CLASSIFICATION OF VPN NETWORK TRAFFIC FLOW USING TIME RELATED FEATURES ON APACHE SPARK

2021
MASTER THESIS
COMPUTER ENGINEERING

Salma Abdullah ASWAD

Thesis Advisor
Assist. Prof. Dr. Emrullah SONUÇ

CLASSIFICATION OF VPN NETWORK TRAFFIC FLOW USING TIME RELATED FEATURES ON APACHE SPARK

Salma Abdullah ASWAD

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as Master Thesis

Thesis Advisor
Assist. Prof. Dr. Emrullah SONUÇ

KARABUK
Jan 2021

I certify that in my opinion the thesis submitted by Salma Abdullah ASWAD titled "CLASSIFICATION OF VPN NETWORK TRAFFIC FLOW USING TIME RELATED FEATURES ON APACHE SPARK" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Emrullah SONUÇ ....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. Jan 21, 2021

Examining Committee Members (Institutions) Signature

Chairman : Assist. Prof. Dr. Yasin ORTAKCI (KBU) ....

Member : Assist. Prof. Dr. Halil YETGİN (BEU) ....

Member : Assist. Prof. Dr. Emrullah SONUÇ (KBU) ....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Hasan SOLMAZ ....
Director of the Institute of Graduate Programs

Salma Abdullah ASWAD

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Salma Abdullah ASWAD

ABSTRACT

M. Sc. Thesis

CLASSIFICATION OF VPN NETWORK TRAFFIC FLOW USING TIME RELATED FEATURES ON APACHE SPARK

Salma Abdullah ASWAD

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Emrullah SONUÇ
January 2021, 53 pages

The aim of this research thesis is to classify the VPN network traffic flow using the time-related features on the Apache Spark and machine learning model Artificial Neural Network (ANN). This research work presented the detection and classification of network traffic flow and time related features in Virtual Private Network (VPN) with the help of ANN and Apache Spark. The proposed solution, the ANN and Apache Spark engine trained on feature of VPN with multiple nodes. Any node receives a bundle when no processing is needed, as all the routing processing has been completed already. The only task that the node is required to do is to forward the bundle to the right next-hop when the time comes, and it appears in the sight of contact. Applying the proposed ANN prevents unnecessary processing and flooding found in common VPN network traffic classification. The proposed system uses 80% of the dataset for training while 20% is used for the testing and validating with 10-cross fold validation and 50 epochs of training. This is the first study that introduces and utilizes ANN and

Apache Spark engine to implement VPN network traffic flow classification to the best of our knowledge. The categorical features classification of VPN and Non-VPN features concerning overall classification precision where VPN classification precision stands at 96.76%; however, the Non-VPN stands at 92.56%. The ANN + Spark Engine technique outperforms the convolutional neural network compared to the stacked auto-encoder network on CIC-Darknet2020 and ISCXVPN2016, respectively.

Key Words : Network, Classification, Traffic-Flow, VPN, Apache Spark, ANN, Deep Learning.

Science Code : 92414

ÖZET

Yüksek Lisans Tezi

APACHE SPARK'TA ZAMANLA İLGİLİ ÖZELLİKLER KULLANILAN VPN AĞI TRAFİK AKIŞININ SINIFLANDIRILMASI

Salma Abdullah ASWAD

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Emrullah Sonuç
Ocak 2021, 53 sayfa

Bu tezin amacı, Apache Spark üzerindeki zamanla ilgili özellikleri ve makine öğrenimi modeli yapay sinir ağını kullanarak VPN ağ trafik akışını sınıflandırmaktır. Halihazırda bugünün internet trafiğinin yarısı VPN / VPN olmayan gibi protokoller kullanılarak şifrelenmiştir. Bu, klasik derin paket inceleme yaklaşımlarının paket yüklerini analiz etmesini engeller. Son zamanlarda araştırmacılar, eğitilmiş modellerinin şifrelenmiş ağ trafiği yüklerinde kalıpları bulabildiğini ve uygulamaları bu kalıplara göre sınıflandırabildiğini iddia eden bir derin öğrenme yaklaşımı yayınladılar. Bu çalışma, bu iddianın doğru olma ihtimalinin düşük olduğunu göstermektedir, çünkü kullanılan veri kümesi, herhangi bir yük verisi eklemeden son derece doğru sınıflandırmaya izin veren özellikleri ortaya çıkarmaktadır. Bu araştırmanın uygulanması için, MATLAB 2019b, VPN ağlarının artan talebi evrimsel teknolojiyi harekete geçirdiğinden kullanılmıştır. Önerilen yapay sinir ağının uygulanması, gereksiz işlemlerin yanı sıra yaygın VPN ağ trafiği sınıflandırmasında bulunan taşmaları önleyecektir. Önerilen sistem veri setinin %80'i üzerinde eğitilirken, % 20'si 10 katlık doğrulama ve 50 eğitim dönemi ile test ve doğrulama için tutulur. Bildiğimiz kadarıyla, bu, VPN ağ trafiği akışının sınıflandırmasını uygulamak için yapay sinir ağlarını ve apache spark motorunu kullanan ilk çalışmadır. % 96,76 doğrulukla YSA ve Apache Spark Engine kullanan VPN sınıflandırması yapılmış, % 92,56 doğrulukla YSA ve Apache Spark Engine kullanan VPN olmayan sınıflandırması gerçekleştirilmiştir. Bu çalışma, paket düzeyinde şifrelenmiş trafik sınıflandırması için CIC-Darknet2020 adlı veri setini kullanan bir yaklaşımın, bir paketi belirli bir uygulamaya yüksek doğrulukla doğrudan eşlemesine izin verdiği için, paket başlık bilgilerini dahil edemeyeceğini göstermiştir. Yalnızca VPN dışı trafik dikkate alındığında, veri kümesindeki tüm paketlerin %96,76'sı bir uygulama ile ilişkilendirilebilir. Kalan paketler yine de bu akışı kullanan uygulamalara göre tahmin edilerek yüksek olasılıkla sınıflandırılabilir.

Anahtar Kelimeler : Ağ, sınıflandırma, trafik akışı, VPN, apache spark, YSA, derin öğrenme.

Bilim Kodu : 92414

ACKNOWLEDGMENT

Firstly I thank God and a lot of thanks to my supervisor, Assist. Prof. Dr. Emrullah SONUÇ, for his great interest and assistance in the preparation of this thesis

And I would like to thank my family who stood with me throughout my journey specially my father who encouraged and guided me in these two years but didn’t make it to see me graduate and a lot of thanks to my mother with her endless support

And I would like to thank my friend Sipal Atam who was with me during my study.

Finally, I dedicate this thesis to the memory of the soul of my kindhearted father Abdullah Aswad, my model in life. He taught me to live dignity and glory.

CONTENTS

Page

APPROVAL ii

ABSTRACT iv

ÖZET vi

ACKNOWLEDGMENT viii

CONTENTS ix

LIST OF FIGURES xii

LIST OF TABLES xiv

SYMBOLS AND ABBREVITIONS INDEX xv

PART 1 1

INTRODUCTION 1

1.1. MOTIVATION 4

1.2. PROBLEM STATEMENT 6

1.3. RESEARCH CONTRIBUTIONS 6

1.4. THESIS ORGANIZATION 7

PART 2 9

RELATED WORK 9

2.1. LITERATURE REVIEW 9

2.2. VPN LIMITATIONS AND CONSTRAINTS 11

2.3. DELAY TOOL NETWORK (DTN) DRAWBACKS 13

2.3.1. High Latency and Low Data Rate 13

2.3.2. DTN’s Disconnection 13

2.3.3. Unexpected Lengthened Queuing 13

2.3.4. High Error Rate 14

2.4. GRAPH ROUTING IN VPN NETWORKS 14

2.5. MACHINE LEARNING IN VPN 16

2.6. CHARACTERISTICS OF VPN NETWORKS 18

Page
2.7. ROUTING TRAINING FOR LOSS MINIMIZATION IN VPN 19

PART 3 20
METHODOLOGY 20
3.1. MACHINE LEARNING MODEL 22
3.1.1. Artificial Neural Network (ANN) 22
3.1.2. Advantages of Artificial Neural Network 23
3.2. DATASET DESCRIPTION 25
3.3. APACHE SPARK 26
3.4. IMPLEMENTATION STEPS AND FRAMEWORK 28
3.4.1. First Step 28
3.4.2. Second Step 29
3.4.3. Third Step 30
3.4.4. Fourth Step 31
3.4.5. Apache Spark Depth Experiment 33
3.4.6. Gradual Spark Widening Experiment 34
3.4.7. Activation Function in ANN with Apache Spark 34
3.4.8. VPN Cone Network Traffic Flow Experiment 35

PART 4 37
RESULTS 37

PART 5 42
DISCUSSION 42
5.1. DISCUSSION 42
5.2. ANALYSIS OF STUDY 44

PART 6 46
CONCLUSION 46
6.1. CONCLUSION 46
6.2. FUTURE RECOMMENDATION 46

Page

REFERENCES.. 48

RESUME 53

LIST OF FIGURES

Page

Figure 1.1. The classification of virtual private network (VPN) in two different types of base networks with secure remote connection. 2

Figure 1.2. The architecture of apache spark. 4

Figure 2.1. Double-hop VPN for managing the network traffic and routes of traffic from the internet. 11

Figure 2.2. The constraints that are applied on the VPN and Non VPN monitoring over the internet. 12

Figure 2.3. The hybrid DTN routing controls a high error rate for a heterogeneous network in different applications. 14

Figure 2.4. The graph routing in the VPN network assessing all the network traffic from the internet to VPN server. 15

Figure 3.1. Flow diagram of approach being followed. 21

Figure 3.2. Architecture of ANN involving input, hidden and output layers for more abstraction. 23

Figure 3.3. Evaluating the performance of network with the fine tuning and pruning. 24

Figure 3.4. The architecture of Apache Spark for the implementation and experiments on the VPN network. 27

Figure 3.5. The basic processing of VPN data through the ANN and Apache Spark to classify network traffic flow. 28

Figure 3.6. The training accuracy loss of ANN network with Spark Engine vs the number of layers. 29

Figure 3.7. The validation loss of ANN network with Spark Engine vs the number of layers. 30

Figure 3.8. The step for proposed solution by using ANN to create trained model. 32

Figure 3.9. Last step in creating the proposed VPN Network Traffic Flow Classification. 32

Figure 3.10. Solution: VPN Network Traffic Flow with ANN using Apache Spark Engine. 33

Figure 3.11. Topology of VPN that is being considered in this research work. 35

Figure 4.1. The VPN time-related features with x-axis represents the VPN features while y-axis represents the detection and classification precision. 39

Page

Figure 4.2. The classification precision of VPN time-related features using ANN and Apache Spark Engine. 40

Figure 4.3. The classification of major time-related VPN features considering all processed data at 50 epochs. 40

Figure 4.4. The categorical features classification of VPN and Non-VPN features with respect to overall classification precision. 41

Figure 5.1. The comparison of classification (Feature detection rate and recall) between ANN + Spark Engine Technique Vs. the CNN technique. 43

Figure 5.2. The comparison of classification (Feature detection precision and F1-score) between ANN + Spark Engine vs. CNN. 43

LIST OF TABLES

Page

Table 3.1. Applications and traffic types of the CIC-DarkNet2020 dataset. 28

Table 4.1. The implementation classifies 91% of the packets for given application analysis in non-VPN traffic. 37

Table 4.2. The implementation can uniquely classify 93 %of the packets for given applications. This analysis incorporates both VPN and non-VPN traffic. 38

Table 5.1. The ANN + Spark Engine technique outperforms the CNN on CIC-Darknet2020 and ISCXVPN2016 respectively. 44

SYMBOLS AND ABBREVITIONS INDEX

SYMBOLS

ρ	: correlation coefficient
σ	: standard deviation

ABBREVIATIONS

ANN	: Artificial Neural Network
BGP	: Border Gateway Protocol
CNN	: Convolutional Neural Network
DTN	: Delay Tool Network
DNS	: Domain Name Service
IP	: Internet Protocol
LAN	: Local Area Network
OSI	: Open System Interconnection
PPP	: Point to Point Protocol
SSL	: Secure Socket Layer
TCP	: Transmission Control Protocol
VPN	: Virtual Private Network

PART 1

INTRODUCTION

Virtual Private Networks (VPN) are a technology to facilitate secure communication over an insecure network (such as the internet). VPN solutions can be categorized under several types, each with their approach to security, upsides and downsides and reliance on different combinations of protocols and standards. The three major types are IPsec, PPTP and TLS as mentioned in [1] based VPN solutions. Given the complexity of VPN solutions and the fact that various implementations of these types exist, it is not determined that some of these implementations have undiscovered security vulnerabilities, as mentioned in [2]. This thesis focuses on testing one of these implementations, the TLS based VPN with Big Data based well-known technique Apache Spark for the implementation of this research work. The server replies to a connect request that contains the bind port and bind address, that is the address at which the server has connected the target server. The bind address is typically not identical to the Apache Spark server address, to which the client sent the original request. After the successful connect command, the client and the target server can communicate transparently through the Apache Spark server; this forwards any data mentioned in [3].

This work was implemented in MATLAB programming. It means that sending garbled protocol messages to an Apache Spark server to see how it responds as mentioned in [4]. By monitoring the server's behavior and the network traffic between client and server and placing this information besides the VPN messages that caused this behavior it is possible to find flaws in the Apache Spark implementation. These flaws can then potentially lead to concrete security vulnerabilities that can be exploited by an attacker sending a packet with the same mutations as the VPN messages as mentioned in [5]. To set up such a VPN network, clear knowledge of the protocol in question is needed. As one needs to be able to construct valid protocol messages before one can mutate their contents. Clear knowledge of the protocol also helps determine what fields to corrupt and how to corrupt them. Therefore, the goal of this paper is twofold. At first, the Apache Spark must be clearly mapped out as mentioned in [5]. This includes how the protocol behaves under regular operation, the different security options that may be present and the structure of the different message packets themselves so they can be manually constructed. Secondly, a VPN network must be built, which can automatically carry out the actual fuzzing attempts as we show in Figure 1.1.

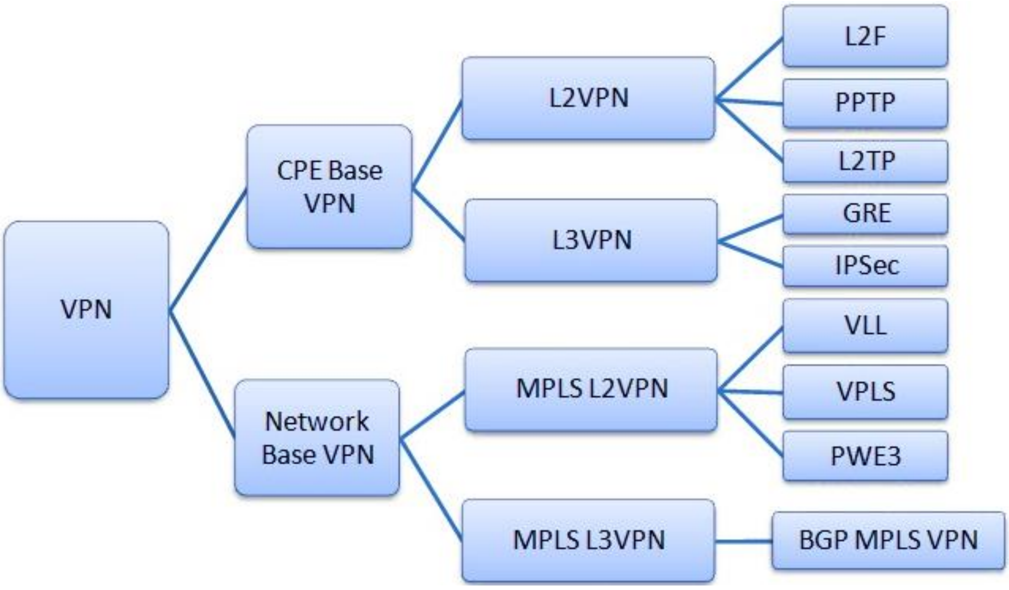

Figure 1.1. The classification of virtual private network)VPN(in two different types of base networks with secure remote connection.

The constant growth of the internet shows that we may do anything "connected" or "online" nowadays. We might do any research in any area. Products are sold over the internet, a market that tends to become more expressive every year as mentioned in [7]. Social Networks have been highlighted as one of the most significant time-consuming activities, being daily used by millions and millions of people worldwide. Financial services such as, buying stocks or making a transaction in accounts in a network located in another country or continent. The research community used a linked connection and several other "remote connection" with possible some decades ago. They are so common in our lives that we do not realize how important and essential they are for a significant part of the companies as mentioned in [8].

Further, a huge amount of information is stored worldwide on network traffic flow servers over the internet. Information is in our hands and we can access everything from our smart devices, in some cases, even in a smartwatch as mentioned in [9]. However, in this sense, to find some specific information, we have to search for it. Before the internet, we used the VPN network for proxy the network. Nowadays, the idea is the same with the internet, but with technology, algorithms, search engines, etc. We use the same idea, the same "VPN" of contents to find some network traffic flow in a dedicated network, it is in that way how the search engines work.

A search engine's main idea is to classify the VPN information from some network traffic flow in a dedicated network and present it in the future to any user looking for something related. In our research, we will not cover this indexing process, but the idea is very close, to gather the information, as we have done, it was necessary as mentioned in [10]. We set up an environment to collect every network traffic flow page from some defined domain to build our dataset. In order to work with this information as many times as we want, without having to access the network traffic flow site every time and most importantly, not taking the risk of this information to have been changed in the meantime.

The first tool applied in this step is a network traffic flow crawler, a tool that gathers information, content and the whole structure of a determined set of network traffic flow pages. We are going to talk briefly about the whole process later in this chapter and also deeply in a more specific section.

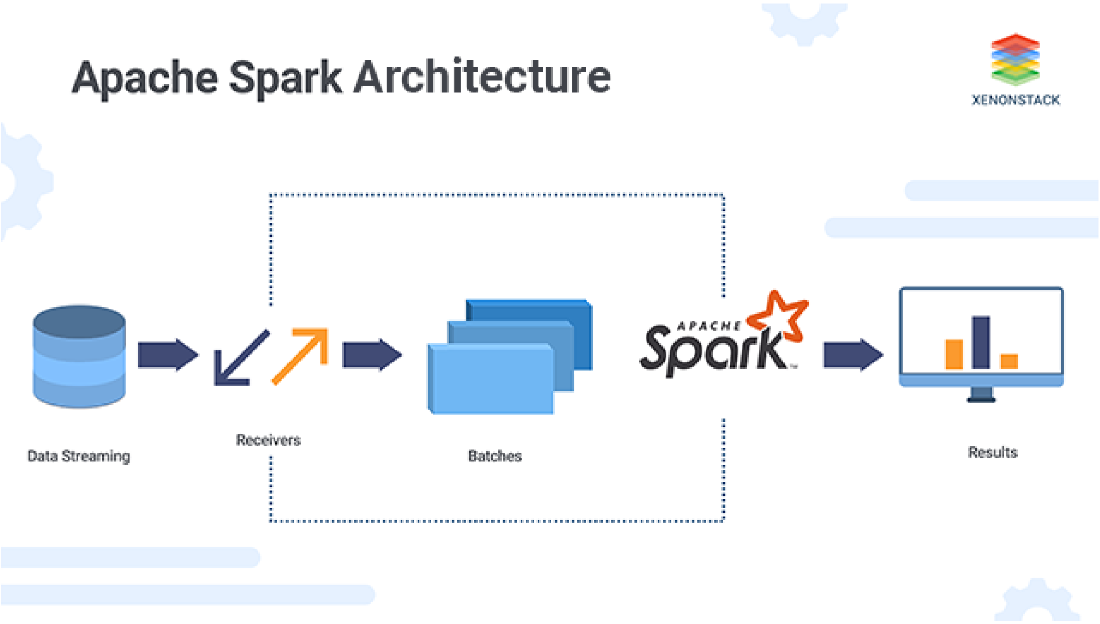

Figure 1.2. The architecture of apache spark.

Besides building a dataset, which is going to be well explained in this work, we have also written some code to extract information from those files, the features to analyses as mentioned in [11]. To do so, we have used the Apache Spark environment, a very concise and powerful framework for Big Data problems. However, we only used a small portion of data initially but increased the amount over time. A chapter is dedicated to cover the Apache Spark framework.

Finally, the last process was the application of Machine Learning techniques, still in the Spark environment. The framework is built on top of libraries, including a Machine Learning library, and is therefore especially designed and developed to deal with Big Data problems. Techniques are broadly known and used, e.g. cross-validation for model selection, classification algorithms and evaluations were applied in practice and we introduce it in the end.

1.1. MOTIVATION

Many researchers have been studying and publishing works in the field of network traffic flow with content mining. Besides, the VPN Network Content Analysis area may be more comprehensive than we think and might express any analysis. One of the most essential papers has been published by researchers in [12]. The area's explanation is very well covered, introducing since from the beginning to the Network traffic flow Content Analysis. The authors highlight in their study, one of the first applications of "pure" Content Analysis which has records, occurred in the 20th century, when the companies have examined amounts of early networks to analyze their contents in a systematic way that presented in [13]. According to researchers in [14, 15], the Content Analysis in VPN networks to analyze a mass of information about communication and classification of VPN networks has been done by the machine learning techniques for effective analysis.

In [16], the authors introduced an experimental study about Network Content Analysis for a set of networks with smooth traffic flow for classification from researchers in the United Kingdom and America. Their study aimed to investigate the users' behaviors to find patterns or preferences in the dataset. They conclude that the application of Network traffic flow Content Analysis is an easy process and may produce very good results in the end, depending on the analysis field.

In [17], researchers introduced Network traffic flow Content Analysis focused on news articles network traffic flow pages, which is our idea with the difference linked to the tools and frameworks we have used. A tool to extract information from those news articles without analyzing them using machine learning focuses more on Natural Language processing. Research in [18] is used as a base for our study to extract information in VPN (Virtual Private Network) classification, which classifies the VPN code of the different networks and nodes. The former approach was the same as the authors present in their paper [19]. They published a paper explaining how they achieved outstanding results using the VPN classification.

Furthermore, in [20] and [21], they have obtained promising results, which based their VPN classification study from news big databased research. Both introduce the machine learning techniques used to evaluate their model and relate it with our present study. In the end, both studies come up with a conclusion about the needs and motivations to study the subject, especially regarding the growth of the internet, which requires each time more precision of the machine learning applications, such as the search engines.

1.2. PROBLEM STATEMENT

The connected devices are facing an immense growth on internet day by day. Communication between computers is also growing in form of email, instant message, remote control, and so on. Companies also use this kind of service and delicate data is exchanged. Today, companies don't have just one Local Area Network (LAN); they have different LANs at different locations globally and need to communicate with each other securely. For this, a secure channel is a priority factor when growing the companies' computer networks all over the world. These secure channels between LANs create a kind of global private network which physically doesn't exist but acts like one, which is a Virtual Private Network (VPN). There is no formal design for the advancement and classification of VPN traffic flow using time-related features on the Apache Spark, a Big Data framework. We want to develop a prototype representing the VPN classification and communication between computers to create a final VPN product using Apache Spark. This VPN communication prototype will not only succeed but will also achieve certain security aspects related to traffic flow. The research community is working on classifying the Virtual Private Network (VPN) using Machine Learning based technology. In response to that, this research aims to classify VPN traffic with machine learning and big data-based Apache Spark engine.

1.3. RESEARCH CONTRIBUTIONS

This thesis has described VPN traffic flow classification using a communication prototype by Big Data based Apache Spark using different protocol. This research has created an open secure channel that crosses the internet and all the scenario's connections. The machine learning based prototype has been capable of sending, in a secure way, socket messages through this channel without affecting the traffic flow and previous security restrictions, proving that a basic socket architecture of the desired final VPN product is reached. Many possible applications do use this channel in the future because Apache Spark is a generic framework and protocol, meaning that several application protocols can run over it. A great part of the thesis is concentrated on the implementation of VPN classification and traffic flow communication, security aspects like authentication and the anonymity of internal nodes are implemented at the prototype/application. The whole prototype is written in MATLAB R2019a for commercial and interoperability reasons with Machine Learning and Processing Toolbox. Our research's main contribution is to study a reasonable way to build a model for the classification of VPN networks to analyze the content of a set of networks using the Apache Spark. It has to comprise the whole process since building the dataset to evaluate the results. Thus, we come up with those research questions:

1. Is it possible to build an Apache Spark enabled Machine Learning model to extract the VPN network's information and analyze them to classify the best content part in a network?
2. If so, how could we evaluate the performance or the precision of the model?
3. Is it possible to apply the model to all types of VPN networks?
4. What could be the best collection of features for analyzing and classifying the VPNs in this research?

Before deciding whether or not a VPN network is detected or classified by this proposed system, we should be supported as a more advantageous technological solution. These questions should be answered. They present decision-makers and the public with the classification and advantages and disadvantages of each classification with Apache Spark. It should be emphasized that this study only focuses on the classification of VPN network and traffic flow management and classifying it with the Apache Spark is very important. Therefore, this research work is only for VPN classification, although the methodology used here is applicable for other technological alternatives.

1.4. THESIS ORGANIZATION

In the first chapter, background and motivation, research question, methodology and outline are presented. The virtual private network and network traffic flow challenges are presented with a specific emphasis on the dissertation's introduction with dependencies of a problem statement and research contribution. The chapter ends with a description of the characteristics of the organization for this thesis.

The second chapter gives a basic description of the literature review on VPN networks and network traffic flow, beginning with a small historical note, defining the commercially available types of available solutions. Major drawbacks are classifying the normal network for smooth traffic flow and the VPN network in large scale organization.

The third chapter presents VPN classification with the methodology background; the concept of machine learning and big data with networking concepts is presented, followed by the methodology for classification techniques, a characterization of the virtual private network market. The method for classification and reducing the cost of classification with Apache Spark with cost analysis, the purpose of this chapter, go through a sensitivity analysis and are presented and discussed.

The fourth chapter in terms of structure is very similar to the third chapter but for the dissertation result. The aspect of this work is the classification of VPN network traffic flow using time-related features on Apache Spark. A typical framework outline is followed: at first, the concept of classification results is described, then the objective and scope of the analysis are being found, followed by data collection for classification of the virtual private network.

The fifth chapter discusses the results with analysis and their respective discussion based on the limitations and comparison with other existing systems on VPN network traffic flow classification using time-related features on Apache Spark.

The sixth chapter presents the main conclusions of this dissertation with a summary of the main findings, the consequences these results have in policymaking, followed by this study's main limitations and ends with leads for future work.

PART 2

RELATED WORK

This thesis is motivated by overcoming the challenges and difficulties that VPN classification has due to their structure and the environmental hostility surrounding them by creating a better routing solution. This solution avoids the shortcomings of the traditional routing protocols controlling VPN operations. For the sake of explaining the motivation of this thesis work, a detailed review of all these difficulties and constraints is discussed in depth in the sections below; furthermore, a comprehensive review of VPN routing protocols and their disadvantages that this thesis work aims to overcome.

2.1. LITERATURE REVIEW

Communication networks are not only spreading rapidly all around us, but they are also getting more advanced in the method they are handling their connectivity. Most of the current networks use protocols like Transfer Control Protocol (TCP)/Internet Protocol (IP) and User Datagram Protocol (UDP), and those require a virtual end-to-end connection between the two endpoints as mentioned in [22]. The infrastructure for such networks is usually well organized, maintained, and located in secure and environmentally controlled spaces. That makes the traditional communications networks eligible to have powerful devices and reliable power sources. Furthermore, and due to all those factors, redundancy is one of these networks' main characteristics. Powerful devices, reliable power sources, environmentally controlled secured spaces, and a wide range of shareable routes enable these networks to have high communication reliability, fast end-to-end data rates, and fewer error rates. Traditional communication networks have lower error rates, but when an error happens, they are quicker to detect errors and send back data to recover the lost data as the case in TCP/IP or to continue transmitting data as in UDP. Unlike TCP/IP, UDP keeps sending data to the destination in real time mostly without user-felt service interruption. In this protocol, continuity of streaming is the goal rather than an utterly error-free service experience TCP/IP as mentioned in [23].

Most research and digital solutions use human-written codes to solve present problems and time related features predictions. A code is just a series of commands written by a human being to be executed on a computer. Having the code written by programmers means that they reflect and follow their chain of thoughts and state them as specific fixed steps to solve the problem as best of their knowledge can provide. As human beings intend to achieve more than ever before, having traditional tools that are already hand-made to reach goals behind their imagination is becoming an obstacle. In other words, hand-made codes are reaching their limit to achieve goals as big as human beings' dreams. We are witnessing a stage of a need for more powerful solutions than mostly fixed hand-made steps of commands, but executing these codes is getting slower as the problems to be solved are getting more complex and the codes are executed sequentially.

For most problem-solving cases, code solutions represent a bottleneck to achieve bigger scientific goals as those solutions count on relatively-slow human interactions to keep working. Not only faster processing solutions are needed, but also autonomous ones. That is when Machine Learning (ML) comes into place. Machine learning is a technology that uses machine speed to solve problems autonomously. One important branch of machine learning is ANN. ANN is the technology that deploys the high machine speed and mimics the human brain to solve problems and performs vast amounts of calculations in parallel rather than the traditional sequential codes method. This research uses ANN to deploy it to perform VPN routing specifically. The ANN will be taught a database of previous patterns of network node CG. In other words, the ANN learns when nodes have a direct line-of-sight of each other. After training the network, a network node should make the best decisions about the best next-hop to reach the destination with a minimum acceptable error rate.

Contacts are the nodes as the VPN network consists of different traffic patterns. Depending on whether the network nodes are having previous knowledge of other nodes or not, we classify VPN contacts into two types: Scheduled or predictable Contacts (Deterministic): these are the contacts that get encountered according to a certain pattern. Intermittent Opportunistic Contacts: these are the contacts that are non-predictable. This case applies to disasters where users are completely disconnected from the rest of the world, and they don't know when they have a connection using, for example, an emergency network node. To be able to provide end-to-end connectivity across heterogeneous networks, VPN relies on creating a "Bundle Layer" between two layers of the Open System Interconnection (OSI) model, and those are transport and application layers [24]. Double-hop VPN for managing the network traffic and routes of traffic from the internet as we show in Figure 2.1.

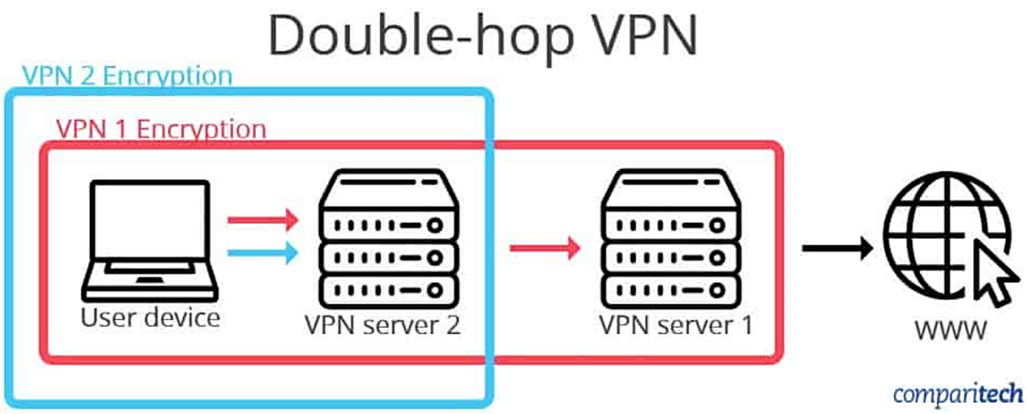

Figure 2.1. Double- hop VPN for managing thenetwork traffic and routes of traffic from the internet.

2.2. VPN LIMITATIONS AND CONSTRAINTS

VPN networks are used when power consumption and direct connection are not always available; for example, a VPN is used in deep space and animal monitoring [26]. In general, VPN nodes perform two tasks, store the bundles and forward a copy to one or more nodes. VPN protocols differ from each other in the methodology of spreading the bundles and the criteria they use to decide that.

Exotic Media Networks: that includes near earth and Inter-planet satellite networks, these are the types of networks that are spread among earth, satellites, planets, and aircraft. This network type usually produces big amounts of data that need to be sent back and forth to the earth's space bases. The internet service on this network is called Interplanetary (IPN) Internet [27].

Sensors/actuator networks: Sensors have limited resources and work periodically to sense and send data; this network is widely used in animal monitoring aforementioned [28].

Military networks: These are the networks in which nodes move with the military wherever it goes, these networks are designed to keep working even if some nodes are lost under action as we show in Figure 2.2.

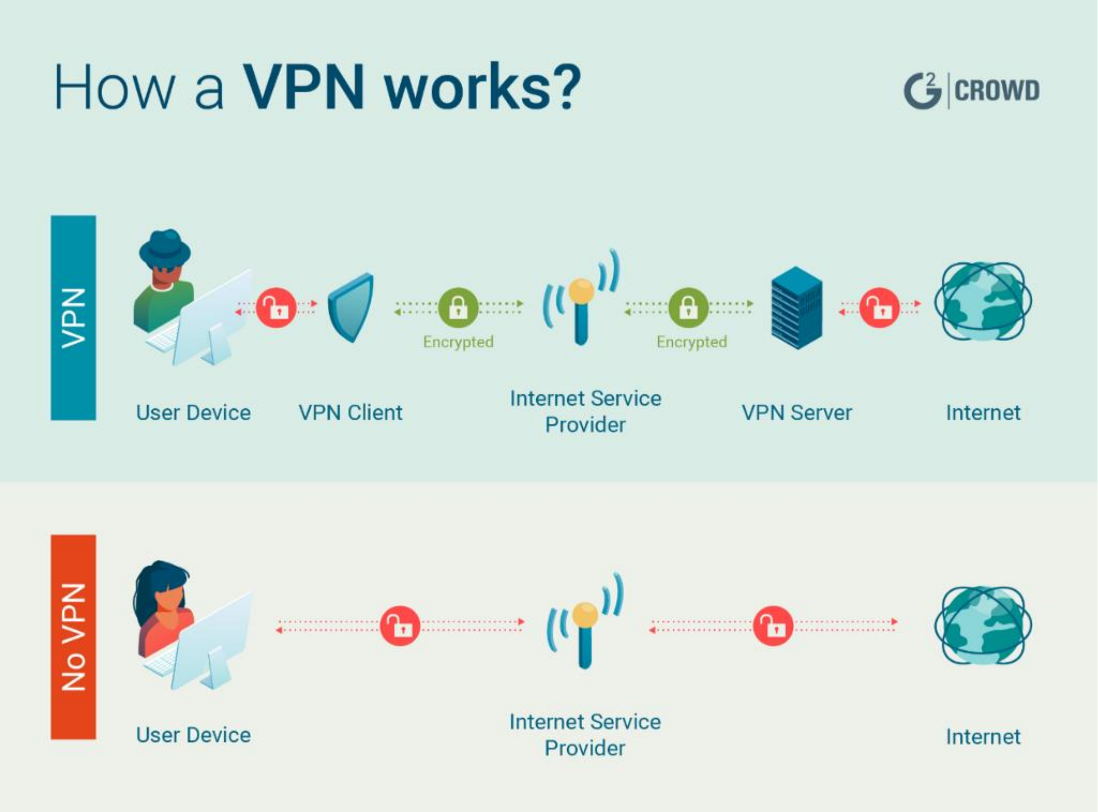

Figure 2.2. The constraints that are applied on the VPN and Non VPN monitoring over the internet.

2.3. DELAY TOOL NETWORK (DTN) DRAWBACKS

Unlike advanced VPN networks; DTNs face many challenges like delays, higher error rates, and instability. It is good to mention some of the DTNs characteristics that these challenges produce.

2.3.1. High Latency and Low Data Rate

In addition to processing and queueing delays, DTN links suffer from propagation delays caused by the transmission medium. One example of that is underwater communication; despite that some links can only carry a speed of 10 kbps, the accompanied link delays could reach one second as mentioned in [30]. Another consideration of link delays is that DTNs usually face asymmetries between the up and downlinks due to the continuous change in the relative nodes' locations. If we take for example, deep space, planets, and satellites are on continuous movement; thus a path that consists of certain sequences and link properties to send a data signal has not been the same path to receive the response. To be more accurate, most of the time if the sequence is the same for sending and receiving the data, links in that path could never match between up-link and down-link only if all nodes stationary, which is likely not the case.

2.3.2. DTN’s Disconnection

Disconnection happens in DTNs due to many reasons. It could be because of a faulty node. Other reasons could be due to nodes' movement, which could be a random walk to single or both nodes of each link or could be periodic due to satellites and planets' movements on orbits. It is also good to mention that some cut-offs occur due to the nodes' periodic power saving, like low power nodes in sensor networks.

2.3.3. Unexpected Lengthened Queuing

In conventional networks, queueing delays could be parts of seconds to seconds in extreme cases. But due to the abnormal circumstances of DTNs queueing delays could reach days if not hours. What contributes to those longer queueing delays is the nature of DTNs, where many copies are traversing among the nodes, not to mention the retransmission that takes place when data cannot reach their destinations.

2.3.4. High Error Rate

Due to node limitations and a hostile environment, the error rate is much bigger than traditional networks. The services and applications in DTN should keep working even with a high Bit Error Rate (BER) undermining the instability network [31]. as we show in Figure 2.3.

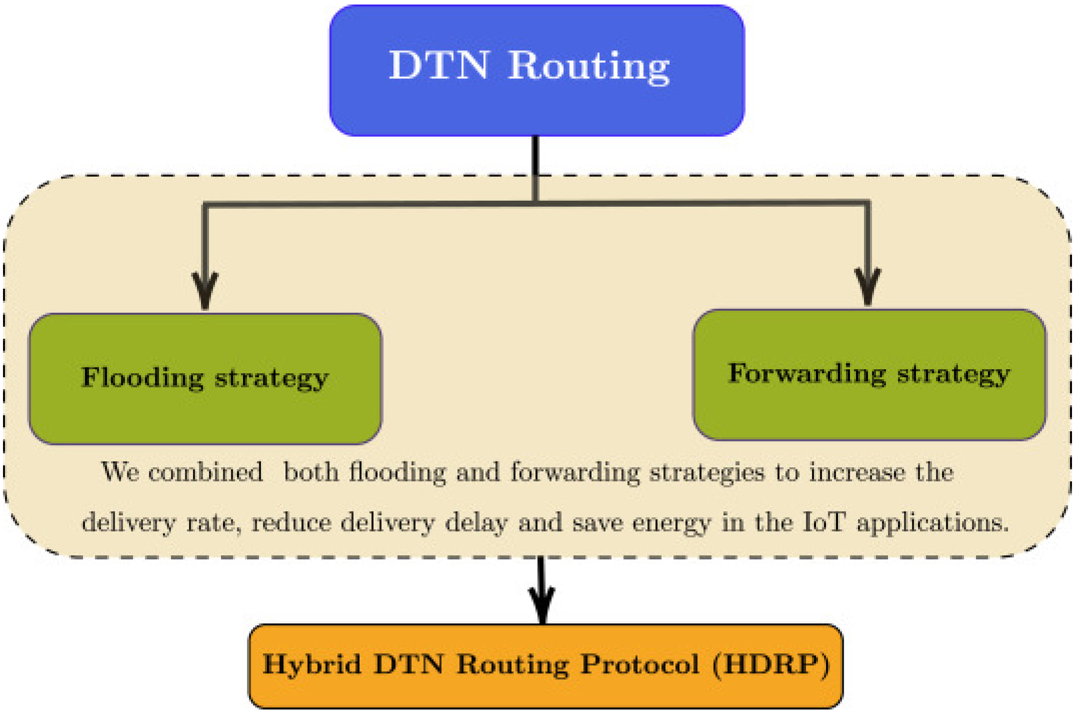
DTN Routing
Flooding strategy
Forwarding strategy
We combined both flooding and forwarding strategies to increase the
delivery rate, reduce delivery delay and save energy in the IoT applications.
Hybrid DTN Routing Protocol (HDRP)

Figure 2.3. The hybrid DTN routing controls a high error rate for a heterogeneous network in different applications.

2.4. GRAPH ROUTING IN VPN NETWORKS

Graph Routing (GR) is a time-varying algorithm that counts on deterministic timed contact periods in VPNs to find the best routes. Despite great previous knowledge of the node's communication plan using historical facts, GR needs heavy calculations to find the best routes. GR has many advantages that no other routing protocol has, some of these advantages are in the following.

Previous network knowledge: GR has prior knowledge about future topologies, as technically most of the nodes exist, yet the topology change is due to lose sight of contact or periodic power cycle.

Ahead of time readiness: Another significant advantage that other routing solutions lack is that the routing data are propagated way ahead of time before they need to use them rises. In VPNs, topology changes are slower than the speed of data spreading around them, which propagates a considerably accurate time-based routing data as given in [33].

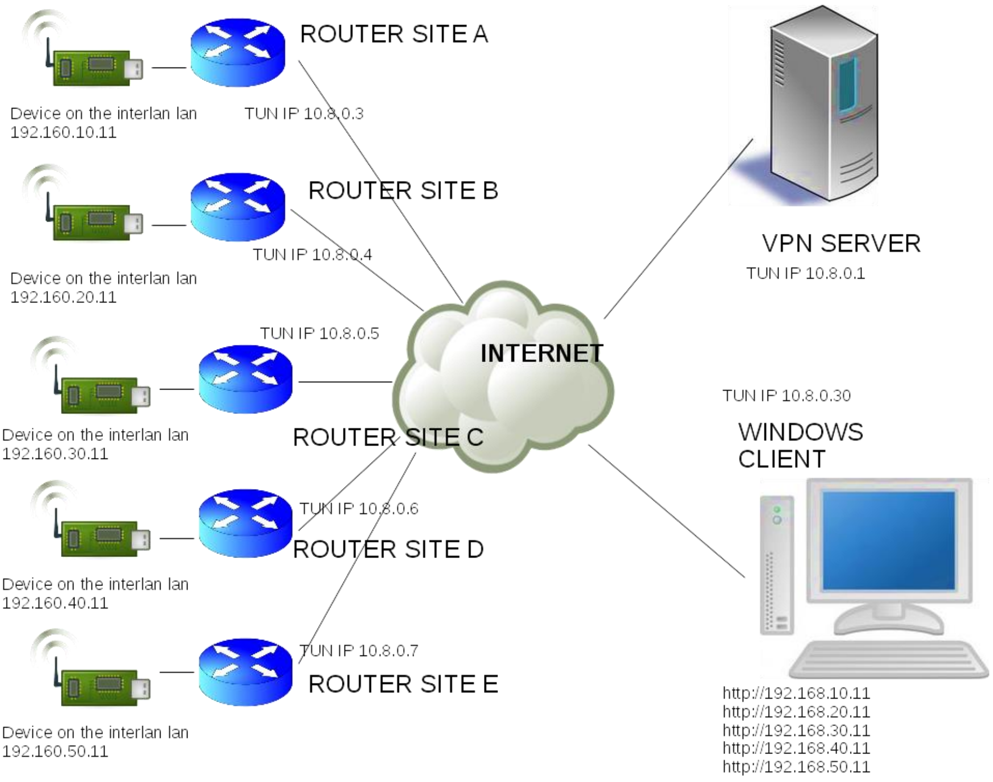

Figure 2.4. The graph routing in the VPN network assessing all the network traffic from the internet to VPN server.

In Figure 2.4 the graph routing in the VPN network assessing all the network traffic from the internet to VPN server [35] addresses solving VPN classification routing problems by presenting enhancements to GR. This work's importance is related to its addressing of planned travel of spaceships, which covers the lost communication and delay in many places of the trip as mentioned in [36]. This research proposes a GR for such time-varying topology; it uses it to make a list of routing plans for each stage of the space trip. The contact plan contains routing plans related to each period of connection change (T1 and T2) and nominal transmission rates (R). Routing table plans built for each stage of the trip using Dijkstra [37] Algorithm. When it comes to accuracy, GR could be compared to IP.

This work provides two modifications for GR; the first one is (Short Term Evolution) when contact happens. A problem occurs if the neighboring node does not already have a load in the buffer, thus delaying the transmission. Due to the priority of having communications done within (T2-T1), GR-Earliest Transmission Opportunity (ETO) is proposed [38]. (T2 and T1) to be replaced with the ETO contact parameter during contact graph traversals. The problem of estimates by other nodes solved by using the Contact Plan Update Protocol (CPUP). The second modification is Long Term Evolution, using Path Encoding GR Extension as mentioned in [39]. It takes the calculated path and attaches it to the message; thus, nodes do not need to do complex path calculations. The paper also suggests using GR extensions not only for deterministic scenarios but also for opportunistic ones. Paper [40] describes four experiments over space networks Deep Impact Network Experiment (DINET) using Interplanetary Overlay Network (ION) software on spacecraft EPOXI. It took four weeks of flight testing DTN, another experiment of JAXA DRTS. Testing in co-operation with the National Aeronautics and Space Administration (NASA) to evaluate DTN and Corporate Governance Recommendation (CGR) (performance of Bundle Protocol (BP), Licklider Transmission Protocol (LTP), and CGR was tested). The third experiment was (Space Data Routers) to improve the dissemination of space mission data concerning volume, timelines, and continuity.

2.5. MACHINE LEARNING IN VPN

The work conducted by researchers in [41] focuses on developing an ML algorithm for VPN classification. This work tried to merge two cognitive learning methods, reinforcement learning and Bayesian learning, to create a new solution that still holds GR's advantages and has more adaptability to VPN changing topologies. Researchers in [42] discussed Mobile Ad hoc Networks (MANET) and they explain that MANETs are the best options to support first responders after disasters in disastrous areas and used with private network. They also explained that MANETs would work well within each group of first responders, yet they suffer from intermittent connectivity among separated groups in private network i.e. VPNs. The authors used the Opportunistic Network Environment (ONE) mentioned in [43] and fed it realistic disaster scenarios data on both first responders movement mentioned in [44] and traffic characteristics in VPN as cited in [45]. Researchers in [46] considered movement characteristics, traffic characteristics as messages spread among groups each with different source-destination for virtual private networks. Results showed enhancement of delivery delay by reducing it by 40% of the same situation without the VPN networks.

In [47], researchers did specific research on end-to-end encrypted traffic classification with one-dimensional convolution neural networks to solve present problems and future traffic predictions. A code is just a series of commands written by network traffic being executed on a computer using one-dimensional CNN. Having the research done with encryption means that they reflect and follow their chain of thoughts and state them as specific fixed steps to solve the problem as best of their knowledge can provide securely. As network traffic is considered for achieving more than ever before, having traditional tools that are already hand-made to reach. An obstacle that slows down the whole scientific progress; in other words, CNN is reaching their limit to achieve goals as big as network traffic classification. CNN is witnessing a need for more powerful solutions than mostly fixed network traffic steps of commands, but executing these steps is getting slower as the problems to be solved are getting more complex and the network traffic encryption is executed sequentially.

Researchers in [48] worked for classifying on an enormous scale for encrypted communication of semi-supervised encrypted traffic using deep convolutional generative adversarial networks (GAN). GAN solutions represent a bottleneck to achieve bigger scientific goals as those solutions count on relatively-slow traffic interactions to keep working. Not only faster processing solutions are needed, but also autonomous ones. That is when Deep Learning (DL) comes into place. Deep learning is a technology that uses machine speed to solve problems autonomously. One important branch of deep learning is GAN. GAN is the technology that deploys the high machine speed for traffic encryption and mimics network traffic to solve problems and performs vast amounts of calculations in parallel rather than the traditional sequential codes method with encryption.

2.6. CHARACTERISTICS OF VPN NETWORKS

Limited Endurance: Since VPNs are made to cover spaces where traditional networks cannot work; they usually spread around hostile environments. Some of the environments could be a desert where an army moves or the wilderness where tracking devices are attached to some animals. It is important to achieve connectivity in such harsh environments. VPN nodes' limitation delegates the delivery assurance to nodes that rely just outside VPNs where nodes are provided with better power supply and maintenance.

Low Duty Cycle Operation: Due to power limitations, most of the VPN nodes are turned on periodically to preserve power. VPN nodes turn on and off promptly just in enough frequency and time length to keep the end-to-end network connectivity while saving power as much as possible for the longest possible timespans.

Limited Resources: To keep nodes working the longest, not only low duty cycle type of operation is vital, limiting the resources of the node itself can make a lot of difference when it comes to saving power. Two nodes under the same circumstances and duty cycle do not last for the same time if one of them uses less power to function, that is by having, for example, less memory, transmission range, and power of processing unit.

VPNs' special thing is that they don’t have a systemized architecture because the focus of building them is still on producing more reliable connectivity rather than standardizing many small networks under one umbrella.

2.7. ROUTING TRAINING FOR LOSS MINIMIZATION IN VPN

The routing training could be done anytime and as early as desired, which could be way ahead of the time to receive any data bundle. Not only the proposed solution processes could be performed on the early stages, but it could also be synchronized with nodes’ periodic power cycling. Furthermore, the proposed solution processing could be done when nodes are idling.

Regardless of time insignificancy on the proposed solution as discussed above, this research is already considering time as one of its essential metrics to decrease. The time needed to train the proposed neural network is around 120 seconds only. The proposed neural network has 27,052 trainable parameters, which layouts the proposed ANN structure with big data as this research predicts. The dataset used for this experiment consists of (600,000) data entries, having the same structure of datasets used in the previous experiments, that is six inputs and two outputs. The inputs represent the VPN as mentioned earlier time intervals, and the output consists of two values representing the best-chosen route for the data bundle.

PART 3

METHODOLOGY

This chapter discusses smart systems in general, including Apache Spark and ANN which are used later in the thesis work. For this purpose, MATLAB R2019a platform is used with both CPU and GPU utilization. The flow diagram has been shown in Figure 3.1.

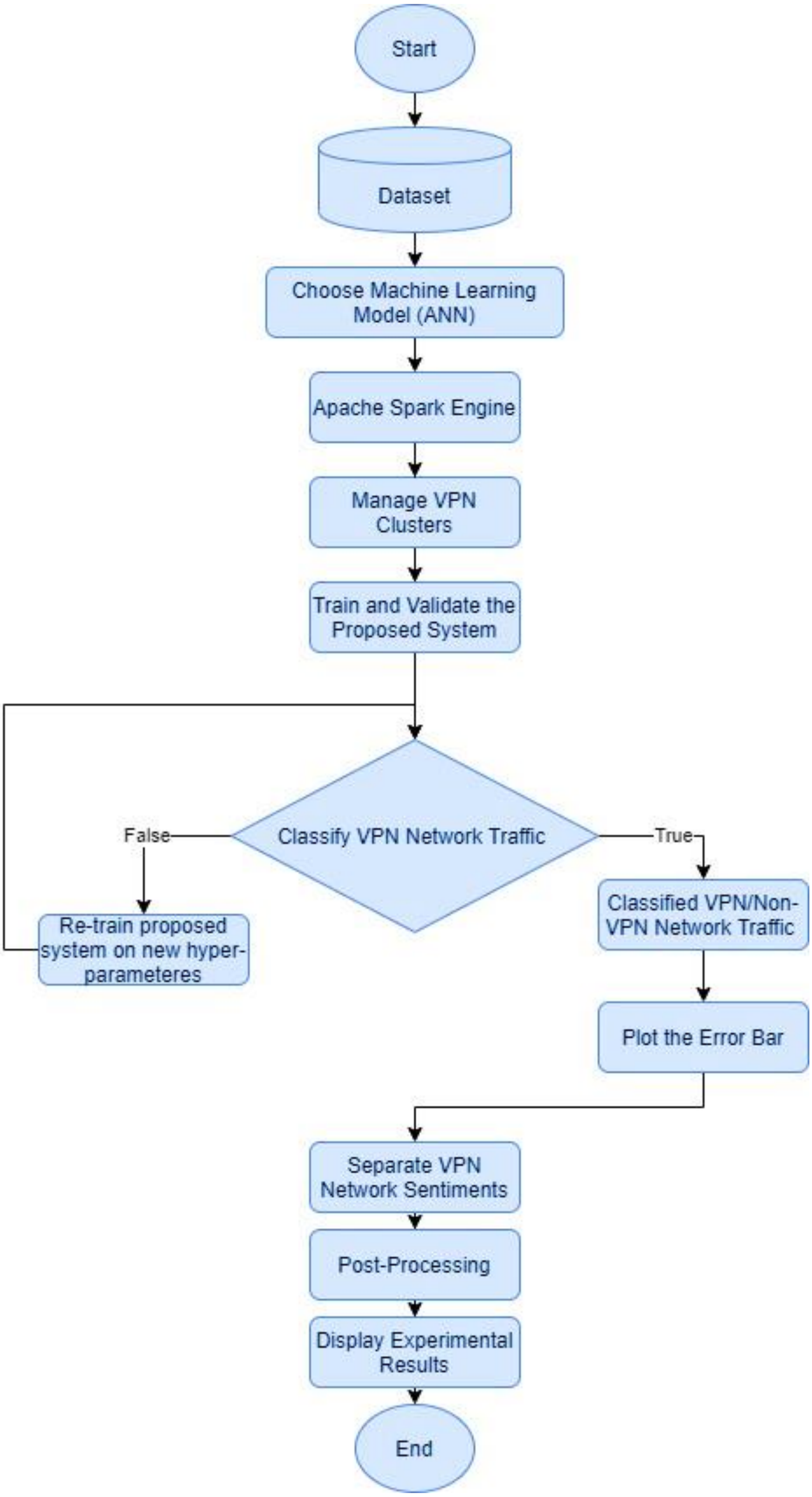
Start
Dataset
Choose Machine Learning Model (ANN)
Apache Spark Engine
Manage VPN Clusters
Train and Validate the Proposed System
Classify VPN Network Traffic
False
True
Re-train proposed system on new hyper-parameteres
Classified VPN/Non-VPN Network Traffic
Plot the Error Bar
Separate VPN Network Sentiments
Post-Processing
Display Experimental Results
End

Figure 3.1. Flow diagram of approach being followed.

3.1. MACHINE LEARNING MODEL

Machine learning is a sub-type of artificial intelligence that counts on algorithms to learn and then predict machine learning uses database statistics. The main reasons for using machine learning are:

1. Solve a problem: problems that machine learning can solve are many. Maybe a business problem, marketing, profit, reduce time, or produce more.
2. Technical achievement: many companies compete through the achievements they make by using machine learning, like the recent breakthrough that Google achieved by beating the world's champion in the Go game.

This work has both the motivations to solve the problems that VPN classification uses Apache Spark and Machine Learning and create a technical achievement in that field. Coding is reaching its limit to implement bigger scientific goals; autonomous solutions are needed.

3.1.1. Artificial Neural Network (ANN)

A human brain is an advanced machine; many people can distinguish effortlessly that the networks. The brain's powerful capability to comprehend the images is due to the 140 billion neurons. How about when we want to find a hand-made machine that can do such tasks? That is when ANNs come in. ANN is a machine designed to model how the brain performs a particular task or function of interest. Neural networks are the artificial copy of the human nervous system where artificial neurons are used to mimic human brain operations. ANNs use mathematical formulas to perform operations on inputs to give the desired output. In general, what ANNs take the input data, apply the designed calculations, and provide an output. These networks keep repeating the calculations process to learn more and get closer to the desired result by reducing the error between the produced output and the desired one; this process is called "Training." as we show in Figure 3.2 Architecture of ANN involving input, hidden and output layers for more abstraction. The input of ANN model is number of features however the outputs are the classification of VPN traffic flow with time-related features.

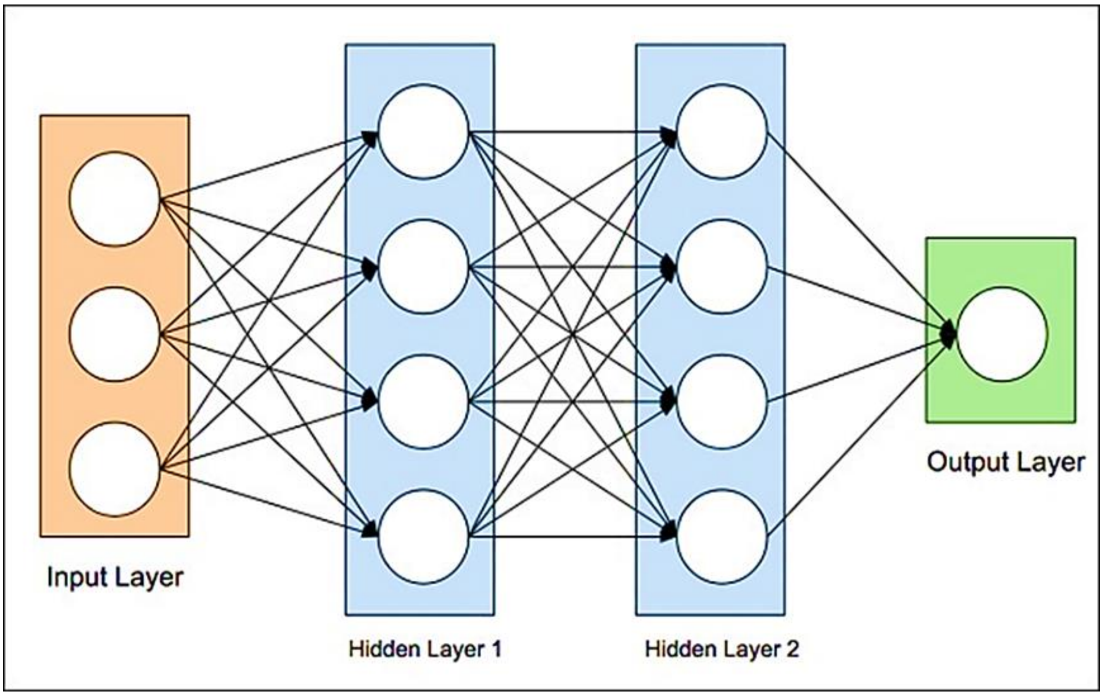

Figure 3.2. Architecture of ANN involving input, hidden and output layers for more abstraction.

3.1.2. Advantages of Artificial Neural Network

It is important to realize that neural networks have many advantages, such as:

1. Nonlinearity: Due to the structural shape of ANN, they are nonlinear themselves. Furthermore, this nonlinearity gives neural networks the ability to adapt to solve more complex problems.
2. Input-Output Mapping: Neural networks have a vital feature, that is their ability to learn a specific pattern of inputs as well as a related output pattern or (samples) which represent the desired output for each information set fed to the input layer, in other words mapping each input set to an output set, this is called supervised training.

3. Adaptivity: This advantage is the soul of ANNs. Neural networks can adapt their synaptic weights with the change of the environment they are put in, changing their weights according to the data fed to them.
4. Fault Tolerance: When it comes to implementing neural networks using hardware, they are fault tolerant. Neural networks can keep giving relatively correct output to some extent when neurons themselves get damages and when links get damaged. In like manner, a network results in abnormal output only when the damage reaches substantial magnitude.
5. Uniformity of Analysis and Design: ANNs have a universal processing method with small adaption. This feature makes it easier to understand, design, and implement neural networks to solve problems and find solutions.
6. Neurobiological Analogy: Neural networks principle is the human brain, and its goal is to achieve whatever a human brain can. Moreover, the human brain is living proof that learning and parallel processing are an excellent strength to possess and use to find unsolved problems.

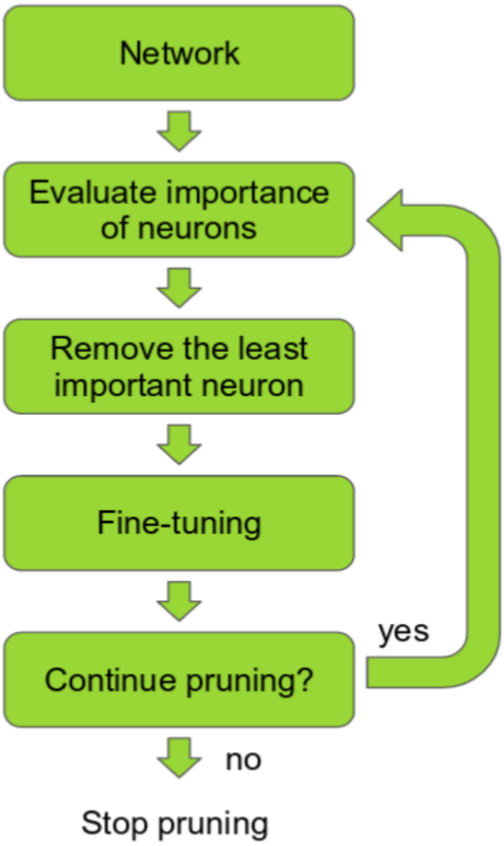
Network
Evaluate importance of neurons
Remove the least important neuron
Fine-tuning
Continue pruning?
yes
no
Stop pruning

Figure 3.3. Evaluating the performance of network with the fine tuning and pruning.

In Figure 3.3 Evaluating the performance of network with the fine tuning and pruning Despite the similarity of the basic principles of how ANN work, they differ in many ways. ANNs could be categorized according to their learning, depth or number of layers, connection numbers, connection endpoints, and many other distinctive factors. Major types of ANNs are discussed in this part. Neural networks could be categorized depending on the way neurons are connected within.

3.2. DATASET DESCRIPTION

For this research, CIC-Darknet2020 has been acquired from an open-source repository [51]. The study worked on the DarkNet File only. The datasets files reflect how network nodes act in space; nodes have sight of contacts with each other from time to time for certain periods intervals. Those Datasets reflect nodes' behavior inside their VPN, whether they are active (the node gets connected or disconnected due to its movement), or passive (the node is stationary, but it gets connected or disconnected due to other nodes movement), or a combination of both. We should not forget that connections and disconnections could be caused by third-party nodes (other than the sender or the receiver) by being in the middle of the sight of contact and thus blocking nodes from seeing each other. The ISCXVPN2016 used to be a representative dataset of real-world traffic on internet with features denoted in the .csv files for processing for VPN network traffic with related features. ANN learns more efficiently with CIC-Darknet2020 in comparison with ISCXVPN2016 when exposed to a broader range of features in which more possible behavioral scenarios in dataset. To achieve that goal, 900,000 space VPN data entries were produced using MATLAB. The aforementioned vast dataset was used to train the final solution ANN. On the other hand, for the tuning experiments, a smaller dataset of 700,000 space VPN entries was used. Using a smaller dataset is that there is no need for a big dataset if the goal is just experimenting with comparing performances rather than training to reach high accuracy. Furthermore, a smaller dataset means less time to perform experiments and hence more time available to do even more experiments. The dataset could be downloaded from the link: https://www.unb.ca/cic/datasets/darknet2020.html.

3.3. APACHE SPARK

Apache Spark is an open-source web crawler of Apache Software Foundation. It can run on a single and local machine, but it may also run over a Hadoop cluster. There are two different versions. The first is more suitable for batch processing, related to Hadoop data structures. The second one based on the prior version, it is more complicated and flexible. It allows working with many other storage model solutions. As Spark is built on Apache Hadoop and Apache Tomcat, it is easy to integrate Apache Kafka, an indexing Apache tool. It would be possible to create a custom search engine and gather information from the specific domains, either over the internet or intranet. Regarding the functionalities, the main idea is to crawl the VPN content. It allows downloading of the content of the VPN network. Also, the tool may search for links on specific pages/domains and index the links or contents for future access, as we mentioned before. Apache Spark is an open-source tool for processing high volume of data by Apache Software Foundation falls in the domain of big data. The VPN mobile communication technology that is about to be harnessed into use during 2021 has been an anticipated evolution of the currently used VPN technology. The extensive performance and multiple new services are enabling innovations in multiple different verticals. The VPN wireless mobile communication with Apache Spark is one of the verticals that benefits from this technological leap. The research evaluates if the VPN technology offers sufficient performance for the fully automated communication using the Apache Spark based technology for processing. Modern automated communication is requiring reliable and low-latency data transfer from the communication channel to exchange information with other entities. With the help of network slicing, the VPN network can be split into different data profiles serving customers with various needs simultaneously. The Big Data based Apache Spark will serve the end-users with ultra-low latency and highly reliable classification of features that is required by autonomous devices to communicate with smart city infrastructure. The Apache Spark offers high data throughput that can be used to transfer data-intensive sensor input to the cloud for further analysis.

The spark architecture, well documented in this work, is modular and divided into 4 steps as we show in Figure 3.4.

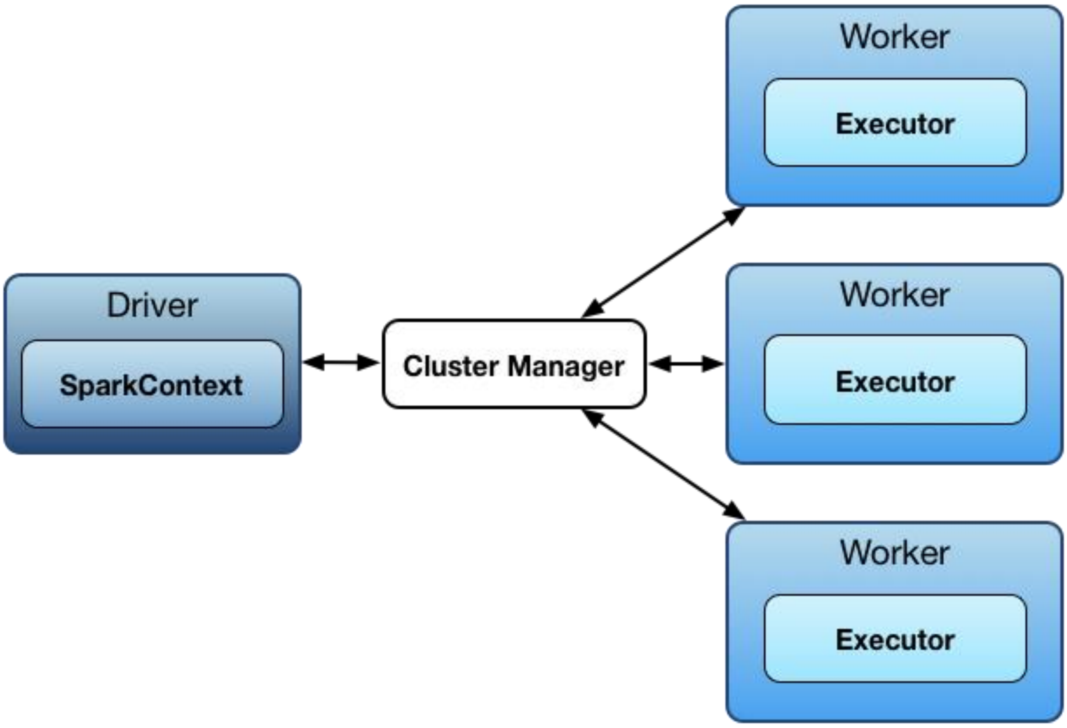

Figure 3.4. The architecture of Apache Spark for the implementation and experiments on the VPN network.

Driver: This first step is responsible for finding all the links and content in a given list of domains or dedicated networks.

Workers: This step might be optional if the goal is to build a search engine or just to "work" as a slave for the driver with the content from the prior step. It is responsible for indexing the data to provide reliable network classification.

Cluster Manager: It stores and manages the network data using a machine learning algorithm and the links in the gathered information. Also, it may manage different indices.

Executors: It is the step where network data gathered is executed with specific information from the worker nodes' cluster manager, exactly where the information is extracted, e.g., links and time-related features.

3.4. IMPLEMENTATION STEPS AND FRAMEWORK

The proposed method roadmap describes finding the best solution and then designing and building final solution as we show in Figure 3.5.

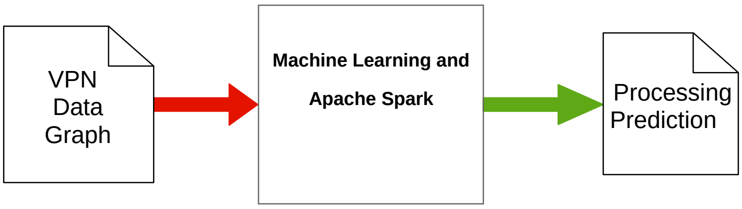

Figure 3.5. The basic processing of VPN data through the ANN and Apache Spark to classify network traffic flow.

3.4.1. First Step

The first step of the proposed solution roadmap is to build a machine learning model that can predict the right next hop when the related contact graph data entry is plugged into its application types of VPN and non-VPN as given in Table 3.1. This thesis work aims to use ANN techniques due to its great benefits discussed before in this chapter, as it helps avoid the shortcomings of the traditional space VPN network traffic flow. Figure 3.5 explains the general goal of this thesis (Step One of the Proposed Solution).

Table 3.1. Applications and traffic types of the CIC-DarkNet2020 dataset.

TRAFFIC TYPE	CONTENT
Email	Email, Gmail (SMTP, POP3, IMAP)
VPN-Email	
Chat	ICQ, AIM, Skype, Facebook, Hangouts
VPN-Chat	
Streaming	Vimeo, YouTube, Netflix, Spotify
VPN-Streaming	
File transfer	Skype, FTPS, SFTP
VPN-File transfer	
VoIP	Facebook, Skype, Hangouts, Voip-buster
VPN-VoIP	
P2P	uTorrent, Bitorrent
VPN-P2P	

3.4.2. Second Step

The second step in designing the solution is creating a Space VPN Contact Graph dataset representing three space VPN nodes. The dataset created consists of six-inputs which are mapped into two outputs. The six inputs are divided into three pairs; each pair represents the start and end time of the availability of space VPN link. Depending on the three pairs of inputs, the two mapped outputs serve as an indication of which next-hop to take. The dataset is generated by creating an algorithm code using (MATLAB). The dataset consists of 600,000 data entries, and each has six inputs mapped to two outputs. Another data set consisting of 10,000 data entries is also created to be used in the solution tuning results in chapter 4. A third data set consisting of 60,000 data entries used to test the final proposed solution to measure its prediction accuracy and performance. The proposed system uses 80% of the dataset for training while 20% is used for the testing and validating with 10-cross fold validation and 50 epochs of training as we show in Figure 3.6 and Figure 3.7.

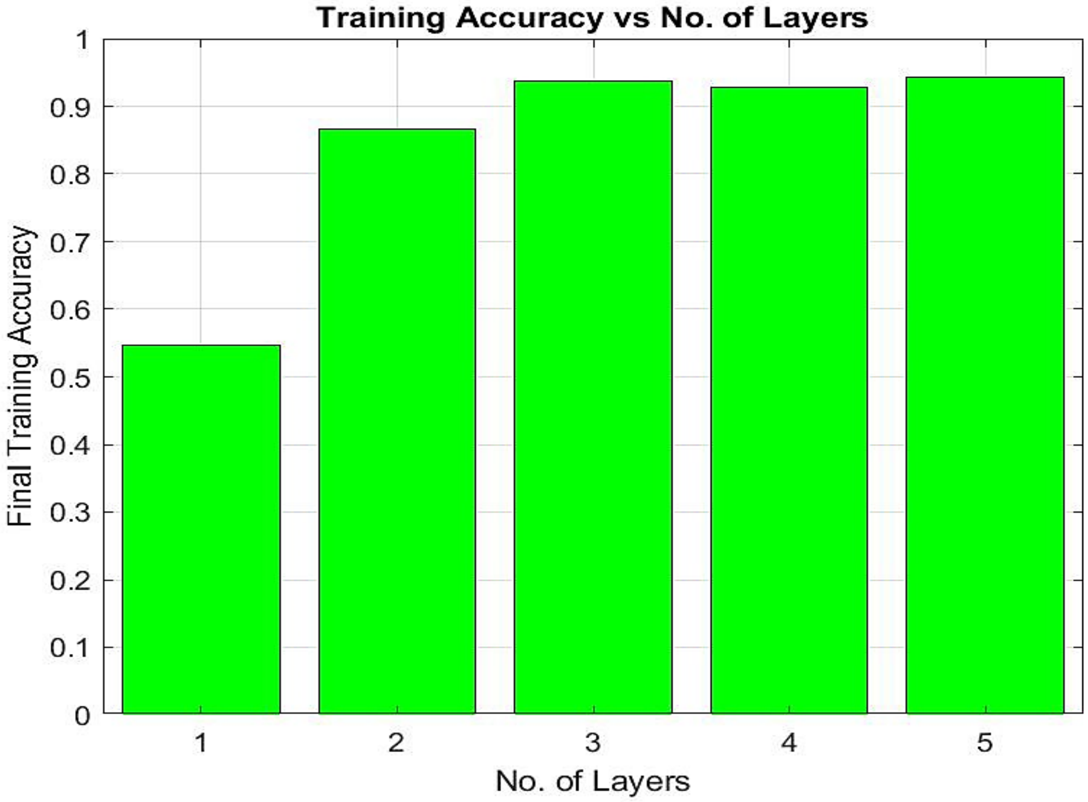

Figure 3.6. The training accuracy loss of ANN network with Spark Engine vs the number of layers.

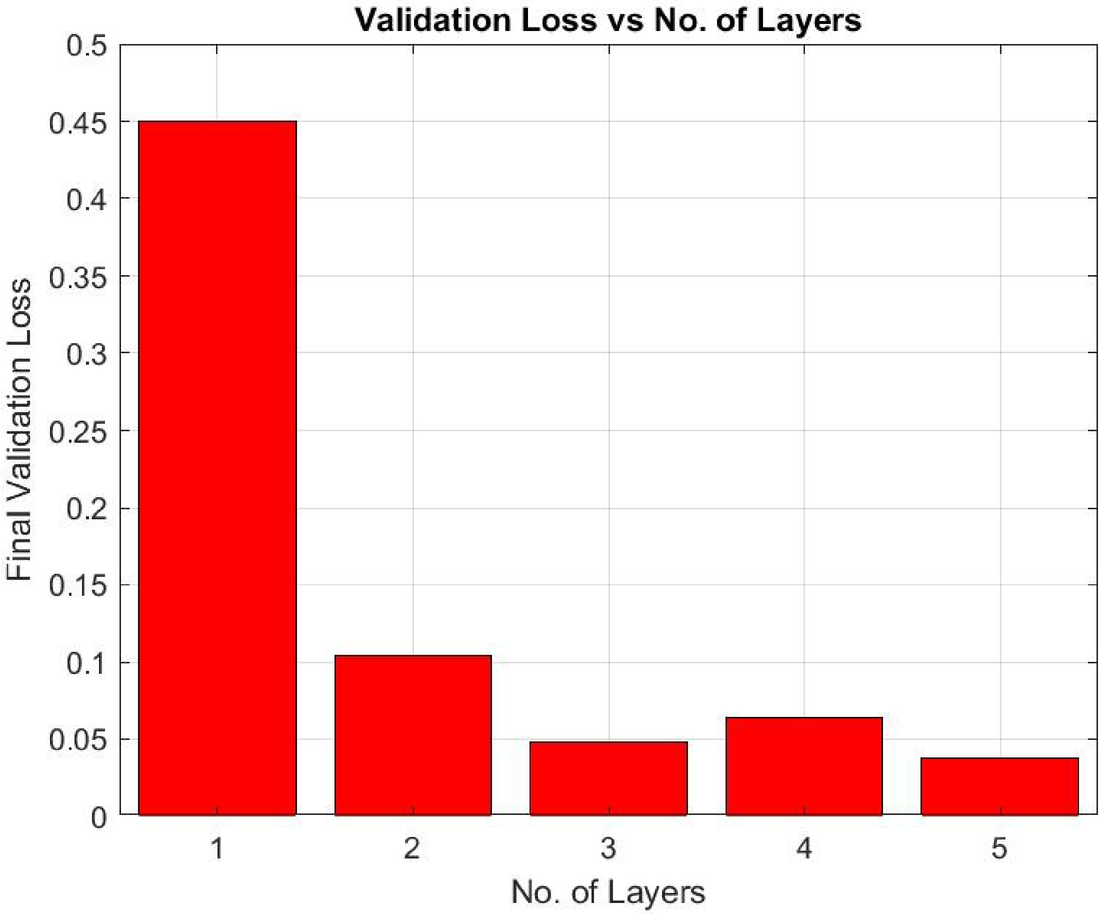

Figure 3.7. The validation loss of ANN network with Spark Engine vs the number of layers.

3.4.3. Third Step

Earlier in this chapter, we have highlighted the significant advantages of ANN, including their adaptability and customization ability to target problems and design and tailor a solution for them. This thesis chooses ANN; to take advantage of the beneficial characteristics they are offering. The main factor that made this work choosing ANNs is the “Fitting” ability to map specific input data-entries to match them with certain output data-entries, which could be achieved using Supervised Training.

After creating an ANN, it passes through a set of experiments to tune and customize it. The customization target builds a neural network that uses the least possible resources yet gives an acceptable performance value. The set of experiments include:

1. Neural network depth experiment to be able to choose good depth for the proposed neural network.
2. Gradual neural network widening experiment, to be able to test the effect of having different widths layers.
3. Activation functions experiments to examine the activation functions and choose a suitable activation function that can give the best performance when working on our space contact graph dataset.
4. Exponential Increase of neural network width experiment. This experiment is to test the impact of widening the whole neural network drastically and check its effect on the training time and performance.
5. The cone effect experiment test how much the network performance is affected by having all layers of an equal width versus having them gradually decreasing (like a cone shape).
6. Final space VPN neural network routing solution. This is the last experiment in which all the knowledge gathered from all the previous experiments utilized to build a robust neural network using the least possible resources.

3.4.4. Fourth Step

In this step, the ANN trained model was tested. The test is implemented using the new contact graph dataset that the neural network has never seen before. Using a new dataset with a fair evaluation for the proposed space VPN ANN routing solution with Apache Spark. Figure 3.8 and Figure 3.9 demonstrates the last step of the proposed solution roadmap, which is testing the final ANN model. Figure 3.10 demonstrates this thesis's proposed solution, which is a space delay-tolerant network ANN routing model.

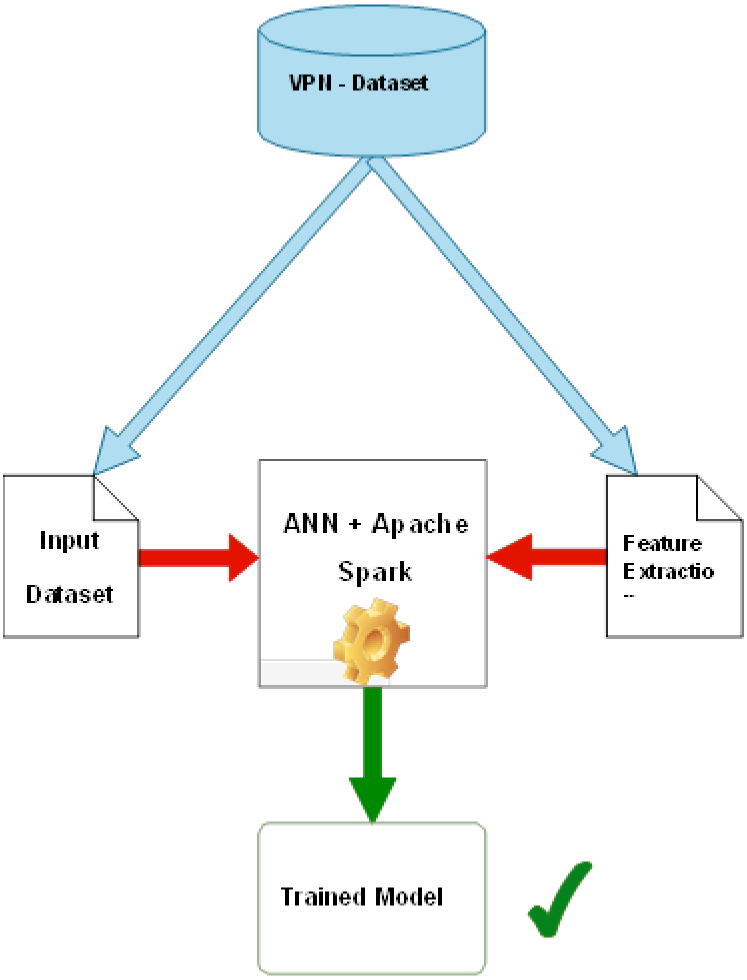

Figure 3.8. The step for proposed solution by using ANN to create trained model.

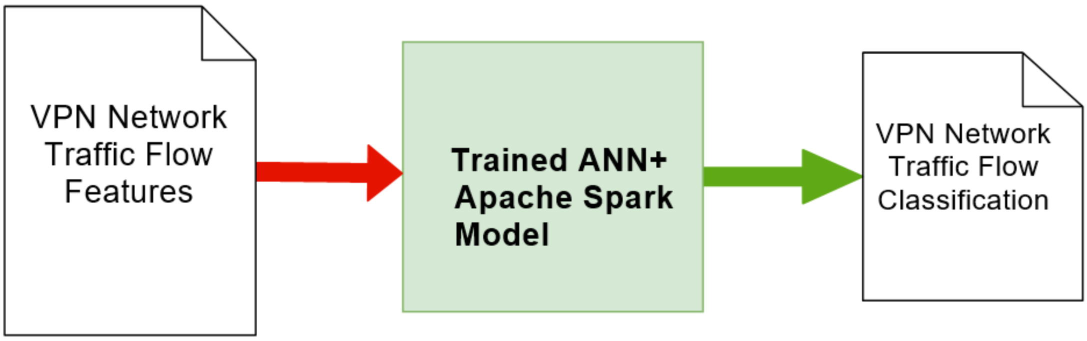

Figure 3.9. Last step in creating the proposed VPN Network Traffic Flow Classification.

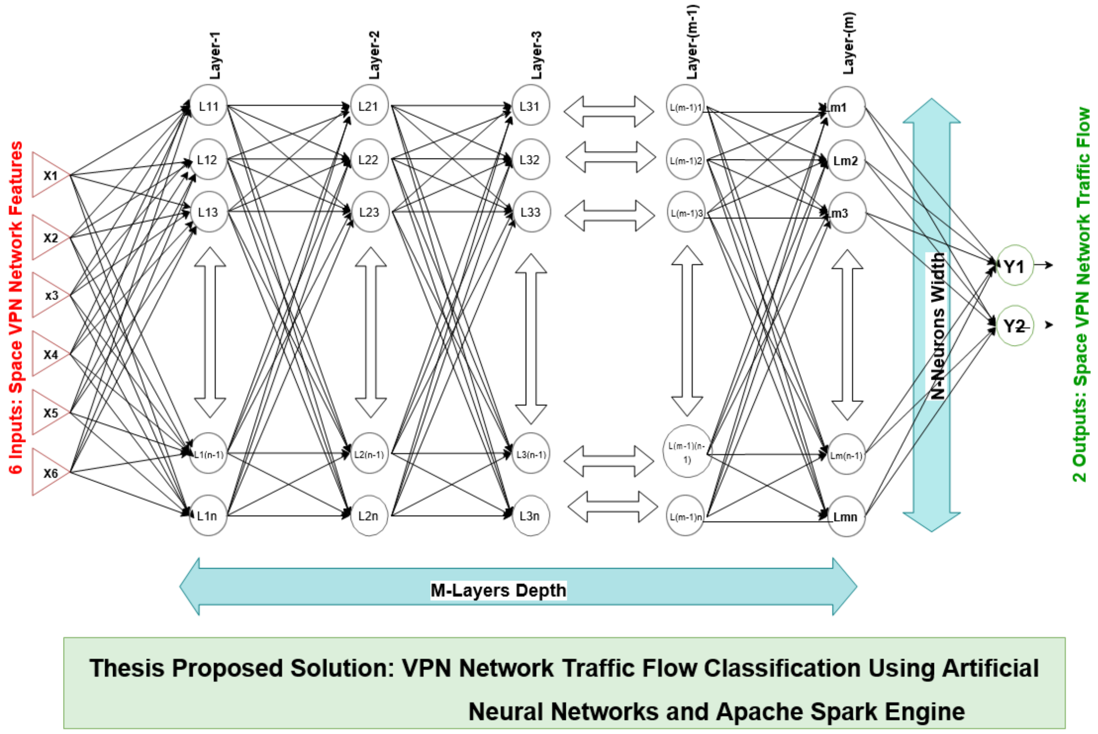

Figure 3.10. Solution: VPN Network Traffic Flow withANN using Apache Spark Engine.

3.4.5. Apache Spark Depth Experiment

In this experiment, a feed-forward network was used to show the impact of changing the network layers' number on its performance. Before performing this experiment, network width is fixed as well as its training parameters. The sigmoid is used in the last section of layers and it’s a logistic function with curved and derived formula used to calculated the curving of ANN for multi-class classification. The first trial begins by training the network with one layer of depth only, then two, and so on up to five layers. Accuracy and loss of training calculated. The network is then exposed to a new set of data to perform a validation test, the validation accuracy and loss recorded. Finally, the results are discussed per the number of layers as well as training and validation results per epoch.

3.4.6. Gradual Spark Widening Experiment

In this experiment, a feed-forward network used to show the impact of gradually changing layers' width on the neural network performance from both training and validation perspective. Before performing this experiment, network depth fixed as well as its training parameters. The first trial begins by training the network with n number of neurons width only, then gradually widen more layers to be of a (10*n) neurons width. Accuracy and loss of training calculated. The network is then exposed to a new set of data to perform a validation test, the validation accuracy and loss calculated. Subsequently, the results discussed per number of wide layers. Finally, training each of the produced networks was measured to examine if there is any time difference among the different trials.

3.4.7. Activation Function in ANN with Apache Spark

After experimenting with the Apache Spark experimental topology's depth and width, it is essential to spend some time and effort exploring and testing neural networks' activation functions. This group of experiments applies training tests on the primary activation functions to demonstrate their strength, enabling us to choose the most suitable activation function for our VPN network traffic flow classification. The following experiments test Relu, Softmax, Tanh, and Sigmoid activation functions. To magnify the effect of activation functions, a neural network of five layers was used. In every experiment, one activation function is chosen, and it is the only function that is applied throughout the whole network. This scheme put the entire focus on one activation function at a time, and thus the result counting totally on it. Using this methodology ends up giving as much difference as possible among the training results.

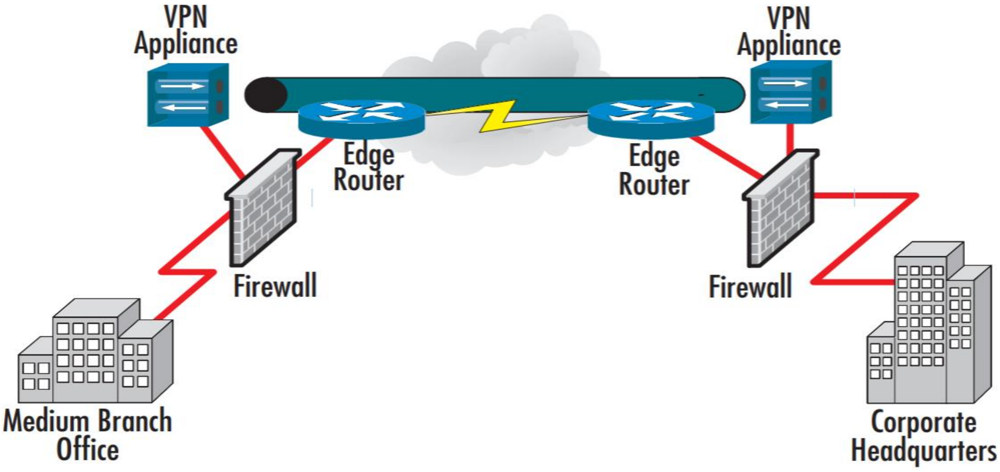

Figure 3.11. Topology of VPN that is being considered in this research work.

In this group of experiments, in Figure 3.11 same topology of the network was used for all activation functions; the only difference here is to change the activation functions into ReLU, SoftMax, Tanh, and then sigmoid. The number of epochs used to train all the networks has been fixed to a constant value. Furthermore, for a fair comparison among the activation functions, training parameters of the learning rate and the batch number kept constant throughout the experiments. Several training processes were performed to evaluate the results of the training accuracy, training loss, evaluation accuracy, and evaluation loss. For any specific activation function, the training and evaluation results stay almost the same, only with a negligible change in values. Thus, the next step is to stop any randomness in the training process by fixing the seed value and stopping the shuffling in the training epochs to make an accurate comparison. Eliminating the randomness in all of the experiments does not change the results of both training and evaluation accuracy and loss on the contrary, it makes all of the training trials start from the same training point. Thus, the process becomes more effective. Using this approach helps a lot to do more experiments in a shorter time.

3.4.8. VPN Cone Network Traffic Flow Experiment

This experiment presents the difference between having an ANN with similar width layers versus cone-shaped architecture. Cone shape is nothing but the input layer being the widest and the output layer the thinnest, while the medium layers gradually decrease in width as we move toward the output layer. To investigate this effect before starting to generalize, an experiment is implemented to compare a fully wide neural network with a cone-shaped network. In both training trials, the number of epochs and training parameters are fixed. Both training and validation accuracy and loss be measured and plotted for analysis purposes.

PART 4

RESULTS

To explore the behavior of neural networks in different situations, many experiments were implemented in this thesis. The experiments involve investigating the effect of changing the parameters that are required to build ANNs. Not only the parameters changed, but additional components also are varied to check the difference in the network behavior. Activation functions are elements that do not modify an ANN's topology, but they may impact its performance. On the other hand, changes like network depth and width change the topology itself, and we investigate whether they change the performance. Working with ANN could not only be described as being a mathematical or experimental effort; at this point of the thesis, we can also say fine-tuning the neural network and customizing it is becoming a type of art. The following experiments try to find a potent enough solution by using the least possible resources, giving the best accuracy, least loss, and least training time.

Table 4.1. The implementation classifies 91% of the packets for given application analysis in non-VPN traffic.

Application	**Number of flows**	**Number of packets**	**Unique flows**	**Unambiguous packets**
Audio-Stream	865	6656	0.65	0.69
Browsing	5123	64568	0.85	0.88
Chat	6421	84121	0.76	0.84
Email	22145	984655	0.75	0.79
P2P	214	7656	0.71	0.75
Transfer	42422	212100	0.89	0.92
Video-Stream	484	6654	0.62	0.66
VOIP	655	979440	0.79	0.85
Average	**65660**	**985550**	**0.86**	**0.91**

Table 4.2. The implementation can uniquely classify93 %of the packets for given applications. This analysis incorporates both VPN and non-VPN traffic.

Application	Number of flows	Number of packets	Unique flows	Unambiguous packets
Audio-Stream	1545	9845	0.75	0.78
Browsing	5644	91098	0.65	0.71
Chat	54545	5601864	0.87	0.88
Email	2545	8638433	0.79	0.81
P2P	566	14931	0.62	0.78
Transfer	98780	7889219	0.72	0.76
Video-Stream	980	12096	0.86	0.88
VOIP	569	1250836	0.79	0.83
Average	**96550**	**45765510**	**0.91**	**0.93**

This work focuses on the same packets that were used in the Deep Packet in Apache Spark. That means that only TCP and UDP packets with an application related payload were considered. Therefore, DNS packets and SYN, ACK, FIN packets were discarded. We have not published code for the data preprocessing stage but based on the number of packets they mention for every application; it is reasonable to assume that they only used non-VPN traffic for the application classification task.

To examine the flows of the dataset, every packet header is inspected. But instead of looking at the 5-tuple consisting of source IP, destination IP, source port, destination port, and protocol, only the source port, destination port, and protocol are used to define a flow. This accounts for the IP masking step in the Deep Packet as mentioned in paper [54]. As this eliminates the uniqueness of a flow in the dataset by removing the application-specific IP address, flows should not be associable with applications. Table 4.1 shows the results of implementation classifies 91% of the packets for given application analysis in non-VPN traffic. At least two applications use all other flows. Like VOIP or Email, only a few applications could be identified with high accuracy based on a given flow. This suggests that the Deep Packet approach should not classify applications with high accuracy only by looking at the header information. However, taking into account the number of packets per flow leads to a different conclusion. The unique flows include 91% of all packets in the dataset. It means that one can get a very high classification accuracy by simply mapping flows to applications. If a packet belongs to a non-unique flow, guessing the application still has a relatively high chance of being correct, increasing accuracy beyond 93%. The VPN time-related features with x-axis represents the VPN features while y-axis represents the detection and classification precision As we show in Figure 4.1.

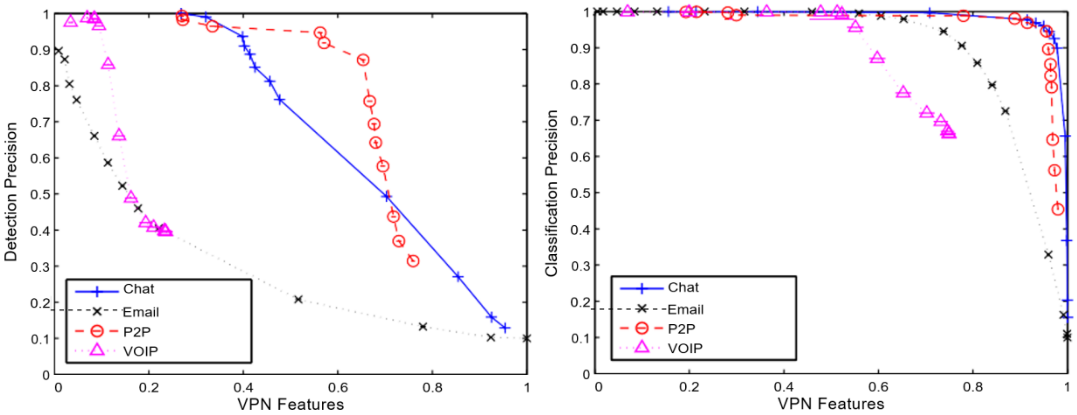

Figure 4.1. The VPN time-related features with x-axis represents the VPN features while y-axis represents the detection and classification precision.

Although the achieved results are so promising, it helps compare them to perfect world scenario results. In an ideal world, the results would be (100%) validation accuracy, which is a complete 1.0 and a (0%) validation loss, that is (0.0). The difference between the ideal accuracy and the proposed solution's accuracy is just (0.0096). Furthermore, when the proposed neural network has an accuracy of (96.7654%) that does not mean that it gives (0.39%) wrong outputs, instead this accuracy is just showing the difference of predictions before applying the thresholds. For example, let us consider a case when the ANN gives an output of (0.9,0) instead of an ideal (1.0,0), even though the accuracy indicates (0.9) after applying the threshold, the regulated output would be (1.0,0), which is the same output of a 100% accurate system.

Table 4.1 shows only non-VPN traffic. If the whole dataset is taken into account, the fraction of unique flows is higher, but the number of packets associated with a specific app decreases to 93% (see Table 4.2). Again, guessing the application for ambiguous packets increases the classification accuracy. However, the whole dataset is only used for traffic type identification in this research work. The classification precision of VPN time-related features using ANN and Apache Spark Engine as we show in Figure 4.2.

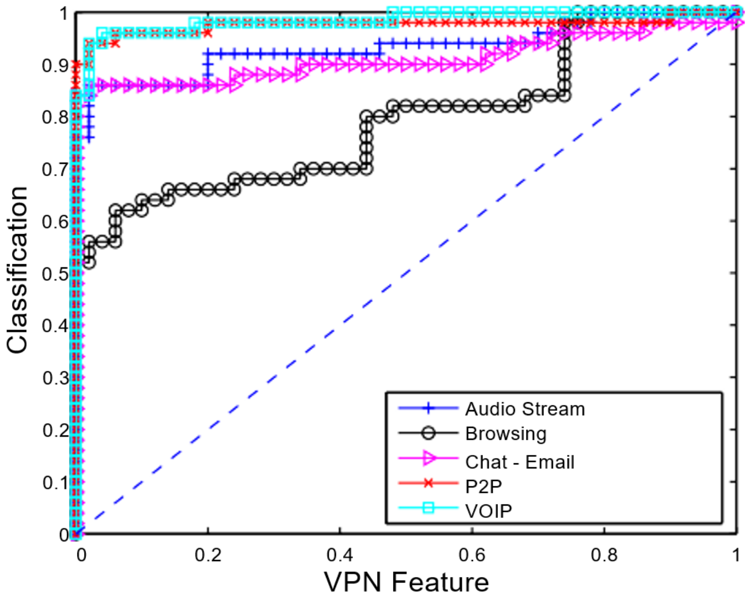

Figure 4.2. The classification precision of VPN time-related features using ANN and Apache Spark Engine.

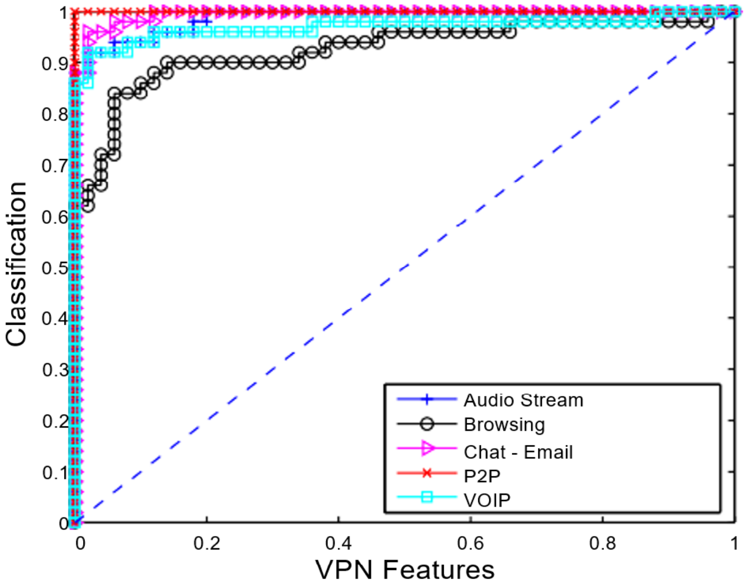

Figure 4.3. The classification of major time-related VPN features considering all processed data at 50 epochs.

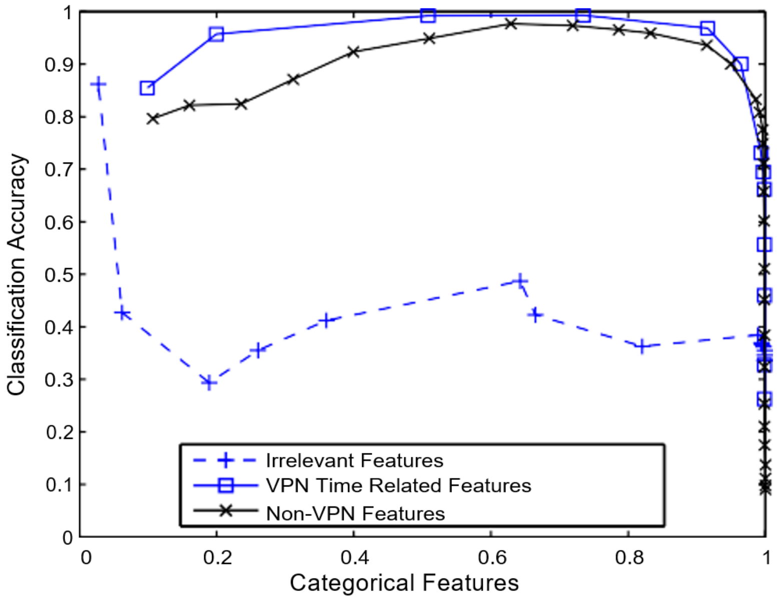

Figure 4.4. The categorical features classification of VPN and Non-VPN features with respect to overall classification precision.

From a routing perspective, if the proposed solution gave a route a value of 0.99 and 0.01 to another, by using a threshold of 0.9 as a deciding edge, then after applying this threshold, the ANN + Spark will still give an answer of 1 to the first route and 0 to the second. In conclusion, the proposed ANN + Spark can provide highly accurate results for the ideal world scenarios.

PART 5

DISCUSSION

5.1. DISCUSSION

As a result, the usage of statistical-based ANN and behavioral-based ANN approaches works well with Apache Spark Engine. Statistical-based approaches leverage web traffic diversity (e.g., short packet bursts in Voice over IP applications in contrast to long, steady flows of file downloads). Examined features can be packet length, several packets in a given time, or used protocol. On the other hand, behavioral-based approaches look an endpoint in a network and analyze how many hosts are contacted, which protocol, and how many different ports. These approaches originally used hand-crafted features and “classic” machine-learning techniques, like ANN. The advent of real-life applicability of neural networks since 2019 promoted research on using ANN and Apache Spark engine to solve the VPN traffic classification problem with an end-to-end approach. Apache Spark and machine learning models usually do not use hand-crafted features but directly learn the input data's underlying statistics. A schematic comparison of the traffic classification approaches can be seen in Fig 4.5. Both approaches use one-dimensional ANN on single packets, including header and payload. Both papers use the CIC-Darknet2020 for training and testing their models. The CIC-Darknet2020 consists of around 72MB of recorded traffic. It can be used for traffic identification or application classification. For this reason, the used applications have been grouped into traffic groups. We showed the applications and traffic types. Comparing the results of Tables 5.1 (CNN results) and 2 shows a correlation between the proposed model's predictive performance and the number of unambiguous packets associated with each application. The applications with low F1-scores are also applications that have a very low percentage of unambiguous packets. Namely, AIM chat and ICQ have the lowest F1-scores and are also the applications with the lowest percentage of un-ambiguous packets, followed by Email. It is very unlikely that our approach learned how to find patterns in the encrypted data. It is more likely that the model learned to memorize the mapping from flows to applications.

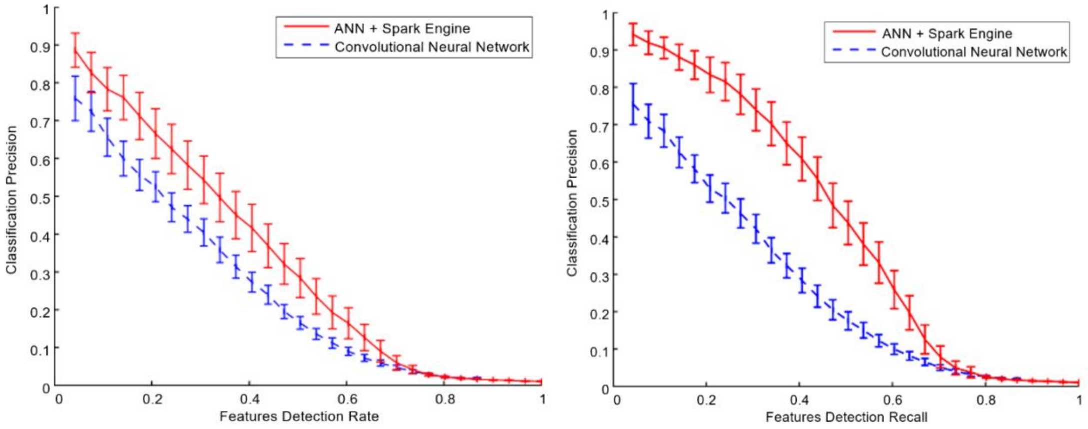

Figure 5.1. The comparison of classification)Feature detection rate and recall(between ANN+ Spark Engine Technique Vs. the CNN technique.

In Figure 5.1 and 5.2 The comparison of classification (Feature detection rate and recall) between ANN + Spark Engine Technique Vs. the CNN technique. Using the whole dataset (VPN and non-VPN) reduces the accuracy for application classification but as applications are grouped into traffic types and grouped applications probably share flows, the traffic type identification performed with the ANN + Apache Spark approach most likely also just memorizes the flow-to-traffic-group mapping.

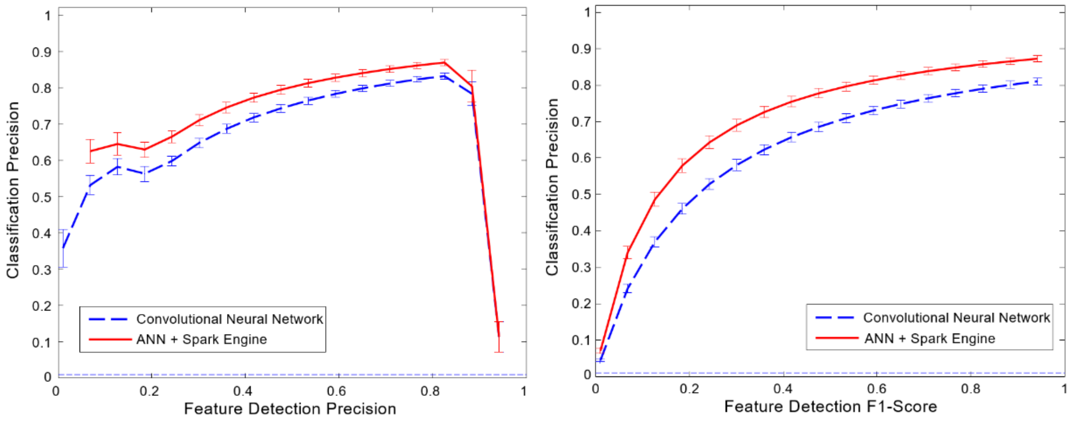

Figure 5.2. The comparison of classification (Feature detection precision and F1-score) between ANN + Spark Engine vs. CNN.

Table 5.1. The ANN+ Spark Engine technique outperforms the CNN on CIC-Darknet2020 and ISCXVPN2016 respectively.

Application	**ANN + Spark Engine**			**CNN + SAE [49-50]**		
	Recall	**Precision**	**F1-Score**	**Recall**	**Precision**	**F1-Score**
Audio-Stream	0.87	0.89	0.86	NA	NA	NA
Browsing	0.89	0.89	0.86	NA	NA	NA
Chat	0.85	0.89	0.87	0.82	0.85	0.83
Email	0.91	0.89	0.87	0.78	0.82	0.80
P2P	0.92	0.95	0.91	0.91	0.87	0.90
Transfer	0.94	0.96	0.95	0.94	0.90	0.91
Video-Stream	0.88	0.91	0.89	0.89	0.81	0.79
VOIP	0.95	0.97	0.94	0.95	0.90	0.92
Average	**0.97**	**0.98**	**0.97**	**0.96**	**0.94**	**0.95**

Overall, the situation in current research regarding encrypted traffic classification is methodically unsatisfying. Authors do not publish code, use non-public datasets, or do not explain data preprocessing steps in detail. This makes it hard to evaluate their results. Nonetheless, statistical and time series data usage seems to be the state-of-the-art solution for encrypted traffic classification.

5.2. ANALYSIS OF STUDY

These are the analysis of the study obtained by going throw the experimental steps to build this solution, and they are:

1. Adding more layers into ANN empower them to reach higher accuracy values for the same number of epochs.
2. Adding more layers to neural networks may cause them to over-train, so when you get good accuracy, no more layers are needed.

3. It is much harder for an ANN with one layer to solve problems. Some problems can never be solved with one layer, as in the case of a one-layer perceptron trying to solve an XOR logic function.
4. Adding more layers to ANN causes longer training times for the same number of epochs.
5. For every problem to be solved using ANN, there is a point when adding more layers becomes inefficient, either by spending longer training times, wasting resources, or both.

The deeper the VPN nodes go into space, the more efficient they work if using the proposed solution. As VPN network traffic classification using ANN and Apache Spark provide the nodes with future knowledge about when to transmit/receive the bundles, that could be used to turn off the nodes for more extended periods and turn them on just before data handshaking is about to take place. Thus, this solution minimizes VPN nodes' power consumption. The proposed solution can increase network security, as nodes out of the grid or wholly powered off for the time they are not needed. That prevents nodes from being open for security attacks. Furthermore, the nodes know which their next encounter is and does not contact any other node. It is easy to add or remove new nodes to the ANN network; it only takes is to feed the ANN network with the latest trained model. The following study [53] has been carried out within the scope of the thesis:

1. Aswad, S. A., & Sonuç, E. (2020, October). Classification of VPN Network Traffic Flow Using Time Related Features on Apache Spark. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-8). IEEE.

PART 6

CONCLUSION

6.1. CONCLUSION

This research work presented the detection and classification of network traffic flow and time related features in virtual private network with the help of ANN and Apache Spark. The proposed solution, the ANN and Apache Spark engine trained on feature of VPN with multiple nodes. Any node receives a bundle when no processing is needed, as all the routing processing has been completed already. The only task that the node is required to do is to forward the bundle to the right next-hop when the time comes, and it appears in the sight of contact. Applying the proposed ANN prevents unnecessary processing and flooding found in common VPN network traffic classification. The proposed system uses 80% of the dataset for training while 20% is used for the testing and validating with 10-cross fold validation and 50 epochs of training. This is the first study that introduces and utilizes ANN and Apache Spark engine to implement VPN network traffic flow classification to the best of our knowledge. The categorical features classification of VPN and Non-VPN features concerning overall classification precision where VPN classification precision stands at 96.76%; however, the Non-VPN stands at 92.56%. The ANN + Spark Engine technique outperforms the convolutional neural network compared to the stacked auto-encoder network on CIC-Darknet2020 and ISCXVPN2016, respectively.

6.2. FUTURE RECOMMENDATION

There are several areas of possible future work that could be built on top of the solution proposed by this thesis, a few of them are listed below:

1. Virtual Private Network (VPN) becomes possible if the presented solution is used by feeding the chosen group of VPN nodes, a trained neural network model that includes only those nodes with no other. That is simply how a VPN could be created.
2. With the proposed solution nodes could have different levels of priority. Express routes could be built by synchronizing priority nodes according to specific patterns or priority node power-cycle.
3. Emergency hot-lines are possible by using the proposed implementation, a trained model can contain zero-possibility to some standby nodes.
4. An emergency message from space crew back to earth, they will just have to send” Emergency Model” into neighboring nodes, and that will turn-on and deploy the emergency route for that purpose only.

REFERENCES

1. Dhote, Y., S. Agrawal, and A.J. Deen. “A survey on feature selection techniques for internet traffic classification”, ***International Conference on Computational Intelligence and Communication Networks (CICN)***: IEEE, Jabalpur 1375-1380, (2015).

2. Rathore, M.M., et al. “Exploiting encrypted and tunneled multimedia calls in high-speed big data environment”, ***International Conference on Computational Intelligence and Communication Networks (CICN)***: IEEE, Jabalpur, India, 4959-4984, (2018).

3. Velan, P., et al. “A survey of methods for encrypted traffic classification and analysis”, ***International Journal of Network Management***: Institute of Computer Science, Masaryk University, Brno, Czech Republic, Brno, Czech,355-374, (2015).

4. Hirvonen, M. and M. Sailio. “Two-phased method for identifying SSH encrypted application flows”, ***2011 7th International Wireless Communications and Mobile Computing Conference***: IEEE, Istanboul, 1033-1038, (2011).

5. Wang, Z.J.B.U. “The applications of deep learning on traffic identification”, ***Z Wang - BlackHat USA, 2015 - covert.io***: covert.io, USA,1-10, (2015).

6. Tan, X., et al. “Recognizing the content types of network traffic based on a hybrid DNN-HMM model”, ***Journal of Network and Computer Applications***: Science Direct, China,51-62, (2019).

7. Usama, M., et al. “Unsupervised machine learning for networking: Techniques, applications and research challenges”, ***Advanced Search Browse Journals & Magazines >IEEE Access***: IEEE, USA,65579-65615, (2019).

7. Limthong, K., et al. “Unsupervised learning model for real-time anomaly detection in computer networks”, ***IEICE***: IEICE TRANSACTIONS on Information and Systems, Japan,2084-2094, (2014).

8. Suthaharan, S.J.A.S.P.E.R. “Big data classification: Problems and challenges in network intrusion prediction with machine learning”, ***ACM SIGMETRICS Performance Evaluation Review***: Special internet group,Uk,70-73, (2014).

9. Krawczyk, B., et al. “Ensemble learning for data stream analysis: A survey”, ***Journals & Books***: Elsevier, USA, 132-156, (2017).

10. Thanh Noi, P. and M.J.S. Kappas. “Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery”, ***Sensors***: MDPI, Germany, 18, (2018).

11. Oehmcke, S., O. Zielinski, and O. Kramer. “kNN ensembles with penalized DTW for multivariate time series imputation”, ***International Joint Conference on Neural Networks (IJCNN)***: IEEE, Vancouver/Canada, 2774-2781, (2016).

12. Liu, Z., et al. “Mobile app traffic flow feature extraction and selection for improving classification robustness”, ***Journal of Network and Computer Applications***: Elsevier, USA, 190-208, (2019).

13. Draper-Gil, G., et al. “Characterization of encrypted and vpn traffic using time-related”, ***2nd International Conference on Information Systems Security and Privacy***: Researchgate, Canada,407-414, (2016).

14. Yin, C., H. Wang, and J. Wang. “Network data stream classification by deep packet inspection and machine learning”, ***Advanced Multimedia and Ubiquitous Engineering***: Springer, China, 245-251, (2018).

15. Sherry, J., et al. “Blindbox: Deep packet inspection over encrypted traffic”, ***Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication***: ACM, USA, 213-226, (2015).

16. Meiners, C., et al. “Flowsifter: A counting automata approach to layer 7 field extraction for deep flow inspection”, ***2012 Proceedings IEEE INFOCOM***: IEEE, USA,1746-1754, (2012).

17. Zeng, X., et al. “Flow context and host behavior based shadowsocks’s traffic identification”,***journals & Magazines >IEEE Access***: IEEE, USA, 41017-41032, (2019).

18. Hejun, Z., et al. “Online and automatic identification of encryption network behaviors in big data environment”, ***Concurrency and computation practice and experience***: Wiley online library, China, e4849, (2019).

19. Zygmunt, M., M. Konieczny, and S. Zielinski. “Accuracy of statistical machine learning methods in identifying client behavior patterns at network edge”, ***42nd International Conference on Telecommunications and Signal Processing (TSP)*** IEEE, Budapest, Hungary,575-579, (2019).

20. An, H.-M., et al. “Traffic Identification Based on Applications using Statistical Signature Free from Abnormal TCP Behavior”, ***Journal of Information Science and Engineering 31***: nmlab.korea.ac.kr, Korea, 1669-1692, (2015).

21. Wang, H., et al. “Practical network-wide packet behavior identification by AP classifier”, ***IEEE/ACM Transactions on Networking***: IEEE, USA, 2886-2899, (2017).

22. Kim, J., A.J.J.o.C.S. Sim, and Technology. “A new approach to multivariate network traffic analysis”, ***journal & magazine***: Springer, USA, 388-402, (2019).

23. Cha, S. and H. Kim. “Detecting encrypted traffic: a machine learning approach”, ***International Workshop on Information Security Applications***: Springer,Cham, Switzerland, 54-65, (2016).

24. Sun, G., et al. “Internet traffic classification based on incremental support vector machines”, ***Mobile Networks and Applications***: Springer, China,789-796, (2018).

25. Raje, A. and S. Sinha. “Anonymous Traffic Networks”, ***Journal & Magazine***: Springer, China, 263-286, (2021).

26. Gómez, S.E., et al. “Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal”, ***Journal of Network and Computer Applications***: Elsevier, 68-80, (2017).

27. Aceto, G., et al. “Multi-classification approaches for classifying mobile app traffic”, ***Journal of Network and Computer Applications***: Elsevier131-145, (2018).

28. Xu, G., Z. Wang, and T.J.A.S. Xia. “Mapping Areal Precipitation with Fusion Data by ANN Machine Learning in Sparse Gauged Region”, ***Applied Sciences***: MDPI, China, 2294, (2019).

29. Ma, C., X. Du, and L.J.E. Cao. “Improved KNN Algorithm for Fine-Grained Classification of Encrypted Network Flow”, ***Electronics***: MDPI, China,324, (2020).

30. Song, M., J. Ran, and S. Li. “Encrypted Traffic Classification Based on Text Convolution Neural Networks”, ***7th International Conference on Computer Science and Network Technology (ICCSNT)*** IEEE, Dalian, China, 432-436, (2019).

31. Rezaei, S. and X. Liu. “Multitask learning for network traffic classification”, ***ICCCN*:**IEEE,USA,1-9, (2020).

32. Zekkori, H., S.J.J.o.N. Agoujil, and C. Applications. “Hybrid delay tolerant network routing protocol for heterogeneous networks”, ***Journal of Network and Computer Applications*** Elsevier, Morocco, 102456, (2019).

33. Sun, G., et al. “Active learning method for chinese spam filtering”, ***International Journal of Performability Engineering*** pdfs.semanticscholar.org, China, 511-518, (2017).

34. Singh, G.J.I.J.o.C.A. “A study of encryption algorithms (RSA, DES, 3DES and AES) for information security”, ***International Journal of Computer Applications (0975 – 8887)***: CiteseerX, India, (2013).

35. Hejun, Z. and Z.J.C.C. Liehuang. “Encrypted network behaviors identification based on dynamic time warping and k-nearest neighbor”, ***Cluster Comput 22***: Springer, China, 2571-2580, (2019).

36. Hejun, Z., et al. “Online and automatic identification and mining of encryption network behavior in big data environment”, ***Journals Artificial Intelligent Techniques and its Applications***: Iospress,China, 1111-1119, (2018).

37. Ahmad, S., et al. “A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets”, ***International Conference on Networking Systems and Security (NSysS)***: IEEE,dhaka/Bangladesh, 1-5, (2015).

38. McGaughey, D., et al. “A systematic approach of feature selection for encrypted network traffic classification”, ***Annual IEEE International Systems Conference (SysCon)***: IEEE,Vancouver/canada, 1-8, (2018).

39. Sun, G., et al. “Network traffic classification based on transfer learning”, ***Computers & Electrical Engineering***: Elsevier,China, 920-927, (2018).

40. Saber, A., F. Belkacem, and A. Moncef. “Encrypted Network Traffic Identification: LDA-KNN Approach”, ***Tendances dans les Applications Math´ematiques***: Researchgate.net,Morocco, 23-27, (2019).

41. Manju, N., et al. “Ensemble Feature Selection and Classification of Internet Traffic using XGBoost Classifier”, ***I. J. Computer Network and Information Security***: MECS, India,37, (2019).

42. Jamil, H.A. “Feature Selection and Machine Learning Classification for Live P2P Traffic”, ***Proceedings of the International Conference on Industrial Engineering and Operations Management***: ieomsociety.org,Thailand, (2019).

43. Bugata, P. and P.J.K.-B.S. Drotár. “Weighted nearest neighbors feature selection”, ***Knowledge-Based Systems***: Elsevier, Slovakia, 749-761, (2019).

44. Sun, B., et al. “Short-term traffic forecasting using self-adjusting k-nearest neighbours”, ***IET Intelligent Transport Systems***: IET, Sweden, 41-48, (2017).

45. Saleh, A.I., F.M. Talaat, and L.M.J.A.I.R. Labib. “A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers”, ***Artificial Intelligence Review***: Springer, Egypt, 403-443, (2019).

46. Zeng, Y., et al. “$ Deep-full-range $: A deep learning based network encrypted traffic classification and intrusion detection framework”, ***IEEE Access***: IEEE,China, 45182-45190, (2019).

47. Wang, W., et al. “End-to-end encrypted traffic classification with one-dimensional convolution neural networks”, ***IEEE International Conference on Intelligence and Security Informatics (ISI)*** IEEE,China, 43-48, (2017).

48. Iliyasu, A.S. and H.J.I.A. Deng. “Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks”, ***IEEE Access***: IEEE, 118-126, (2019).

49. Lotfollahi, M., et al. “Deep packet: A novel approach for encrypted traffic classification using deep learning”, ***Soft Computing***: Springer,Tehran, Iran, 1999-2012, (2020).

50. Vincent, P., et al. “Extracting and composing robust features with denoising autoencoders”, ***Proceedings of the 25th international conference on Machine learning***ACM,Canada, 1096-1103, (2008).

51. Aswad, S.A. and E. Sonuç. “Classification of VPN Network Traffic Flow Using Time Related Features on Apache Spark”, ***2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*** IEEE,Turkey, 1-8, (2020).

52. Antonello, R., et al. “Deep packet inspection tools and techniques in commodity platforms: Challenges and trends”, ***ournal of Network and Computer Applications***: Elsevier1863-1878 , (2012).

RESUME

Salma Abdullah Aswad was born in Duhok/ Iraq in 1987 and she graduated from Duhok elementary school. She completed her high school education in Duhok High School. And she obtained bachelor degree from University of Duhok/College of Science/Computer Department in 2009. Then in 2019, she started her master education in Karabuk University/Faculty of Applied Science/Computer Engineering Department.

CONTACT INFORMATION

Address **:** Duhok/Iraq
E-mail **:** salmaaswad1988@gmail.com
salmaabdullah1988@gmail.com