Dark Web Traffic Detection

Using Supervised Machine Learning

By

Sahra Zangeneh Nezhad

A Report Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering

In The Department of Electrical and Computer Engineering
University of Victoria

© Sahra Zangeneh Nezhad, 2023

University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

Supervisory Committee

Dr. Amirali Baniasadi, **Supervisor**
(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, **Committee Member**
(Department of Electrical and Computer Engineering)

Contents

List of Figures: 5
List of Tables 6
Acronyms 7
Acknowledgment 9
Dedication 10
Abstract 11
Chapter 1 12
Introduction 12
1.1 Overview 12
1.2 Motivation 14
1.3 Related work 14
1.4 Report outline 15
Chapter 2 16
2 Background 16
2.1 Virtual Private Network 16
2.2 The Onion Router 17
2.3 Machine Learning 18
2.4 Machine Learning Algorithm Types 18
2.4.1 Supervised Learning 19
2.4.2 Unsupervised Learning 19
2.4.3 Semi-Supervised Learning 19
2.4.4 Reinforcement Learning 20
2.5 WEKA 20
Chapter 3 22
3.1 CIC-Darknet2020 Dataset 22
3.2 Proposed Framework 24
3.3 Preprocessing 24
3.3.1 Data Cleaning 24
3.3.2 Data Transformation 25
3.4 Data splitting 26
3.5 Machine Learning Classifiers 26

3.5.1 Random Forest 26
3.5.2 Naïve Bayes 27
3.5.3 Support Vector Machine 27
3.5.4 Decision Tree J48 28
Chapter 4 28
4.1 Model Building and Training 28
4.2 Performance Evaluation 28
4.3 Evaluation Metrics 29
4.4 Performance with 5-fold Cross-Validation 29
4.5 Performance with 10-fold Cross Validation 30
4.6 Performance with 66/34 Split 31
4.7 Performance with 80/20 Split 31
4.8 Discussion 32
Chapter 5 32
5.1 Conclusion and Future Work 33
Bibliography 34

List of Figures:

Figure 1:Layers of Internet[6] 13
Figure 2 VPN Configuration [12] 17
Figure 3 How Onion Routing Works[14] 17
Figure 4Machine Learning Algorithms [18] 18
Figure 5The WEKA GUI Interface 21
Figure 6 samples of benign and darknet traffic 23
Figure 7 Proposed Framework 24
Figure 8 NaN values in Dataset 25

List of Tables

Table 1Darknet Network Traffic Details 22
Table 2 Count of Traffic Type 23
Table 3Number of samples of benign and darknet traffic 23
Table 4 System Parameters 28
Table 5 5-Fold Cross Validation Results 30
Table 6 10-Fold Cross Validation Results 30
Table 7 66/34 Split Results 31
Table 8 80/20 Split Results 32

Acronyms

AI: Artificial Intelligence

CPU: Central Processing Unit

GUI: Graphical User Interface

IP: Internet Protocol

IT: Information Technology

KNN: K-Nearest Neighbor

ML: Machine Learning

NB: Naïve Bayes

RAM: Random Access Memory

RF: Random Forest

SMOTE: Synthetic Minority Oversampling Technique

SVM: Support Vector Machine

DT: Decision Tree

WEKA: Waikato Environment for Knowledge Analysis

VPN: Virtual Private Network

TOR: The Onion Router

CNN: Convolutional Neural Network

VoIP: Voice over Internet Protocol

WAN: Wide Area Network

ANN: Artificial Neural Network

-NN: Not specified

MLP: Multilayer Perceptron

RF: Random Forest

DT: Decision Tree

GBDT: Gradient Boosted Decision Tree

DNN: Deep Neural Network

LSTM: Long Short-Term Memory

PCA: Principal Component Analysis

GRU: Gated Recurrent Unit

ISP: Internet Service Provider

TCP: Transmission Control Protocol

API: Application Programming Interface

Acknowledgment

I am deeply grateful to Professor Amirali Baniasadi for their consistent support and guidance during my time at the University of Victoria. Their expertise and dedication to their students have been invaluable to my academic and professional growth. I am thankful for the time and effort they generously invested in my development as a researcher, and I will always cherish their teachings as I progress in my studies and career. Thank you, Professor Banisadi, for everything you have done for me.

Dedication

To my mother and husband, whose support, encouragement, guidance, and prayers were invaluable to me, I offer my heartfelt thanks. Additionally, I am deeply grateful to my professors, instructors, friends, and colleagues who supported and encouraged me throughout this journey.

Abstract

The purpose of this study is to examine the feasibility of utilizing machine learning algorithms for distinguishing and categorizing VPN and TOR traffic on the dark web. The dark web, often referred to as the inaccessible or shadow aspect of the internet, is marked by its anonymity and inability to be indexed by search engines, making it a common platform for illegal activities such as drug trafficking, money laundering, and cybercrime. Both Virtual Private Networks (VPNs) and The Onion Router (TOR) are commonly employed technologies for anonymizing web traffic and accessing the dark web. While these technologies can be used for legitimate purposes, such as protecting the privacy and bypassing internet censorship, they can also be exploited by cybercriminals.

To achieve our objective, we will leverage a dataset of dark web traffic, specifically, the CIC-Darknet2020 dataset, which comprises a comprehensive and diverse collection of network traffic captures from the dark web, incorporating traffic features from both The Onion Router (TOR) and Virtual Private Network (VPN) technologies. Our model will be constructed using supervised machine learning methods, specifically classification algorithms including Random Forest (RF), Support Vector Machine (SVM) , Naive Bayes (NB) , and the Decision Tree (J48) classifiers. The experiments will be performed using five-fold and ten-fold cross-validation, and 66/34 and 80/20 percentage splits, utilizing the open-source software WEKA. The performance of the model will be evaluated based on parameters such as execution time, accuracy, precision, F-measure, and recall.

The results of this study indicate that the Decision Tree (J48) classifier surpasses the other classifiers in terms of accuracy, achieving 99.6% accuracy with an execution time of 15 seconds for a ten-fold cross-validation.

Chapter 1
Introduction

1.1 Overview

The Internet is a global network of interconnected computer networks that allows information and data to be exchanged. It consists of a range of interconnected networks and devices, including computers, servers, and other connected devices. The World Wide Web (WWW), also known as the surface web, is the indexed portion of the internet and publicly accessible through a web browser and found by traditional search engines such as Google.

The Deep Web, also referred to as the Invisible Web, is an extensive portion of the internet that is not indexed by search engines and hence cannot be reached simply with a web browser. It includes content such as government databases, medical records, bank statements, and other sensitive information not intended for public access [1].

The Dark web, on the other hand, is a portion of the internet that can only be accessed by using specialized software, such as the Tor browser, and is known for its anonymous nature. It is often used for illegal activities, including the sale of drugs, weapons, and stolen data, as well as for exchanging illegal information and materials [2][6].

The growth of the Internet has revolutionized global communication, allowing individuals to connect with each other from anywhere in the world with an Internet connection. However, during the early stages of its development, privacy and anonymity were not prioritized, leaving individuals vulnerable to tracking and tracing. In response to these concerns, a team of computer scientists and mathematicians at the Naval Research Laboratory (NRL), a division of the United States Navy, initiated the development of a new technology referred to as Onion Routing or TOR, aimed at addressing the privacy concerns of individuals [2].

In the process of anonymous communication, the sender must first transmit the data to the Onion Routing Overlay Layer before it reaches the receiver. The Onion Router acts as an intermediary between the sender and receiver and creates a sequence of nodes in a chain, the number of which depends on the destination of the receiver. The first node is connected to the sender, and the final node is connected to the receiver. The Onion Routing chain does not reveal the IP address, instead, it employs a sequence of bits to communicate with the nearest neighbor, making it difficult for an attacker to uncover the user's private information as the chain can only determine its immediate neighbor router node [3].

Another technology that enables secure and private communication over the internet is Virtual Private Network (VPN). VPN establishes a virtual point-to-point connection through tunneling protocols over existing networks. This resulted in creating a secure, encrypted tunnel between the user's device and the VPN server. All data transmitted through this tunnel is encrypted and

protected from unauthorized access, ensuring the privacy and security of the user's online activities [4].

It is crucial to identify dark web traffic to prevent cybercrime activities like hacking, espionage, and advanced persistent threats. This is where machine learning techniques for identifying and classifying VPN and TOR traffic in the dark web come into play. Such methods can help cybersecurity professionals, organizations, and government agencies to detect and prevent cybercrime on the dark web.

In this study, we aimed to address the challenge of detecting VPN and TOR traffic by leveraging machine learning techniques. The use of machine learning algorithms in this area offers several advantages over traditional methods, including increased accuracy, scalability, and the ability to handle large amounts of data.

To this end, we utilized the CIC-Darknet2020 dataset, a comprehensive and publicly available dataset that contains a wide range of network traffic data, to train and evaluate our machine learning models. Our analysis considered four different classifiers: Random Forest (RF), Support Vector Machines (SVM), Naive Bayes (NB) and C4.5 (J48). These classifiers were chosen based on their robustness and success in previous network traffic classification tasks.

Our analysis showed that the Decision Tree (j48) classifier outperformed the other classifiers in terms of accuracy, with a rate of 99.6%. This high accuracy demonstrates the effectiveness of machine learning techniques for detecting VPN and TOR traffic and provides insights for future research in this field.

In conclusion, This study emphasizes the significance of detecting VPN and TOR traffic as well as the potential of machine-learning techniques. Furthermore, the results obtained from our analysis can be used to improve the security and privacy of network traffic and contribute to the development of more sophisticated and effective methods for detecting malicious network activities.

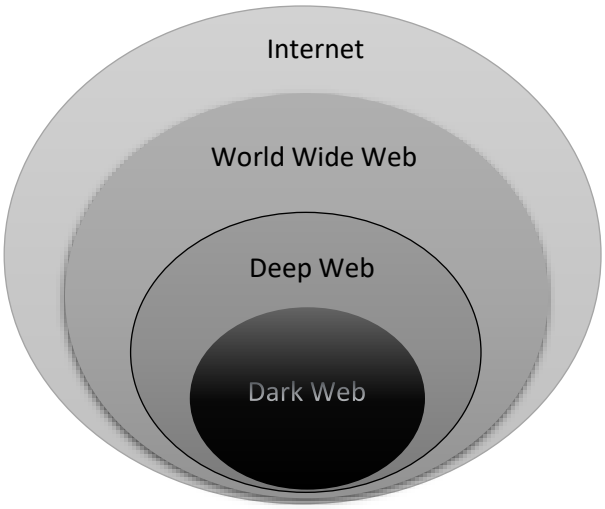

Figure 1:Layers of Internet[6]

1.2 Motivation

The widespread use of Virtual Private Networks (VPNs) and The Onion Router (TOR) network has created both opportunities and challenges for organizations and individuals alike. While VPNs and TOR provide users with much-needed privacy and security in their online activities, they can also be used to hide illegal or malicious activities. This highlights the importance of developing effective methods for detecting VPN and TOR traffic.

The rapid growth of network traffic and the increasing complexity of network security threats have motivated the need for efficient and accurate solutions for detecting VPN and TOR traffic. With their ability to handle large amounts of data and identify complex patterns in network traffic, machine learning algorithms offer a promising solution to this challenge.

This project addresses the need for effective VPN and TOR traffic detection by leveraging machine learning techniques. By using the CIC-Darknet2020 dataset and several state-of-the-art machine learning algorithms, we aim to demonstrate the feasibility and effectiveness of using machine learning for this task. Our research results can provide valuable insights for organizations and individuals in improving the security and privacy of their online activities and contribute to developing more advanced methods for detecting malicious network activities.

1.3 Related work

Detecting darknet traffic has been addressed by a number of researchers. There are, however, just a few public darknet databases accessible [5]. created the CIC-Darknet2020 dataset that was utilized in the studies described in this study. This dataset has also been utilized in previous studies and it has become a well-known darknet traffic dataset due to its ease of access.

In [5] researchers classified Tor and VPN traffic as darknet traffic, whereas non-Tor and non-VPN traffic was classified as innocuous traffic (Clearnet) using Convolutional Neural Network. Their CNN model classified traffic as darknet or benign with an overall accuracy of 94% and the application type utilized to produce the traffic with an accuracy of 86%. The application traffic was classified as browser, chat, email, file transfer, peer-to-peer, audio streaming, video streaming, or VOIP. In [6] they further divided the application categories into 11 subcategories and classified the data using Weighted Agnostic Neural Networks (WANN). WANNs, unlike ordinary ANNs, do not alter neuron weights but rather their own network architecture piecemeal. WANNs rank alternative designs based on performance and complexity, building new network layers from the architecture with the highest ranking. Their top WANN model has an application layer classification accuracy of 92.68%.

The research [7] concentrated solely on traffic type using the same dataset. They performed classification using -Nearest Neighbors (-NN), Multi-layer Perceptron (MLP), RF, DT, and

Gradient-Boosting Decision Trees (GBDT). Similar to [5], they divided the data into two categories for binary classification: benign and darknet. They employed the basic four traffic type classes for the multi-class challenge (Tor, non-Tor, VPN or non-VPN). They discovered that RF was the most successful traffic classifier, with F1-scores of 98.61% for multi-class classification. [8] employed Deep Neural Networks to categorize Tor and non-Tor traffic using the UNB-CIC Tor and non-Tor dataset, also known as ISCXTor2016 [9] (DNN). They created two models, DNN-A with three layers and DNN-B with five layers. DNN-A distinguished Tor samples from non-Tor samples with 98.81% accuracy, whereas DNN-B did so with 99.89% accuracy. The study described in [10] involved identifying traffic and application type using a CNN and two other deep-learning techniques: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). They extracted 20 characteristics using Principle Component Analysis (PCA), Decision Trees (DT), and Extreme Gradient Boosting (XGBoost) before putting the data into CNN-LSTM and CNN-GRU architectures. Their CNN layer was used to extract features from the input data, while LSTM and GRU were utilized to predict sequences based on these features. CNN-LSTM using XGBoost as the feature selection generated the highest F1-scores, with 96% classifying traffic type and 89% classifying application type.

1.4 Report outline

The remainder of this paper is structured as follows.

Chapter 1: it defined the introduction and considers related work on darknet traffic detection.

Chapter 2: gives a brief background discussion of Tor, VPN, Machine Learning Algorithms and WEKA.

Chapter 3: describes the dataset used in our experiments and provides background knowledge on the machine learning techniques used in our experiments and gives implementation details.

Chapter 4: discusses the results of our experiments

Chapter 5: summarizes our research and considers possible directions for future work.

Chapter 2

2 Background

In this part, we will first go through the two primary groups of data in our dataset, which are Tor and VPN traffic. In addition, we explain several machine learning approaches and strategies that have been used in our study.

2.1 Virtual Private Network

A Virtual Private Network (VPN) is a technology solution that creates a secure and encrypted communication channel between a user and a network. This secure connection enables the user to remain anonymous while accessing network resources. To utilize a VPN, a user must download and install a VPN client, which is software provided by VPN service providers. The installation and configuration of this client software are generally straightforward, and it is compatible with a broad range of devices and operating systems. The user runs the client software and selects the desired VPN server to connect to. Additionally, the user may configure connection options such as the TCP/UDP connection type and VPN protocol. Upon activation, the VPN client initiates the encryption process, transforming data from a readable format to an encoded format. Only the VPN client and server, possessing the decryption key, can convert the data back to its original readable form. The level of encryption is determined by the protocol used, with stronger encryption providing higher levels of security at the cost of slower data transfer speeds [11].

Once encryption is established, the VPN client creates an encrypted data tunnel between the VPN server and the user's Internet Service Provider (ISP). This tunnel ensures that outside entities cannot view the transmitted data. The VPN server then masks the user's IP address, substituting it with its own, and decrypts and forwards the incoming data to the Internet. Conversely, incoming data from the Internet is encrypted and sent to the VPN client for decryption. In conclusion, a VPN provides a secure and encrypted communication channel for users to access network resources while maintaining anonymity. The client software and encryption protocol determine the level of security and data transfer speed [12][13].

Figure 2 VPN Configuration [12]

2.2 The Onion Router

TOR is an anonymous and encrypted network, a free and open-source service used for anonymous communication, developed by the US Naval Research Laboratory using "Onion Routing" as its main idea. [14] Onion Routing is a distributed overlay network that was created to anonymize TCP-based services such as web browsing, secure shell, and instant messaging. Clients select a network path and construct a circuit in which each node in the path knows its predecessor and successor but not the other nodes in the circuit. Traffic flows in fixed-size cells down the circuit, which are unwrapped by a symmetric key at each node (similar to the layers of an onion) and transmitted downstream [15].

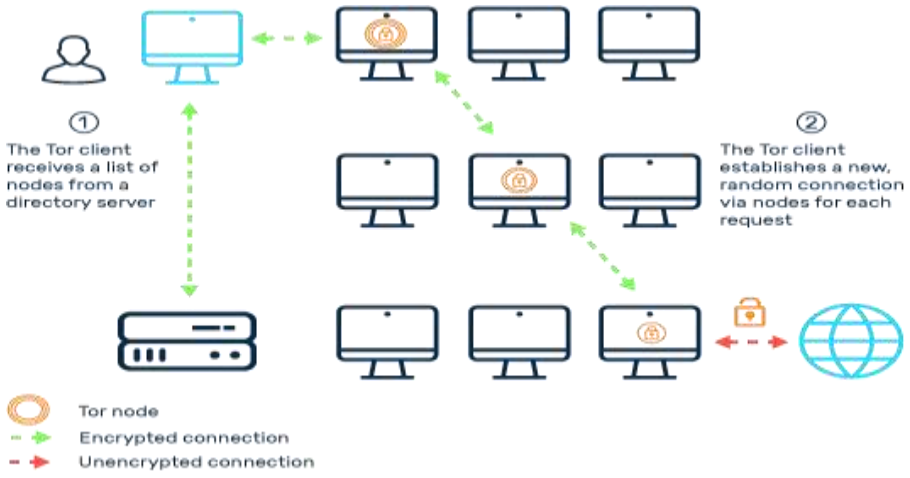

Figure 3 How Onion Routing Works[14]

Moreover, TOR allows the user to access the Deep Web or Darknet, which is used by cybercriminals, journalists, whistleblowers, and political activists. The TOR network's hidden services include a wealth of information, including information on Intranet systems, confidential government documents, individual personal accounts, stolen credit cards, and even drug

trafficking. It is possible to connect to the TOR network using the TOR browser bundle, a Firefox addon [15][16].

2.3 Machine Learning

Machine learning is a growing field of computational algorithms that aim to simulate human intelligence by learning from their surroundings. The goal of machine learning is to learn to process sensory (input) data in order to achieve a goal. A machine learning algorithm is a computing process that uses input data to complete a task without being explicitly programmed to do so. These algorithms are "soft programmed" in the sense that they naturally adjust or adapt their design with repetition, developing better and better at completing the target objective. Training is the adaption process in which samples of input data are supplied together with intended consequences. The algorithm is then ideally configured so that it cannot only create the desired outcome when confronted with the training inputs, but also generalize to achieve the desired outcome in the future [17].

2.4 Machine Learning Algorithm Types

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Reinforcement Learning

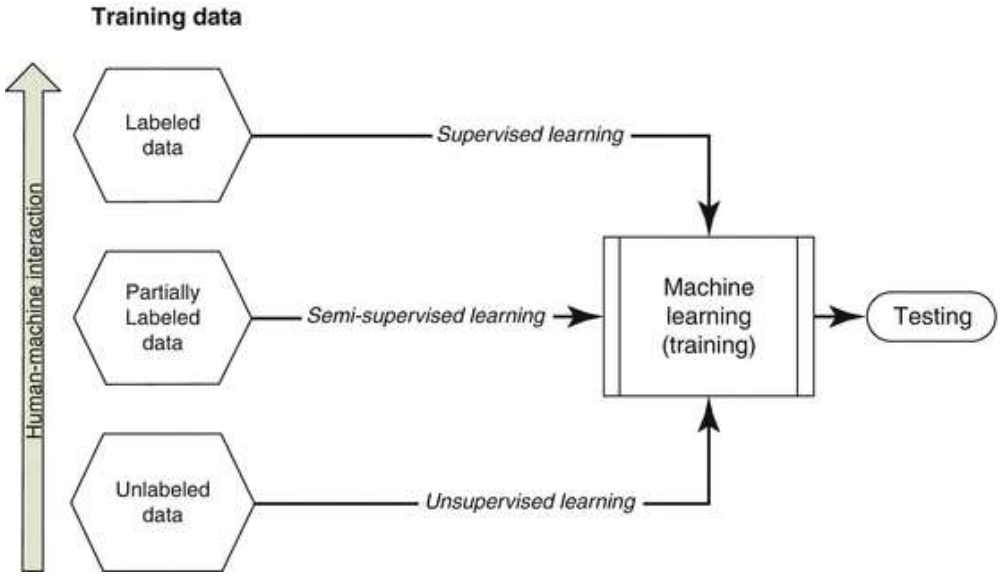

Figure 4Machine Learning Algorithms [18]

2.4.1 Supervised Learning

A subcategory of machine learning and artificial intelligence is supervised machine learning. It is distinguished by the use of labeled datasets to train algorithms that accurately classify data or predict outcomes. As input data is fed into the model, the weights are adjusted until the model is well-fitted, which occurs as part of the cross-validation process. A training set is used in supervised learning to teach models to produce the desired output. This training dataset contains both correct and incorrect outputs, allowing the model to learn over time. The algorithm evaluates its accuracy using the loss function and adjusts until the error is sufficiently minimized [19].

When it comes to data mining, supervised learning can be divided into two sorts of problems: classification and regression:

Classification: An algorithm is used in classification to accurately allocate test data to certain categories. It identifies specific entities within the dataset and tries to derive conclusions about how those items should be labeled or described. Linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbors, and random forest are examples of common classification techniques [19].

Regression: this is a statistical method for determining the relationship between dependent and independent variables. It is widely used to produce forecasts, such as those for a company's sales revenue. Popular regression algorithms include linear regression, logistic regression, and polynomial regression [19].

2.4.2 Unsupervised Learning

Unsupervised learning is a form of machine learning that operates on unlabeled data, in contrast to supervised learning which relies on labeled data. This approach aims to uncover patterns and relationships within a dataset in order to resolve clustering or association difficulties. [18] When subject matter experts are uncertain about the properties of a dataset, unsupervised learning proves to be especially valuable. Common clustering algorithms used in unsupervised learning include hierarchical clustering, k-means, and Gaussian mixture models. Unsupervised machine learning techniques detect patterns from a dataset without the use of previously known or labeled outcomes. As a result, unsupervised methods cannot be immediately applied to regression or classification problems, as the values of the output data are unknown, making it challenging to train the algorithm. On the other hand, unsupervised learning can be employed to uncover the intrinsic structure of the data.[20]

2.4.3 Semi-Supervised Learning

Semi-supervised learning is a field of machine learning that leverages both labeled and unlabeled data to perform specific learning tasks. By combining large amounts of unlabeled data with smaller sets of labeled data, semi-supervised learning provides a compromise between supervised and unsupervised learning.[21]

Semi-supervised classification methods are especially relevant in situations where labeled data is scarce. In such cases, building a reliable supervised classifier can be a challenge. This is often seen in applications where collecting labeled data is expensive or difficult, such as computer-aided diagnosis, drug development, and part-of-speech tagging. Overall, semi-supervised learning provides a valuable solution for utilizing the vast amounts of available unlabeled data in combination with limited labeled data to perform effective learning tasks.[21]

2.4.4 Reinforcement Learning

Reinforcement learning is a type of machine learning that deals with teaching an agent to make decisions in an environment by performing certain actions and receiving rewards or punishments. It is a paradigm for modeling decision-making processes in dynamic and uncertain environments. The agent learns to maximize its rewards over time by adjusting its actions based on the feedback it receives. The learning process in reinforcement learning occurs through trial and error, where the agent tries different actions and learns from their experiences [22]

2.5 WEKA

WEKA (Waikato Environment for Knowledge Analysis) is a well-known and widely used open-source software for machine learning and data mining [23] Developed at the University of Waikato in New Zealand, WEKA provides a rich set of algorithms and tools for various data analysis tasks, including data pre-processing, classification, regression, clustering, association rule mining, and visualization [24] The software is designed to be user-friendly and easy to use, with a graphical user interface and a simple, intuitive Application Programming Interface (API) [23].

WEKA is popular among academic researchers and practitioners, as it offers a comprehensive suite of machine learning algorithms, including popular algorithms like decision trees, support vector machines, and neural networks [25]. Additionally, the platform supports various data formats however .arff is the native file format and is compatible with a wide range of operating systems, making it accessible to a broad audience [23].

WEKA software has several tabs on its graphical user interface (GUI) that provide access to different functionalities and tools for data analysis tasks. The main tabs are:

- Explorer: This is the main tab for performing machine learning tasks. It provides a visual interface for data pre-processing, selecting and applying algorithms, evaluating model performance, and visualizing results.

- Experimenter: This tab allows you to perform experiments to compare the performance of different algorithms on a particular dataset. You can set up and run multiple experiments and compare the results.

- Knowledge Flow: This tab provides a visual interface for building and executing machine learning pipelines. It allows you to connect pre-processing, modeling, and evaluation components in a flow-based interface, making it easy to visualize and understand the steps involved in the analysis.

- Simple CLI: This tab provides a command line interface for WEKA. It allows you to perform machine learning tasks using WEKA's Java API, making it useful for automating tasks or integrating WEKA with other tools.

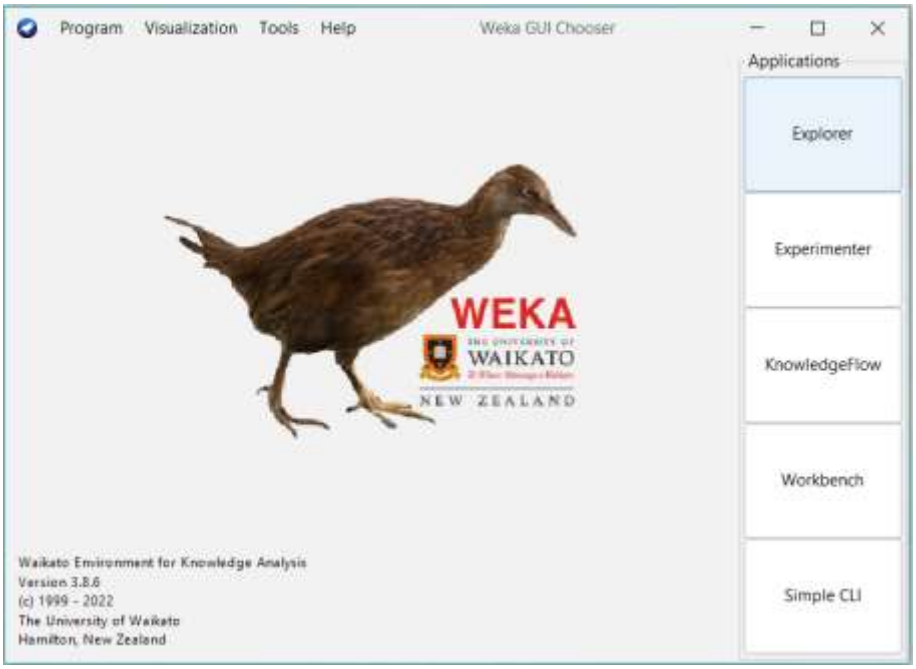

Figure 5 The WEKA GUI Interface

Chapter 3

3.1 CIC-Darknet2020 Dataset

The CIC-Darknet2020 dataset is a cybersecurity dataset released by the Canadian Institute for Cybersecurity (CIC) in 2020. The dataset contains network traffic data captured from various simulated scenarios, including normal and malicious traffic patterns. The goal of the CIC-Darknet2020 dataset is to provide researchers and practitioners with a representative and diverse set of network traffic data for the development, evaluation, and testing of cybersecurity techniques [26].

The CICDarknet2020 dataset is organized into two tiers, where the first tier comprises a two-layered method for generating benign and darknet traffic. The second layer of the darknet traffic incorporates a diverse range of traffic scenarios, including Audio-Stream, Browsing, Chat, Email, P2P, Transfer, Video-Stream, and VOIP. In order to create a representative dataset, the CICDarknet2020 dataset integrates previous datasets, including ISCXTor2016 and ISCXVPN2016, and merges their respective VPN and Tor traffic into corresponding darknet categories. As outlined in Table 1, the different types of darknet traffic and the programs that generate network traffic are described in detail [26].

Table 1Darknet Network Traffic Details

Traffic Category	Applications used
Audio Stream	Vimeo and YouTube
Browsing	Firefox and Chrome
Chat	ICQ, AIM, Skype, Facebook and Hangouts
Email	SMTPS, POP3S and IMAPS
P2P	uTorrent and Transmission (BitTorrent)
Transfer	Skype, FTP over SSH (SFTP) and FTP over SSL (FTPS) using Filezilla and an external service
Video Stream	Vimeo and Youtube
VOIP	Facebook, Skype and Hangouts voice calls

Based on the previous section's combining explanation, table2 shows the number of samples each traffic type.

Table 2 Count of Traffic Type

Traffic Type	Count
P2P	48520
Browsing	32808
AUDIO-STREAMING	18064
Chat	11478
File-Transfer	11182
Video-Streaming	9767
Email	6145
VOIP	3566
Grand Total	**141530**

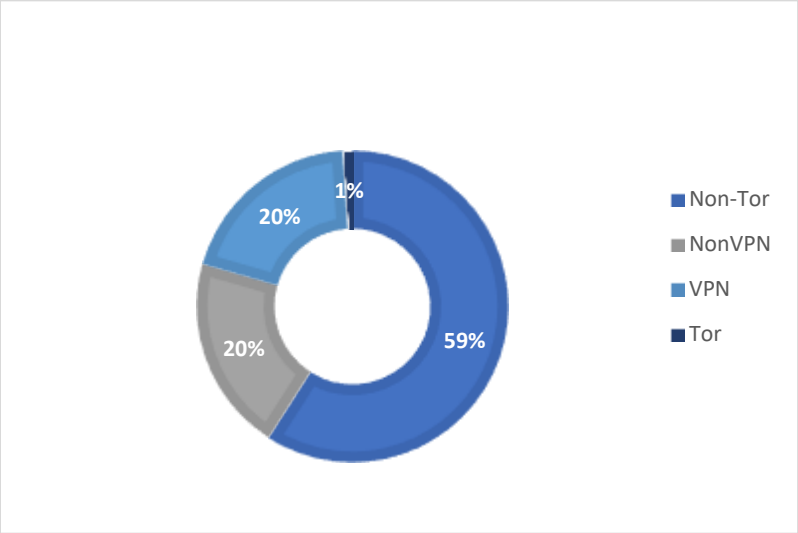

Figure 5 samples of benign and darknet traffic

Table 3Number of samples of benign and darknet traffic

Traffic Sample	Count
Non-Tor	93356
NonVPN	23863
VPN	22919
Tor	1392
Grand Total	**141530**

3.2 Proposed Framework

In the present study, our emphasis was on the first layer of the CICDarknet2020 dataset, which consisted of data samples labeled as dark web traffic (VPN and TOR) and benign traffic (Non-VPN and Non-Tor). In the preprocessing stage, we applied a range of techniques to refine the data and make it suitable for the classification phase. Subsequently, we employed four diverse classifiers and utilized four distinct cross-validation techniques to partition the training data. To determine the most effective approach, we conducted a comprehensive evaluation of the performance metrics of the different classifiers. A flowchart is presented to provide a comprehensive overview of the methodology adopted in this project.

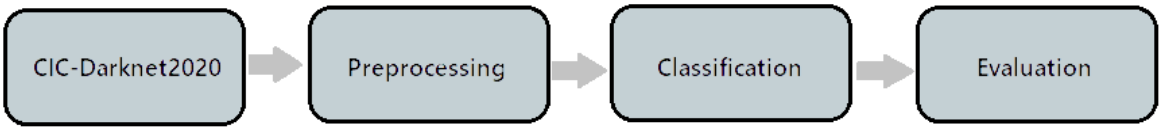

Figure 6 Proposed Framework

3.3 Preprocessing

3.3.1 Data Cleaning

The act of removing or modifying data that is corrupt, missing, unnecessary, redundant, or poorly formatted in order to prepare it for analysis is known as data cleaning.

The CIC-Darknet2020 dataset contains samples with missing data, notably "NaN" feature values. In our data cleaning phase, we delete samples with these values. Previous work with this dataset excluded the flow labels, particularly, Flow Id, Timestamp, Source IP, and Destination IP. The Flow Id and Timestamp, which are likewise removed in our study.

Moreover, the dataset has zero value columns which has been removed. After removing these attributes, dataset attributes reduced to 68 in total which was 85 before.

Also we used remove duplicate filter in WEKA in order to remove duplicate values which reduced the number of values from 141530 to 117033.

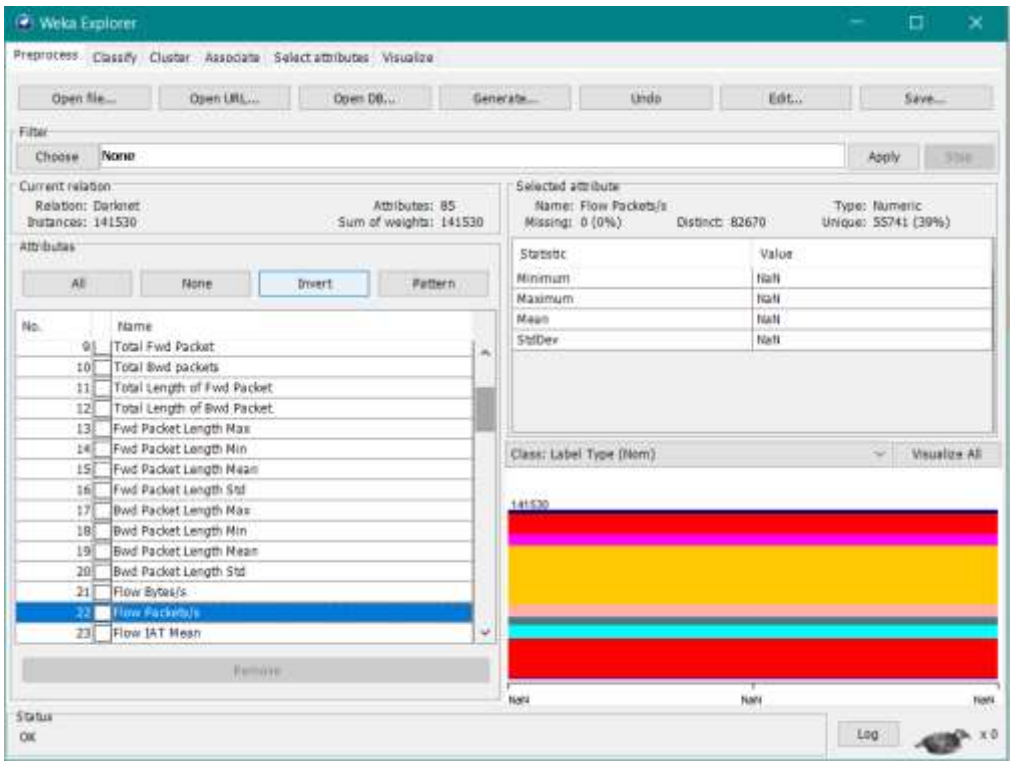

Figure 7 NaN values in Dataset

3.3.2 Data Transformation

The CIC-Darknet2020 dataset likely contains a mix of features with different scales and ranges. Normalizing the data in this dataset means transforming the values of the features so that they are on the same scale and have similar ranges. This is important because some machine learning algorithms are sensitive to the scale of the features and can perform poorly when the features have vastly different scales. Normalizing the data in the CIC-Darknet2020 dataset ensures that each feature contributes equally to the analysis and helps prevent features with larger values from dominating the analysis. Additionally, normalizing the data can help improve the convergence and training speed of some machine learning algorithm. We used Normalize filter in WEKA in order to normalize our data.

Moreover, The CIC-Darknet2020 dataset may contain data that has an inherent structure or ordering, such as time-series data. Randomizing the data in this dataset means rearranging the rows of the data in a random manner, so that any inherent structure or ordering is removed. This is important because some machine learning algorithms can be sensitive to the order of the data and may give different results depending on the order of the data. Randomizing the data in the CIC-Darknet2020 dataset helps to ensure that the results of the machine learning algorithms are reliable and not biased towards any specific pattern in the data. Additionally, randomizing the data can help improve the generalization performance of the models, as well as improve cross-

validation performance by ensuring that the data is split into training and validation sets in a random manner.

3.4 Data splitting

In this study, data splitting was performed as a crucial preprocessing step in order to evaluate the performance of machine learning algorithms. Four different data splitting techniques were utilized, including 5-fold and 10-fold cross-validation, as well as 66 and 80 percentage split. Cross-validation is a widely used technique in machine learning, where the original dataset is divided into several folds, and the model is trained and evaluated several times, each time using a different fold for testing. The results from the different folds are then aggregated to obtain a more robust estimate of the model's performance. The 5-fold and 10-fold cross-validation techniques were used to assess the performance of the machine learning algorithms on the CIC-Darknet2020 dataset. Additionally, the 66 and 80 percentage split were used to evaluate the performance of the algorithms, where a 34 and 20 percentage of the data was reserved for testing, and the remaining data was used for training the models. The choice of the specific data splitting technique used in this study was informed by the size of the CIC-Darknet2020 dataset and the requirements of the machine learning task. The use of multiple data splitting techniques helps to ensure that the results obtained in this study are reliable and representative of the performance of the machine learning algorithms on the CIC-Darknet2020 dataset.

3.5 Machine Learning Classifiers

In order to detect VPN and TOR traffic in the CIC-Darknet2020 dataset, four different classifiers were used in this study. One of the classifiers used was Random Forest (RF), which has been used in previous researches for similar purposes. The use of RF in this study was motivated by the desire to compare the results obtained with those reported in previous studies, and to assess the performance of the classifier in the context of VPN and TOR traffic detection. By using RF as one of the classifiers in this study, the results can be compared to previous findings and provide a benchmark for future research in this area. The use of multiple classifiers in this study helps to ensure that the results are robust and not dependent on a single classifier, and also provides insight into the relative performance of different classifiers for VPN and TOR traffic detection.

3.5.1 Random Forest

Random Forest (RF) is an ensemble learning method for classification and regression problems in machine learning. It is based on the concept of decision trees, where multiple decision trees are trained on different subsets of the training data and their predictions are combined to produce a final prediction. RF is considered a robust and accurate machine learning method due to its

ability to reduce the variance of individual decision trees and its ability to handle high-dimensional and complex data.

In a Random Forest classifier, each decision tree is trained on a different bootstrapped sample of the training data, and the final prediction is made by taking a majority vote of the predictions of individual decision trees. This process helps to reduce overfitting and improve the generalization performance of the classifier. The effectiveness of Random Forest has been demonstrated in numerous studies and applications, including in areas such as bioinformatics, computer vision, and natural language processing. One of the key advantages of using Random Forest is its ability to handle imbalanced datasets and noisy data, and its ability to provide feature importance scores, which can be used for feature selection and interpretation [27].

3.5.2 Naïve Bayes

Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. It is based on Bayes' theorem, which states that the probability of a class given a set of features is proportional to the likelihood of the features given the class, multiplied by the prior probability of the class. In a Naive Bayes classifier, the features are assumed to be conditionally independent given the class, and the prediction for a new instance is made based on the maximum a posteriori (MAP) estimate of the class. The algorithm can be implemented in several forms, including Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, depending on the type of data and the distribution assumptions made. Naive Bayes is a fast and simple algorithm that works well with large datasets and high-dimension data. It is often used in text classification, spam filtering, and sentiment analysis, among other applications. One of the disadvantages of the algorithm is its "Naive" assumption of independence among features, which may not always hold in practice, leading to suboptimal performance [28].

3.5.3 Support Vector Machine

Support Vector Machine (SVM) is a popular machine learning algorithm for classification and regression tasks. It is based on the idea of finding the hyperplane that best separates the data into different classes. The hyperplane is chosen so that it maximizes the margin, which is the distance between the closest instances of each class to the hyperplane. These closest instances are called support vectors, hence the name "Support Vector Machine" In SVM, the data is transformed into a higher-dimensional space using a kernel function, which allows for non-linear decision boundaries. SVM is known for its ability to handle high-dimensional and sparse data, as well as its ability to handle problems with class imbalance [29].

3.5.4 Decision Tree J48

J48 is a decision tree algorithm that is implemented in the WEKA machine learning library. It is an implementation of the popular C4.5 algorithm, which is a decision tree induction algorithm used for classification. The algorithm builds a decision tree by recursively partitioning the data into smaller subsets based on the values of the features. Each internal node of the tree represents a test on a feature, and the branches represent the outcome of the test. The leaves of the tree represent the class predictions. J48 uses a greedy approach to choose the best feature to split the data at each node, based on a measure of information gain, such as entropy or gain ratio. The algorithm also prunes the tree to prevent overfitting, by removing branches that do not improve the accuracy of the tree. J48 is a simple and intuitive algorithm that is easy to understand and interpret, making it a popular choice for data mining and machine learning tasks. It is also relatively fast and efficient, making it a good choice for large datasets. However, it is prone to overfitting and can be affected by noisy or irrelevant features [30].

Chapter 4

4.1 Model Building and Training

In this step, the models are trained and evaluated using the CIC-Darknet2020 dataset and the features selected in the preprocessing stage. Evaluation is performed through five and ten-fold cross-validation, as well as 66/34 and 80/20 training/testing splits to assess the classifier's effectiveness.

4.2 Performance Evaluation

We present the results of our experiments on four supervised machine learning models: Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (J48). These tests were performed on a personal computer utilizing the hardware and software specifications listed in Table 4.

Table 4 System Parameters

Item	Specifications
Manufacturer	Lenovo
Model	ThinkPad E15
Operating System	Windows 11
System	64 bit OS, x64 based processor

Processor	11th Gen Intel(R)
RAM	16.0 GB
Core	Core™ i7
Machine Learning Tool	Version 3.8.6

4.3 Evaluation Metrics

Our evaluation metrics are designed to compare the performance of our machine learning classifiers. We use various metrics such as accuracy, precision, recall, and F1-score to determine the effectiveness of each classifier. The accuracy measures the proportion of correct predictions, while precision evaluates the ability of the classifier to not label as positive instances that are negative. Recall measures the ability of the classifier to find all positive instances, and F1-score is the harmonic mean of precision and recall, giving a balance between the two as well as execution time, the amount of time spent on training and testing the classification model. By comparing the results of these metrics, we can determine which classifier is the most effective in terms of both precision and recall.

- Accuracy $= \frac{TN + TP}{TN + TP + FN + FP}$
- Precision$= \frac{TP}{TP+FP}$
- Recall$= \frac{TP}{TP+FN}$
- F1 Score$= \frac{2\,TP}{2TP+FP+FN}$

True Positives (TP) are instances that are correctly classified as positive.

True Negatives (TN) are instances that are correctly classified as negative.

False Positives (FP) are instances that are incorrectly classified as positive.

False Negatives (FN) are instances that are incorrectly classified as negative.

4.4 Performance with 5-fold Cross-Validation

"5-fold cross validation" refers to a technique for evaluating the performance of a machine learning algorithm by dividing the original sample into five equal parts, or "folds". The algorithm is trained on four of the folds and tested on the remaining one. This process is repeated five times, with each of the five folds used exactly once as the test set. The results of the five tests are then averaged to give an overall performance measure. As results mentioned in table below, the results show random forest algorithm has highest accuracy with 90.5 seconds execution time, while decision tree (j48) ranked as second-best results with better execution time.

Table 5 5-Fold Cross Validation Results

Classifier	Accuracy	Precision	Recall	F1 Score	Execution time (s)
NB	77.52	81.2	77.5	75.	13
J48	98.19	98.2	98.2	98.2	22.4
RF	98.70	98.7	98.7	98.7	90.5
SVM	89.76	89.8	89.8	89.7	476

4.5 Performance with 10-fold Cross Validation

10-fold cross validation" means a similar technique as 5-fold cross validation, but with ten folds instead of five. The sample is divided into ten equal parts, or "folds". The algorithm is trained on nine of the folds and tested on the remaining one. This process is repeated ten times, with each of the ten folds used exactly once as the test set. The results of the ten tests are then averaged to give an overall performance measure of the algorithm. 10-fold cross validation provides a more robust estimate of the algorithm's performance than a single train/test split, as it makes use of all the data for both training and testing. In this test, Decision Tree (j48) ranked as the best algorithm with 99.60 percentage accuracy and lowest execution time.

Table 6 10-Fold Cross Validation Results

Classifier	Accuracy	Precision	Recall	F1 Score	Execution time (s)
NB	77.60	81.4	77.6	75.1	16
J48	99.60	99.6	99.6	99.6	15
RF	98.72	98.7	98.7	98.7	102.95
SVM	90.09	90.1	90.1	90.1	973.32

4.6 Performance with 66/34 Split

In WEKA, a "66/34 split" refers to the division of a sample into two parts, where 66% of the sample is used for training and the remaining 34% is used for testing. This is a common ratio for a train/test split in machine learning the goal of the split is to partition the data into two sets: one for training the machine learning algorithm and one for evaluating its performance. The performance of the algorithm is then measured based on its ability to make accurate predictions on the unseen test data. In this experiment, the Random Forest algorithm with 98.60 accuracy gave us the best results in the shortest amount of time.

Table 7 66/34 Split Results

Classifier	Accuracy	Precision	Recall	F1 Score	Execution time (s)
NB	89.47	92.3	89.5	90.6	9.73
J48	98.09	98.1	98.1	98.1	31.46
RF	98.60	99.9	98.6	98.6	11.21
SVM	89.67	89.7	89.7	89.7	2658.62

4.7 Performance with 80/20 Split

An 80/20 split refers to the division of a data sample into two parts, with 80% being allocated for training a machine learning model and the remaining 20% being used for testing. This split is commonly used to evaluate the performance of the model by comparing its predictions on the test data with actual results. The goal of this split is to create two distinct sets of data, one for

training the model and one for evaluating its accuracy. This ratio of 80/20 is different from the 66/34 split in that a larger portion of the data is used for training the model and a smaller portion is used for testing. In this test, Random Forest ranked with the best accuracy at 98.7 and Decision Tree with 98.26 and much less execution time ranked as the second best algorithm.

Table 8 80/20 Split Results

Classifier	Accuracy	Precision	Recall	F1 Score	Execution time (s)
NB	89.36	92.2	89.4	90.5	16.73
J48	98.26	98.3	98.3	98.3	28.79
RF	98.74	98.7	98.7	98.4	163
SVM	89.97	90.0	90.0	90.0	1682.83

4.8 Discussion

The results of the different evaluations conducted on the algorithms in this project provide valuable insights into their performance and efficiency. The 5-fold cross validation test showed that the Random Forest algorithm had the highest accuracy with a 90.5-second execution time, while the Decision Tree (J48) ranked as the second best with a better execution time. The 10-fold cross validation test, on the other hand, revealed that the Decision Tree (J48) was the best algorithm with 99.60 accuracy and the lowest execution time. The 66/34 split experiment showed that the Random Forest algorithm had the best results with 98.60 accuracy in the shortest amount of time. Finally, the 80/20 split test indicated that the Random Forest algorithm had the highest accuracy of 98.74, while the Decision Tree (J48) had a close second with 98.26 accuracy and a much lower execution time. These results highlight the strengths and weaknesses of each algorithm and can assist in selecting the most suitable one for a particular task.

Chapter 5

5.1 Conclusion and Future Work

Based on the results of the experiments conducted in this project, it can be concluded that both the Random Forest and Decision Tree (J48) algorithms perform well in detecting VPN and Tor traffic. The Random Forest algorithm had the highest accuracy in most of the tests. The Decision Tree (J48), on was faster in execution and had the best results in the 10-fold cross validation test with 99.60 accuracy.

While the current studies have achieved high accuracy in detecting darknet traffic, there is still room for improvement. Future work can include exploring different feature extraction methods, incorporating additional datasets for training and testing, and developing more advanced machine learning models for traffic classification. Another area for future work could be to study the performance of these models in real-time network environments, to better understand the practical applications of these techniques. Additionally, the study of privacy and security concerns related to darknet traffic can also be an interesting area of research.

Bibliography

[1] Finklea, K. M. (2015). Dark web.

[2] Kaur, S., & Randhawa, S. (2020). Dark web: A web of crimes. Wireless Personal Communications, 112, 2131-2158.

[3] Balasubramanian, K., & Kannan, S. (2019). Onion routing in anonymous network. Appl. Math, 13(S1), 247-253.

[4] Ezra, P. J., et al. (2022). Secured communication using virtual private network (VPN). In Cyber Security and Digital Forensics: Proceedings of ICCSDF 2021 (pp. 309-319).

[5] Lashkari, A. H., Kaur, G., & Rahali, A. (2020, October). Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning. In Proceedings of 10th International Conference on Communication and Network Security (ICCNS 2020) (pp. 1-13).

[6] Demertzis, K., Tsiknas, K., Takezis, D., Skianis, C., & Iliadis, L. (2021). Darknet traffic big-data analysis and network management for real-time automating of the malicious intent detection process by a weight-agnostic neural networks framework. arXiv preprint arXiv:2102.08411.

[7] Iliadis, L. A., & Kaifas, T. (2021, June). Darknet traffic classification using machine learning techniques. In 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST) (pp. 1-4). MOCAST.

[8] Sarkar, D., Vinod, P., & Yerima, S. Y. (2020, June). Detection of Tor traffic using deep learning. In Proceedings of IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-8).

[9] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017, February). Characterization of Tor traffic using time-based features. In 3rd International Conference on Information System Security and Privacy (ICISSP) (pp. 253-262).

[10] Sarwar, M. B., Hanif, M. K., Talib, R., Younas, M., & Sarwar, M. U. (2021). Darkdetect: Darknet traffic detection and categorization using modified convolution-long short-term memory. IEEE Access, 9, 113705-113713.

[11] Rathod, K., & Kondekar, S. (n.d.). Virtual private network.

[12] Waseem, A. (2022, November 23). The ultimate guide to VPNs: What is a VPN & what does it do? https://management.org/what-does-a-vpn-protect-you-from

[13] Costa, L. H. M. K., Fdida, S., & Duarte, O. C. M. B. (2000). An introduction to virtual private networks: towards D-VPNs. Networking and Information Systems Journal, 3(3/4), 575-594.

[14] Myra. (2022, May 10). What is the Tor network? Definition, how it works and much more. https://www.myrasecurity.com/en/tor-network/

[15] Syverson, P., Dingledine, R., & Mathewson, N. (2004). Tor: The second-generation onion router. In Usenix Security (pp. 303-320).

[16] Dutta, N., Jadav, N., Tanwar, S., Sarma, H. K. D., Pricop, E., Dutta, N., ... & Pricop, E. (2022). Tor: The onion router. In Cyber Security: Issues and Current Trends (pp. 37-55).

[17] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? Springer International Publishing.

[18] Ayodele, T. O. (2010). Types of machine learning algorithms. New Advances in Machine Learning, 3, 19-48.

[19] IBM Cloud Education. (2020, August). Supervised learning. https://www.ibm.com/cloud/learn/supervised-learning/

[20] DataRobot. (n.d.). Unsupervised machine learning. https://www.datarobot.com/wiki/unsupervised-machine-learning/

[21] Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. Machine Learning, 109(2), 373-440.

[22] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). Cambridge, MA: MIT Press.

[23] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter, 11(1), 10-18.

[24] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques (3rd ed.). San Francisco, CA: Morgan Kaufmann Publishers.

[25] Frank, E., & Witten, I. H. (2016). A practical guide to feature selection. Journal of Machine Learning Research, 15(1), 725-747.

[26] Lashkari, A. H., Kaur, G., & Rahali, A. (2020, November). DIDarknet: A contemporary approach to detect and characterize darknet traffic using deep image learning. In 10th International Conference on Communication and Network Security (pp. 1-6). Tokyo, Japan.

[27] Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.

[28] Baştanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. In MiRNomics: MicroRNA biology and computational analysis (pp. 105-128).

[29] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273-297.

[30] Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1, 81-106.