Accessed: Jun. 6, 2016. [Online]. Available: https://www.unb.ca/cic/datasets/vpn.html.

[31] "CIC-Darknet2020," Accessed: Nov. 9, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/darknet2020.html.

[32] X. Zheng and H. Li, "Identification of malicious encrypted traffic through feature fusion," *IEEE Access*, vol. 11, pp. 80072–80080, 2023. doi: 10.1109/ACCESS.2023.3279120.

[33] Y. Zhou *et al.*, "Identification of encrypted and malicious network traffic based on one-dimensional convolutional neural network," *J. Cloud Comput.*, vol. 12, no. 1, pp. 12588, Apr. 2023. doi: 10.1186/s13677-023-00430-w.