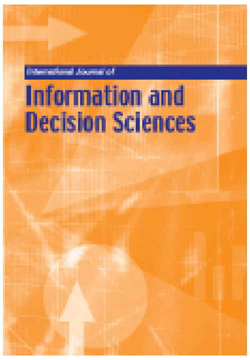


International Journal of Information and Decision Sciences

ISSN online: 1756-7025 - ISSN print: 1756-7017
https://www.inderscience.com/ijids

Discerning the traffic in anonymous communication networks using machine learning: concepts, techniques and future trends

Annapurna P. Patil, Lalitha Chinmayee M. Hurali

DOI: 10.1504/IJIDS.2023.10041362

Article History:

Received:	02 June 2020
Last revised:	22 December 2020
Accepted:	23 January 2021
Published online:	20 March 2023

Discerning the traffic in anonymous communication networks using machine learning: concepts, techniques and future trends

Annapurna P. Patil and
Lalitha Chinmayee M. Hurali*

Department of Computer Science and Engineering,
Ramaiah Institute of Technology (RIT),
Bangalore, India
Email: annapurnap2@msrit.edu
Email: lalithachinmayee@gmail.com
*Corresponding author

Abstract: With the growing need for anonymity and privacy on the internet, anonymous communication networks (ACNs) such as Tor, I2P, JonDonym, and Freenet have risen to fame. Such anonymous networks aim to provide freedom of expression and protection against tracking to its users. Simultaneously, there is also a class of users involved in the illegal usage of these ACNs. An emerging research topic in the field of ACNs is network traffic classification, as it can improve the network security against illegal users as well as improve the quality of service for its legal users. In this study, we review the research works available in the literature relevant to traffic classification in ACNs based on machine learning (ML) and also present to the researchers the general concepts and techniques in this area. A discussion on future trends in this area is also provided to bring out the future enhancements required in ML-based network traffic classification in ACNs.

Keywords: anonymous communication networks; ACNs; machine learning; traffic classification; Tor; network security.

Reference to this paper should be made as follows: Patil, A.P. and Hurali, L.C.M. (2023) 'Discerning the traffic in anonymous communication networks using machine learning: concepts, techniques and future trends', *Int. J. Information and Decision Sciences*, Vol. 15, No. 1, pp.94–115.

Biographical notes: Annapurna P. Patil received her PhD from Visvesvaraya Technological University (VTU), Belgaum, Karnataka, India in 2014. She is currently a Professor in Department of Computer Science and Engineering in Ramaiah Institute of Technology (RIT), Bangalore. She is a senior IEEE member, LMCSI, LMISTE, ACM member, and held position as chair, IEEE WIE, Bangalore Section, 2018. She has several publications in reputed conferences and journals. She is involved in collaborative works with CISCO, IBM, HPE, Nihon Communications Ltd, Samsung, Bangalore for various research projects. Her research interests span the area of mobile ad hoc networks, protocol engineering, artificial intelligence, data analytics, and distributed computing.

Lalitha Chinmayee M. Hurali received her BE degree in Electronics and Communication from R.V. College of Engineering, Bangalore, India in 2015. She is currently working towards her MTech degree in Ramaiah Institute of Technology (RIT), Bangalore. She is a student member at IEEE Computer Society. Her research interests are vehicular ad hoc networks, multimedia streaming applications, machine learning and network security.

1 Introduction

The usage of privacy and anonymity services on the internet has increased in recent years. One of the reasons for this is the demand for freedom of expression and protection against tracking and surveillance. Another reason is the availability of anonymous communication networks (ACNs) such as Tor, invisible internet project (I2P), Freenet, JonDonym, and many others. The ACNs provide anonymity to their users through encryption. Although the seekers of privacy and anonymity benefit from ACNs, there are also entities abusing them to perform illegal activities behind the anonymity (Peng, 2014; Rigby 1995).

One of the vital research areas that can improve the performance of ACNs and, at the same time, reduce their abuse is network traffic classification. Discerning the traffic in ACNs is helpful in surveillance systems to block anonymous traffic altogether, thus imposing censorship and enhance network security and in QoS provisioning based on application type running in the ACNs. Hence, network traffic classification in ACNs is an emerging and open field to researchers. Due to the encrypted traffic in the ACNs, traffic classification in ACNs is not as easy as in unencrypted networks (Rigby, 1995; Dainotti et al., 2012; Aceto and Pescape, 2015).

Historically, traffic classification has been performed using port-based methods in which traffic is identified solely based on ports used by an application. The port-based methods have become obsolete due to their inability to tackle dynamic ports and obfuscation techniques. Another method of traffic classification is payload based, which is complex and involves deep packet inspection (DPI). Nevertheless, due to the resource-intensive and sophisticated nature of the payload-based method, it is not much preferred. In recent years, ML based traffic classification has gained much interest from researchers, and hence, ML-based traffic classification in ACNs is an emerging and open field (Finsterbusch et al., 2013; Zuev and Moore, 2005).

Over the few years, some experiments have been conducted by the researchers on the ML-based traffic classification in ACNs with varied interests such as network security, to improve the design of ACNs, to curb the illegal activities on ACNs. In this paper, we review the works available in the literature concerning traffic classification in ACNs based on ML and provide a holistic view of the concepts, available techniques, and trends in this area. As per our knowledge, this is the first of a kind review of ML-based traffic classification in ACNs. With this article, we aim to provide the state of art techniques available in the field of traffic classification in ACNs.

The main contributions of this paper are:

- A comprehensive workflow of an ML model and its application are discussed to provide a background for the ML-based traffic classification.
- We discuss the need for traffic classification in ACNs in addition to its benefits. A discussion of the traditional methods of traffic classification and their limitations compared to ML-based methods is also provided.
- An architectural overview of ACNs such as Tor, I2P, JonDonym, and Freenet is discussed along with its working and features.
- We provide a comparative analysis of the reviewed works covering the steps in ML-based traffic classification, such as dataset generation, data preparation, algorithms, and model evaluation.
- Finally, we provide future directions for traffic classification in ACNs in accordance with the reviewed literature.

The remainder of this paper is organised as follows: Section 2 briefly provides the workflow of ML and its applications. It also discusses the need for traffic classification in ACNs and the merits of ML-based traffic classification. In Section 3, we explain the architecture and working of a few ACNs. Section 4 discusses the methodology of ML-based traffic classification in ACNs and also provides a comparison of the reviewed works. Finally, Section 5 provides a discussion on challenges and future trends in ML-based traffic classification in ACNs and concludes the review.

2 Background

2.1 Machine learning

ML is a technique of building a predictive model from past historical data, which is used to make predictions for new data. ML techniques help in making faster decisions and find applications in various domains such as fraud detection, machine automation, and network security (Mitchell, 1997; Domingos, 2012).

2.1.1 The workflow of ML

Figure 1 shows the workflow of a ML model. The steps involved in developing an ML model are data collection, data preparation, model training, and model evaluation (Wang et al., 2017).

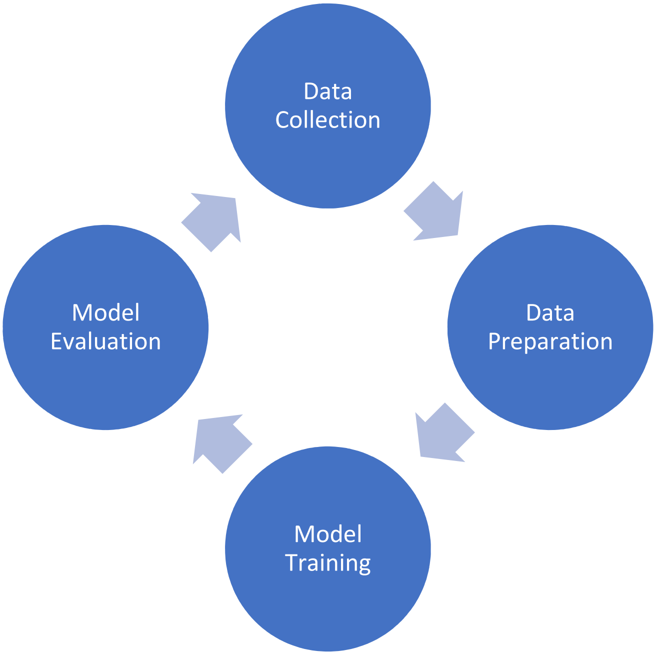
The first step in developing an ML model is obtaining a dataset in the application domain. The quality of the dataset plays a very crucial role in the performance of an ML model. Therefore, the correctness of the dataset must be ensured through proper evaluation of the experimental setup used to generate the dataset. The dataset is divided into training and testing data, which is used in model training and model evaluation steps, respectively.

Data preparation is the next step in building an ML model. Usually, the data available in a particular application domain does not work as-is for the problem at hand. Hence, preparing the data to cater to the needs of the problem is an essential step in ML workflow. Data preparation includes feature selection, feature extraction, sampling, normalisation, etc. (Kotsiantis et al. 2006).

An ML model is built by training an algorithm with past historical data. A part of the data called the training data is used to build the model. An ML model built can be a classification model, regression model, or a clustering model depending on the characteristics of the problem and data availability. During the process of model building, the parameters of the ML algorithms are fine-tuned to minimise the error in the model predictions.

The final step in the workflow of ML is model evaluation. One of the ways of model evaluation is the train-test split, which uses the testing data to determine the performance metrics. An alternative approach to evaluate a model is to perform cross-validation in which the dataset is divided into k subsets, and one out of k subset is used as testing data. The remaining k – 1 subset are used as training data. This process is performed k times with different subsets used as testing data during each iteration and is called k fold cross-validation.

Figure 1 Workflow of ML model (see online version for colours)



2.2 *Applications of machine learning*

ML finds applications in various domains in the real world. From industrial automation to healthcare, stock market analysis, to network management, ML has shown its ability to help applications in making informed decisions based on past data.

The following list shows some of the application domains of ML and its specific applications (Domingo, 2012):

- Healthcare: ML helps medical practitioners in predicting the ailments in patients based on the pattern developed from past patient data. This application domain of ML has gained momentum due to the availability of a massive amount of real-time data from wearable sensors and devices. The specific applications in the healthcare domain include blood pressure monitoring, drug discovery, kidney disease prediction and overall health monitoring, etc. (Panch, et al., 2018).
- Financial services: ML helps the financial sector to identify key insights of the historical data and also helps in predicting future trends in the trade. Stock market prediction, risk assessment, and fraud detection are few applications of ML in the finance sector (Mathur, 2019).
- E-commerce: ML can help e-commerce vendors in areas like recommender systems, warehouse monitoring, route prediction, cost, and demand prediction. ML solutions have demonstrated excellent results in several E-commerce companies like Amazon and Alibaba.
- Network management: Another domain that benefits from ML are the area of network management for applications like traffic analysis, malware analysis, anomaly detection, spam detection, etc. ML-based applications show improved network security, quality of service and, traffic management (Ayoubi et al., 2018).
- Manufacturing: From preventative maintenance to the automation of human tasks, ML has found its importance in the area of manufacturing and maintenance. ML also finds its application in product quality, business productivity, and customer relationship. The application of ML in the manufacturing sector helps the sector in eliminating unnecessary time delays and cost expenses (Wuest et al., 2016).

The application domains of ML are vast, and listing every such domain is avoided here for the sake of brevity. One of the applications in the domain of network management, i.e., network traffic analysis, is the topic of this paper. The network traffic analysis is a process of examining the network traffic to discover patterns in the traffic, which can help in applications like surveillance systems, QoS provisioning etc. One of the significant tasks in network traffic analysis is network traffic classification.

Traffic classification is a process of categorising a network's traffic into multiple classes. Traffic classification has gained importance in the last few years due to its application in network traffic management, network security, and research and development of networks (Pescape et al., 2018). In line with the growing popularity of ACNs, the classification of traffic in ACNs is an emerging and open field.

In the following subsection, we describe the need for traffic classification in ACNs and also the traditional and trending techniques employed for network traffic classification.

2.1.2 Traffic classification in (ACNs)

Traffic classification in ACNs is a challenging research area as these ACNs employ encryption techniques to preserve privacy. Discerning the traffic in ACNs can help in improving the performance of ACNs and also provides a solution to curb its illegal usage.

Traffic analysis and classification of ACNs can be helpful to internet users in multiple ways, as pointed out below:

- Classification of anonymous traffic against background traffic can be helpful in Network Surveillance Systems to block Anonymous traffic completely and enhance network security. Due to the inherent resistance that the ACNs put forth to internet censorship, traffic classification is helpful to the Law Enforcement Agencies (LEAs) in certain countries to enforce censorship to the internet.
- The designers of the ACNs can benefit from the classification of the anonymous traffic in robustifying the privacy provided to its users.
- Also, the classification of anonymous traffic and the identification of the application running on it (such as browsing, streaming, file transfer) can help the providers of ACNs to improve their performance by providing different QoS to different classes of traffic.

Over the years, traditional methods of traffic classification, like Port-based and Payload based methods are in use in network management applications. In the Port-based approach, the traffic is identified solely based on the port used by it; this is by far the most straightforward traffic classification technique. However, due to the usage of a dynamic range of ports and obfuscation techniques, port-based methods have become obsolete. The payload based method classifies traffic using DPI and is highly accurate. Nevertheless, payload based methods are sophisticated and resource-intensive operations. Recently, ML based methods have shown benefits in traffic classification and hence are suitable for traffic classification in ACNs. ML-based methods are light weighted, scalable, and suitable for real-time traffic classification as well.

The following section lists various ACNs available to internet users and also provides an overview of the architecture and working of each of the ACNs.

3 Overview of ACNs

ACNs are a subset of communication networks prevalently developed as a solution to protect the privacy of internet users. Although the internet was not originally designed to provide anonymity, the need for freedom of expression, censorship, protection against tracking, and surveillance paved the way for ACNs. The provision of anonymity on the Internet brings a few drawbacks along with it, such as abuse and illegal activities carried out through the anonymous network (Peng, 2014; Rigby, 1995).

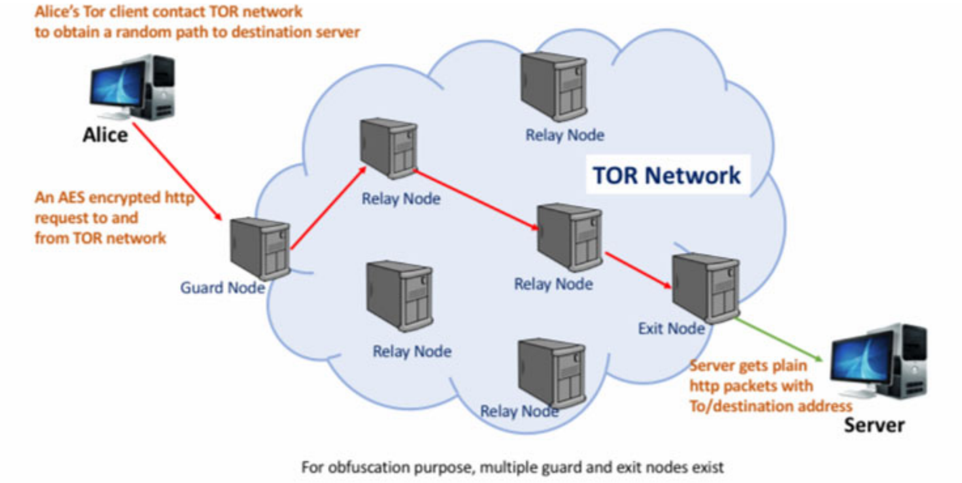
A few of the anonymity services available to protect the privacy of the users on the Internet are Tor, I2P, Jondonym, and Freenet. All these ACNs prevent IP tracking with the help of encryption techniques.

3.1 Tor anonymous network

Tor is a free anonymous network based on onion routing developed at the United States Naval research lab and works only for TCP streams. In networks employing onion routing, the messages are encrypted in layers similar to the layers of an onion with a different encryption key for each layer (Tor: Overview, 2020).

For Tor communication, a circuit consisting of relays is built for every hop in the communication path. Each relay node is aware of only the relay node that provides the encrypted data to it and the node to which it sends the encrypted data. Figure 2 shows a HyperText Transfer Protocol (HTTP) communication through the Tor network.

Figure 2 Tor anonymous network (see online version for colours)



In Figure 2, for the establishment of the circuit, Alice's Tor client obtains a list of available Tor entry/guard nodes from a directory server, and the Tor client has access to all the encryption keys of the relay nodes. The selected entry node sends the decrypted message to another relay node, and this process continues till the exit node, where each relay node decrypts the received message with its encryption key analogous to the peeling of layers of an onion. The exit node sends the message to the server without any encryption, and the response from the server reaches Alice's Tor client through the same nodes wherein each of the relay nodes encrypts the message. Since the client has all the encryption keys, it can decrypt the response from the server. The Tor network provides anonymity through its encryption technique, and any eavesdropper listening to a connection can only learn an encrypted message and information on relay nodes before and after it and not the actual message and its source and destination.

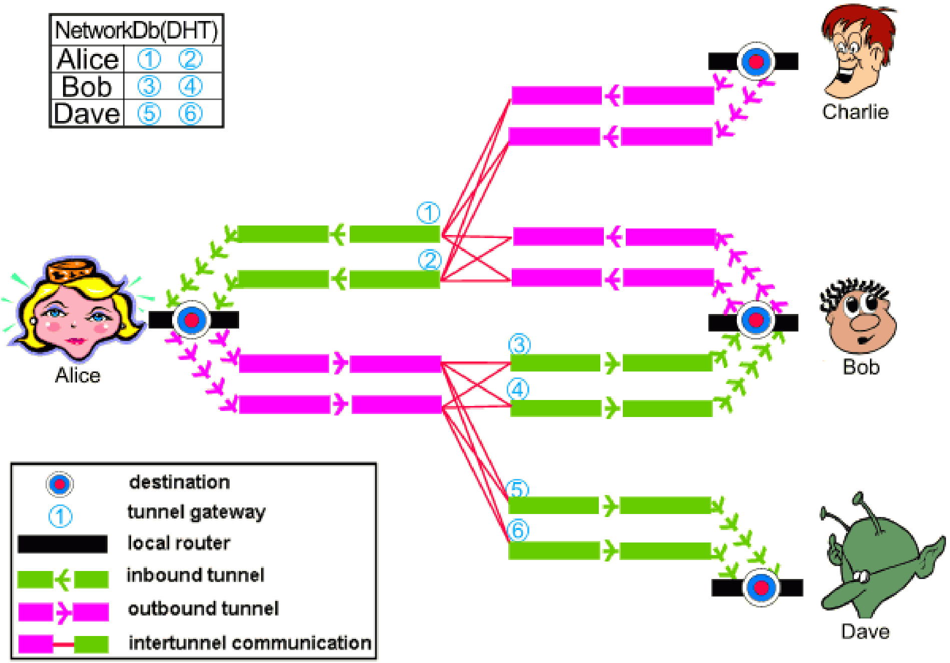
The Tor network supports applications such as Streaming, Torrent and Browsing, Chat, VoIP, etc. The Tor network prevents a censoring entity from blocking it with the help of pluggable transports (PTs). The PTs obfuscate encrypted traffic and thus bypassing the censoring entity. Some of the applications in PTs available on the Tor anonymous network are ScrambleSuit, Meek, FlashProxy, etc. (Tor: Pluggable Transport, 2020).

3.2 The I2P anonymous network

The I2P is a packet-based anonymous network with end to end encryption based on garlic routing. The garlic routing is an extended version of onion routing with support for encryption of multiple messages together (Invisible Internet Project, 2020).

The I2P network is composed of several nodes, which act like routers and a set of inbound and outbound tunnels to enable anonymous communication. The tunnels in the I2P network are made of peer nodes in the network and is always unidirectional. The I2P has a distributed network database to distribute routing and destination contact information. Figure 3 shows an example of I2P communication between a few users.

Figure 3 I2p anonymous network (see online version for colours)



In Figure 3, Alice, Bob, Charlie, and Dave are the users on the I2P network; each of them has a local router running within them to connect to other gateways on the network. In the example provided, Alice and Bob have both inbound and outbound tunnels, whereas the users' Charlie and Dave have only outbound and inbound tunnels, respectively. If Alice wishes to communicate with Bob, then she sends the message out of her outbound tunnel and towards the inbound tunnel of Bob through the tunnel gateway. The network database provides information about the tunnel and tunnel gateway to the users on the I2P network. The messages that pass through the tunnels and the gateway are garlic encrypted (I2P team, 2020).

Table 1 Applications supported on the I2P network

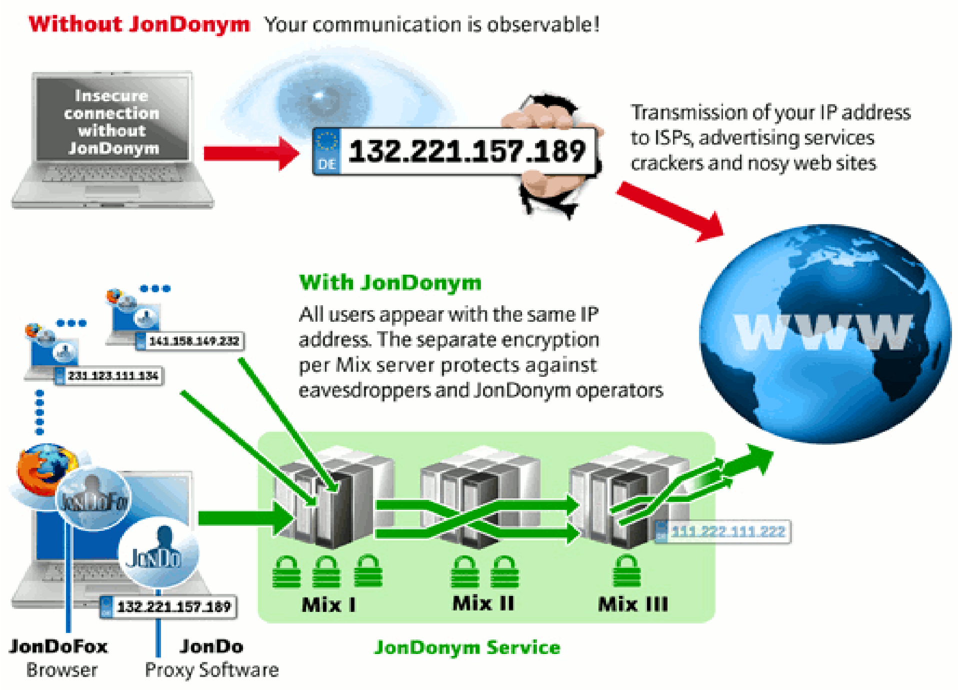
Application	*Functionality*	*Type*
I2P-Bote	Email	Plugin
I2PSnark	File sharing – BitTorrent client	Bundled
I2P Messenger	Instant messaging client	Standalone
jIRCii	Internet relay chat client	Plugin, standalone
Eepsites	Anonymous websites	Service
Putty	SSH client	Standalone

I2P supports a wide variety of applications for users seeking anonymity. Table 1 provides the details of the applications supported on the I2P network. Some of the applications supported on an I2P network are Email, File sharing, Anonymous clients, etc. The application of the I2P network can be bundled with the I2P browser or as a standalone or a plugin software. Some applications can also be used as a service in the I2P network (Supported Applications, 2020)

3.3 JonDonym anonymous network

Jondonym is a Java-based anonymity service that provides anonymity to its users through a set of mix cascades. The JonDonym users can select the mix cascades, and a cascade consists of two or three encrypted mix servers. The servers in the cascades are either operated by Public authorities, private companies, or by private persons. Figure 4 shows the working of the JonDonym network (JonDonym, 2020).

Figure 4 JonDonym anonymous network (see online version for colours)


Without JonDonym
Your communication is observable!
Insecure connection without JonDonym
DE
132.221.157.189
Transmission of your IP address to ISPs, advertising services crackers and nosy web sites
WWW
With JonDonym
All users appear with the same IP address. The separate encryption per Mix server protects against eavesdroppers and JonDonym operators
141.158.149.232
231.123.111.134
132.221.157.189
111.222.111.222
Mix I
Mix II
Mix III
JonDoFox
Browser
JonDo
Proxy Software
JonDonym Service

The operators of the cascade servers follow provisions to protect the anonymity of the users. As shown in Figure 4, users of the JonDonym network connect to its target through a series of mix servers that encrypt the connection, which helps in preserving the privacy of the users from the eavesdroppers.

Table 2 Summary of data generation methods and characteristics of the generated dataset

Ref.	*Technique*	*Publicly available*	*Dataset variants*	*Classes*	*Attributes*
AlSabah et al. (2012)	Public anonymous network (TOR)	No	Offline dataset (circuit level)	Browsing, BitTorrent, streaming	Circuit lifetime, amount of data sent upstream and downstream, variance of the cell IAT of a circuit.
			Online dataset (cell level)	Browsing, BitTorrent, streaming	IAT of the current cell and its mean, variance, exponential weighted moving average of cells sent on a circuit
He et al (2015)	Private anonymous network (TOR)	No	Only one variant	P2P, FTP, IM, web	Burst volumes and directions
Lashkari et al. (2017)	Public anonymous network (TOR)	Yes, available as ‘UNB-CIC dataset’	Scenario A	Tor, nonTor	Forward IAT (mean, min, max, std. deviation), backward IAT (mean, min, max, std. deviation), flow IAT (mean, min, max, std. deviation), active time (mean, min, max, std. deviation), Idle time (mean, min, max, std. deviation), flow bytes per second, flow packets per second, flow duration
			Scenario B	Browsing, audio streaming, chat, video streaming, mail, VoIP, P2P, file transfer	
Jia et al. (2017)	Private anonymous network (TOR)	No	Only one variant	Web pages, video, receive e-mail, send e-mail	Packet length entropy, 600-byte packet frequency, zero data packet frequency in the first 10 packets, average packet interval time.
Hodo et al. (2017)	Publicly available dataset (UNB-CIC dataset)	-	Only one variant	Browsing, audio streaming, chat, video streaming, mail, VoIP, P2P and file transfer.	Forward IAT (mean, min, max, std. deviation), backward IAT (mean, min, max, std. deviation), flow IAT (mean, min, max, std. deviation), active time (mean, min, max, std. deviation), Idle time (mean, min, max, std. deviation), flow bytes per second, flow packets per second, flow duration
Shahbar and Zincir-Heywood (2018)	Public anonymous network (TOR, I2P, JonDonym)	Yes, available as Anon17	Only one variant	Tor, I2P, JonDonym: each of the anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Flow direction (A/B), starting/ending timestamps, and duration of the flow, no. of bytes/packets Tx/Rx, mean, min, max, median, quartiles of packet length (PL) statistics, IAT statistics, joint PL-IAT statistics respectively, TCP and IP header related features, no. of connections 1 From source (destination) IP to different hosts 2 Between source and destination IP during the lifetime of the flow.
Pescape et al. (2018)	Publicly available dataset (Anon17)	-	Flow-based variant	Tor, I2P, JonDonym: Each of the Anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Flow direction (A/B), starting/ending timestamps, and duration of the flow, No. of bytes/packets Tx/Rx, mean, min, max, median, quartiles of Packet Length (PL) statistics, IAT statistics, Joint PL-IAT statistics respectively, TCP and IP header related features, No. of connections (i) from source (destination) IP to different hosts and (ii) between source and destination IP during the lifetime of the flow.
			Early variant	Tor, I2P, JonDonym: each of the Anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Sequence of pairs (payload length, IAT) of the first K packets of each flow

Table 2 Summary of data generation methods and characteristics of the generated dataset (continued)

Ref.	*Technique*	*Publicly available*	*Dataset variants*	*Classes*	*Attributes*
Kim and Anpalagan (2018)	Publicly available dataset (UNB-CIC Dataset)	-	Only one variant	Browsing, audio streaming, chat, video streaming, mail, VoIP, P2P and file transfer.	Mean, min, max, std. of forward IAT, backward IAT, flow IAT, active time, idle time, respectively. Flow bytes per second flow packets per second flow duration
Rao et al. (2018)	Emulation via experimentor	No	Only one variant	Tor, DNS, HTTP, SSH, SSL	The flow duration, the minimum/maximum/average/variance of uplink/downlink packet interval. The packet size: total uplink/downlink load, the total number of uplink/downlink packet, the total number of uplink/downlink packets with payload, the minimum/ maximum/average/variance size of upside/downside payload packet length distribution:
Cai et al. (2019)	Publicly available dataset (Anon17)	-	Only one variant	Tor, I2P, JonDonym: Each of the Anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Flow direction (A/B), starting/ending timestamps, and duration of the flow, no. of bytes/packets Tx/Rx, mean, min, max, median, quartiles of packet length (PL) statistics, IAT statistics, joint PL-IAT statistics respectively, TCP and IP header related features, no. of connections 1 From source (destination) IP to different hosts 2 Between source and destination IP during the lifetime of the flow.
Montieri et al. (2019)	Publicly available dataset (Anon17)	-	Flow-based variant	Tor, I2P, JonDonym: Each of the Anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Flow direction (A/B), starting/ending timestamps, and duration of the flow, no. of bytes/packets Tx/Rx, mean, min, max, median, quartiles of packet length (PL) statistics, IAT statistics, joint PL-IAT statistics respectively, TCP and IP header related features, no. of connections 1 From source (destination) IP to different hosts 2 Between source and destination IP during the lifetime of the flow.
			Early variant	Tor, I2P, JonDonym: Each of the anonymous network further divided into different traffic type and each traffic type subdivided into different application type	Sequence of pairs (payload length, IAT) of the first K packets of each flow

3.4 Freenet anonymous network

Freenet is a decentralised anonymity network for storing and publishing users' data without compromising privacy. It is a peer-to-peer network with a location-independent distributed file system. The main aim of this anonymity network is the provision of anonymity to both the producers and consumers of the data through the usage of hash keys (Watson et al., 2020).

4 Anonymous traffic classification approaches based on ML

In this section, we study and analyse different articles available in literature in the field of ACNs for traffic classification. This study provides an insight into the available literature by underlining the various aspects of an ML model such as dataset, feature engineering, ML model, and model evaluation. In addition, This section provides the researchers with a view of the current state of the art in the area of ML-based traffic classification of ACNs.

4.1 Dataset

Datasets are an integral part of the ML workflow. A dataset that is used in developing a learning model must be relevant to the application area, thus improving the model's performance. In the area of traffic classification, the collection of the dataset is a critical task due to the complex and scalable nature of the internet network.

Researchers have resorted to various ways to generate the dataset for developing a learning model for traffic classification such as

1 using emulation

2 using private anonymous networks

3 using real traffic from the real anonymous network like TOR, I2P

Another set of researchers use a dataset that is made publicly available through any of the methods as mentioned earlier. The schematic diagram of generating the datasets for traffic classification for ACNs is of the form shown in Figure 5.

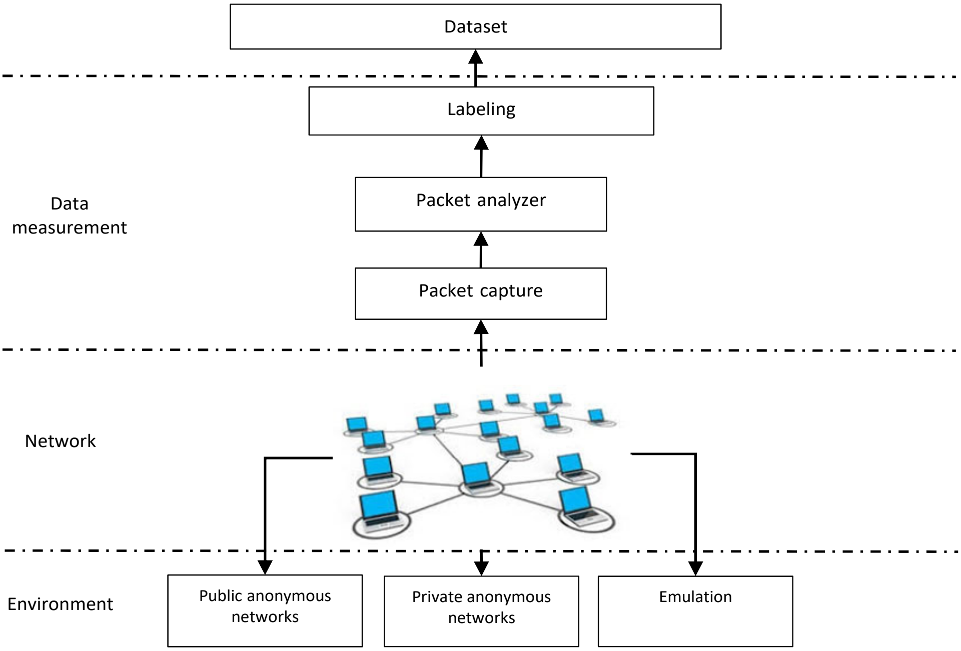
As shown in Figure 5, the dataset generation has three abstract layers, with Data measurement, Network, and Environment. The nodes in the network layer are set up depending on the type of environment used in the data generation. In the reviewed literature, we observe three types of environments: Public anonymous networks, Private anonymous networks, and the emulated network.

- *Public Anonymous networks*: Obtaining data from public anonymous networks is the optimal way to generate the dataset. However, due to the privacy-preserving nature of ACNs, data collection from real ACNs are often complicated. Despite the complexity, few of the works such as (AlSabah et al. 2012; Lashkari et al., 2017; Shahbar and Zincir-Heywood, 2018) have managed to obtain data from real ACNs, and out of these works, few (Lashkari et al., 2017; Shahbar and Zincir-Heywood, 2018) have also published the dataset publicly for the use of research community. Research works like Hodo et al. (2017), Pescape et al. (2018), Kim and Anpalagan

(2018) and Cai et al. (2019) have used the publicly available dataset for their research.

- *Private anonymous networks*: Another type of environment to generate data for the traffic classification in ACNs is private anonymous networks. In He et al. (2015) and Jia et al. (2017), a set of machines are set up to work together as a private Tor network. This method involves configuring a set of virtual machines to act as Tor nodes and directory servers. Although this method is not as accurate as real anonymous networks, it is still a feasible way of data generation for researchers.
- *Emulated networks*: Only one of the works (Rao et al., 2018) that we reviewed uses the technique of emulation via emulator software ExperimenTor (Bauer et al., 2011). ExperimenTor is a large scale Tor emulation testbed to help Tor researchers in setting up their system. The emulation method for generating data for traffic classification is the easiest of the methods available, but the results in an emulated network cannot be generalised for real networks, and to deploy the traffic classification technique, further evaluation on real networks is mandatory.

Figure 5 Schematic diagram of the dataset generation (see online version for colours)



For packet capture, all the works we reviewed use the Wireshark packet analyser (Orebaugh et al., 2006) to obtain the pcap files containing details of the traffic flows. The obtained pcap files are analysed by packet analysing software like Transanalyser2 (2020). Finally, the dataset is labeled based on its ground truth to produce a dataset.

Table 2 provides a summary of different ways in which the researchers have obtained the datasets for developing their learning model and the details on the dataset.

4.2 Data preparation

Preparing the dataset is necessary to improve the classification accuracy of the ML model. A few of the data preprocessing techniques that is required in an ML workflow are feature selection, sampling, missing values etc. While missing values are either filled with mean or zero values or in other situations, the instances with missing values are removed altogether from the dataset. To eliminate class imbalance problems in the dataset, down sampling, or up sampling to performed depending on the type of class imbalance. Feature selection is another pre-processing technique to select more relevant features among the available features in the dataset. Since the task of network traffic classification is a real-time task, considering feature selection as a pre-processing step can help to improve the training time of the model. Generally, A feature selection process involves an evaluation technique and a search technique to generate an optimal feature subset. A few of the commonly used feature selection evaluation methods are correlation-based selection and information gain. In addition to the feature selection evaluation algorithms, a search method is employed during pre-processing. Search methods like ranker and best first are most common, but some also use other methods, as shown in Table 3.

The reviewed works in the field of traffic classification of ACNs employ two types of feature selection evaluation methods: filter method and cluster-based methods.

- *Filter method*: The filter method ranks the features of the dataset based on metrics such as correlation coefficients and mutual information and is computationally fast. A majority of the reviewed works make use of correlation-based feature selection (Lashkari et al., 2017; Hodo et al., 2017; Pescape et al., 2018), few works such as Cai et al. (2019) and Jia et al. (2017) employ feature selection based on information gain, and others employ feature selection based on mutual information and other time constraints (Montieri et al., 2019).

  a *Correlation-based methods* figure out a subset of features based on the degree of redundancy in the feature set. The correlation method ensures that the elements in the feature subset have a high correlation with the target variable and low correlation amongst themselves (Khalid et al., 2014).

  b *Information gain* calculates the increase in entropy in the presence and absence of each feature in the feature set. This measurement of entropy decides the importance of a feature amongst the feature set and helps in selecting a subset of features (Khalid et al., 2014).

  c *Mutual information-based methods* figure out a subset of features based on the mutual information among variables, which is the uncertainty of one variable due to the knowledge of another variable. Often in mutual information-based methods, the objective is to minimise the mutual information among the feature vectors and maximise the mutual information between a feature vector and the target class variable (Hoque et al., 2014).

  d *Time constraint-based methods* are often used for real-time traffic classification in a network. The goal is to select features based on their occurrence in time to evaluate the minimum set of features that help in classifying a traffic flow in real-time. Only the reviewed works, (Pescape et al., 2018; Montieri et al., 2019) employ time constraint-based feature selection methods to perform real-time traffic classification in ACNs.

- *Cluster-based method*: One of the works that we reviewed (He et al., 2015) uses k means clustering for feature selection, which is a cluster-based method. In the cluster-based method, features are grouped using the k means algorithm with a pre-defined value of k. The elements in the feature set which do not belong to any cluster are irrelevant features, and the elements belonging to one cluster are redundant to each other. The centroids of created clusters are the elements of the subset of features used for building the ML model (Ismi et al., 2016).

In addition to the feature selection evaluation algorithms, some of the reviewed incorporate a search algorithm to evaluate a different possible subsets of features and obtaining the best subset of features.

- *Best first:* This search technique involves greedy hill-climbing aided with a backtracking algorithm.
- *Ranker*: In the Ranker search method, the features are listed as per the result obtained from the feature selection evaluation technique used in conjunction with it.
- *Heuristic search:* Heuristic search is based on probability, and this kind of search algorithm avoids evaluation of all the possibilities as in greedy search and calculates the approximate best subset.
- *Greedy search*: The greedy search method performs a forward or backward search on the features of the dataset. Greedy search stops when the addition/deletion of any remaining features results in a decrease in evaluation.

Table 3 Summary of feature selection techniques

Ref.	*Feature selection evaluation technique*	*Search method*
AlSabah et al. (2012)	-	-
He et al. (2015)	K-means clustering + ClustalW algorithm	-
Lashkari et al. (2017)	Correlation-based feature selection	Best first
	Information gain	Ranker
Jia et al. (2017)	Information gain	-
Hodo et al. (2017)	Correlation-based feature selection	Heuristic search
Shahbar and Zincir-Heywood (2018)	-	-
Pescape et al. (2018)	Correlation-based feature selection	Ranker
	Time constraint-based feature selection	-
Kim and Anpalagan (2018)	-	-
Rao et al. (2018)	-	-
Cai et al. (2019)	Modified mutual information and random forest (MMIRF)	Greedy search
Montieri et al. (2019)	Mutual information-based feature selection	Ranker
	Time constraint-based feature selection	-

4.3 ML algorithms and model evaluation

A lot of ML algorithms are readily available for building an ML model on various toolkits such as Weka (Holmes et al., 1994), and TensorFlow (Abadi et al., 2016). Due to the availability of a large number of ML algorithms, selecting the most suitable algorithm is very important to deploy the optimal ML solution in real-world traffic classification. All the works that we reviewed have built and evaluated models on multiple algorithms and selected the model with the highest performance. Table 4 shows the summary of the literature that we have reviewed concerning the ML algorithm used to build the model and the evaluation techniques employed to obtain its performance measure during traffic classification in ACNs.

As is evident from Table 4, all of the reviewed works involve a supervised learning model except for Rao et al. (2018), which uses the gravitational clustering algorithm. Traffic classification in communication networks is broadly divided into two types: offline classification and real-time classification.

- *Offline classification*: In offline classification, classification is performed by utilising the flow information after it has ended. This type of classification has no time constraints on the classifier model, and the majority of our reviewed works perform only offline classification Cai et al. 2019; He et al., 2015; Hodo et al., 2017; Jia et al., 2017; Kim and Anpalagan, 2018; Lashkari et al., 2017; Shahbar and Zincir-Heywood, 2018; Rao et al., 2018).
- *Real-time classification*: Real-time classification involves classifying network traffic into respective classes as early as possible. The on the fly classification of traffic is beneficial in network management tasks such as surveillance systems, QoS provisioning for applications. Real-time classification requires an efficient ML model and is a challenging task in network traffic classification. Few of our reviewed works, (AlSabah et al., 2012; Montieri et al., 2019; Pescape et al., 2018) perform real-time classification in addition to offline classification.

In ML, Train-test split and cross-validation methods are commonly used to validate classification models and evaluate their performance.

- *Train-test split*: In this validation method, the dataset is divided into a training set, testing set, and occasionally there is also a validation set. The training set is to fit the model, validation set to tune the hyper parameters, and test set to evaluate the performance of the built ML model. The researchers in He et al. (2015), Jia et al. (2017), Hodo et al. (2017), and Kim and Anpalagan (2018) incorporate this method for model validation.
- *Cross-validation*: The dataset is divided into k subsets, and one out of the k subset is used as testing data, and the remaining k – 1 subset is used as training data. This process is performed k times with different subsets used as testing data during each iteration and is called k fold cross-validation.

In AlSabah et al. (2012) and Lashkari et al. (2017), authors perform 10-fold cross-validation. Whereas in Pescape et al. (2018), Montieri et al. (2019), and Cai et al. (2019), stratified and nested variants of cross-validation are performed. In stratified, the subsets of data are created to ensure that each subset is representative of the whole set. On the other hand, in nested cross-validation, inner cross-validation is performed to fine-tune hyper parameters and to choose the best model and outer cross-validation to evaluate the performance of selected models.

Finally, in Table 4, we also provide the tool kits used by the reviewed works to provide the researchers with a view of the possible tools to be used for traffic classification in ACNs.

5 Conclusions, challenges and future trends

This review paper presented an overview of traffic classification in ACNs and a general procedure to perform ML-based traffic classification in ACNs. We discussed each step in the ML workflow in detail with a comparison of the relevant works in the traffic classification of ACNs. Finally, this study unveiled many of the challenges and future trends that should be incorporated by researchers in the ML-based traffic classification field to benefit from the usage of ACNs and also to eliminate the abuse of ACNs.

Given the current techniques and trend in the field of ML-based traffic classification in ACNs, we notice the following aspects still unaddressed and has the potential to be incorporated in future research works:

- *Availability of dataset*: Only two of the works (Lashkari et al., 2017; Shahbar and Zincir-Heywood, 2018) amongst the reviewed literature have made their dataset publicly available. The availability of datasets to researchers can help in easy comparison of results and lead to improved traffic classification models for ACNs.
- *Response time*: None of the works that we reviewed provide information on the response time of their model. As the ACNs are real-time, the response time of the model is a critical factor in real-time traffic classification.
- *Model deployment*: The integration of ML classifiers into ACNs is the end goal of the traffic classification research. None of the reviewed works have deployed their model on real networks to evaluate the performance in real-time.
- *Scalability*: Since the internet is enormous, evaluating the scalability of the traffic classification solution is a necessary task to deploy the model in real ACNs. Again, none of the works discussed focuses on this aspect of networks.

As discussed in Section 4 of the review paper, many research works have focused on discerning the traffic in ACNs using the ML approach. While the comparison tables in Section 4 showed that the available literature has employed varied techniques for data collection, data preparation and model building and evaluation. We notice that very few researchers made their dataset public for further research improvements due to the privacy-preserving nature of ACNs. Such public datasets can be beneficial in comparison of existing literature to quantitatively select the best traffic classification system for real-time deployment.

Another important area of network traffic classification in ACNs that needs to be worked upon by future researchers is the real-time or on the fly traffic classification, which provides more insights into real-time network management (QoS provisioning, network security) than compared to offline network traffic classification.

Table 4 Summary of ML algorithms and evaluation techniques

Ref.	*Type of algorithm*	*Algo*		*Evaluation*	*Evaluation results*	*Real-time classification*	*Tool kit*
AlSabah et al (2012)	Supervised	Offline classification	Bayes nets, LMT and FT	10-fold cross-validation	Accuracy: 91% (FT classifier) F-measure: 97% (Ft classifier)	x	Waikato environment for knowledge analysis (WEKA) software suite
		Online classification	Naive Bayes, Bayesian net	10-fold cross-validation	Accuracy: 97.8% (Bayesian net)	✓	
He et al. (2015)	Supervised	Profile HMM		Training set: 66.66% Testing set: 33.33%	Accuracy: 92% F-measure: > 90%	x	ClustalW and hmmer
Lashkari et al. (2017)	Supervised	Scenario A	ZeroR, C4.5, KNN	10-fold cross-validation	Precision: 99% (C4.5) Recall: 99.1% (C4.5)	x	WEKA
	Supervised	Scenario B	Random forest, C4.5, KNN	10-fold cross-validation	Precision: 84.2% (random forest) Recall: 84% (random forest)	✓	WEKA
Jia et al. (2017)	Supervised	TOR-IDT and TRI-TRAINING ALGORITHM		Training set: 75% Testing set: 25%	Recall: 96% accuracy: 94%	x	NL
Hodo et al. (2017)	Supervised	ANN (Levenberg-Marquardt training function) and SVM		Training set: 70 % Testing set: 15% Validate set: 20%	Accuracy: 99.8%	x	NL
Shahbar and Zincir-Heywood (2018)	Supervised	C4.5, naïve Bayes, random forest, and bayesian network		NL	Accuracy: 98.8% (random forest) F-measure: 100% (Bayesian network)	x	WEKA

Table 4 Summary of ML algorithms and evaluation techniques (continued)

Ref.	*Type of algorithm*	*Algo*	*Evaluation*	*Evaluation results*	*Real-time classification*	*Tool kit*
Pescape et al. (2018)	Supervised	Naïve Bayes, multinomial Naïve Bayes, Bayesian networks, C4.5, random forest	Stratified 10-fold cross-validation	Flow-based: accuracy and F-measure: 99.87% (random forest) Early classifier: Accuracy: 99.8% (naïve Bayes) F-measure: 99.78% (naïve Bayes)	✓	WEKA
Kim and Anpalagan (2018)	Supervised	1D CNN	Training set: 80% Testing set: 20%	Accuracy: 99.3%	x	TensorFlow
Rao et al (2018)	Unsupervised	GCA	Cluster validity	Accuracy: 80%	x	NL
Cai et al. (2019)	Supervised	Extreme gradient boosting (XGBoost)	Nested cross-validation scheme with an inner five-fold cross-validation and an outer Monte Carlo cross-validation	Accuracy: 98.52% F-measure: 99.9%	x	WEKA
Montieri et al. (2019)	Supervised	C4.5, random forest, naïve Bayes, Bayesian network	Stratified 10-fold cross-validation	Flow-based: accuracy and F-measure: 99.81% (random forest) Early classifier: Accuracy: 99.80% F-measure: 99.78% (naïve Bayes) and Scikit-Learn	✓	WEKA and Scikit Learn

Acknowledgements

The authors wish to thank Ramaiah institute technology (RIT), Bangalore and Visvesvaraya Technological University (VTU), Belgaum, for providing the facilities and support to carry out our research works.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. and Isard, M. (2016) 'TensorFlow: A System for Large-Scale Machine Learning', in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, Savannah, GA, USA.

Aceto, G. and Pescape, A. (2015) 'Internet censorship detection: a survey', *Computer Networks*, Vol. 83, pp.381–421.

AlSabah, M., Bauer, K. and Goldberg, I. (2012) 'Enhancing Tor's performance using real-time traffic classification', in *Proceedings of the 2012 ACM Conference on Computer and Communications Security.*

Ayoubi, N., Limam, S., Salahuddin, M.A., Shahriar, N., Boutaba, R., Estrada-Solano F. and Caicedo, O.M. (2018) 'Machine learning for cognitive network management', *IEEE Communications Magazine*, Vol. 56, No. 1, pp.158–165.

Bauer, K., Sherr, M. McCoy D. and Grunwald, D. (2011) 'ExperimenTor: a testbed for safe and realistic tor experimentation', in *Workshop on Cyber Security Experimentation and Test, CSET.*

Cai, Z., Jiang, B., Lu, Z., Liu, J. and Ma, P. (2019) 'Is Anon: flow-based anonymity network traffic identification using extreme gradient boosting', in *Proceedings of the International Joint Conference on Neural Networks*.

Dainotti, A. Pescape A. and Claffy, K.C. (2012) 'Issues and future directions in traffic classification', *IEEE Network*, Vol. 26, No. 1, pp.35–40.

Domingos, P. (2012) 'A few useful things to know about machine learning', *Communications of the ACM*, Vol. 55, No. 10, pp.78–87.

Finsterbusch, M., Richter, C., Rocha, E., Muller J-A. and Hanssgen, K. (2013) 'A survey of payload-based traffic classification approaches', *IEEE Communications Surveys and Tutorials*, Vol. 16, No. 2, pp.1135–1156.

He, G., Yang, M., Luo, J. and Gu, X. (2015) 'Inferring application type information from tor encrypted traffic', in *Proceedings – 2014 2nd International Conference on Advanced Cloud and Big Data, CBD 2014*.

Hodo, E., Bellekens, X., Iorkyase, E., Hamilton, A., Tachtatzis, C. and Atkinson, R. (2017) 'Machine learning approach for detection of nonTor traffic', in *Proceedings of the 12th International Conference on Availability, Reliability and Security.*

Holmes, G., Donkin A. and Witten, I.H. (1994) 'Weka: a machine learning workbench', in *Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference*, IEEE.

Hoque, N., Bhattacharyya, D.K. and Kalita, J.K. (2014) 'MIFS-ND: a mutual information-based feature selection method', *Expert Systems with Applications*, Vol. 41, No. 14, pp.6371–6385.

I2P team (2020) *A Gentle Introduction to How I2P Works* [online] https://geti2p.net/en/docs/how/intro (accessed February 2020).

Ismi, D.P., Panchoo, S. and Murinto, M. (2016) 'K-means clustering based filter feature selection on high dimensional data', *International Journal of Advances in Intelligent Informatics*, Vol. 2, No. 1, pp.38–45.

Jia, L., Liu, Y., Wang, B., Liu, H. and Xin, G. (2017) 'A hierarchical classification approach for tor anonymous traffic', in *2017 9th IEEE International Conference on Communication Software and Networks, ICCSN 2017*.

JonDonym [online] https://anonymous-proxy-servers.net/ (accessed 2020).

Khalid, S., Khalil, T. and Nasreen, S. (2014) 'A survey of feature selection and feature extraction techniques in machine learning', in *2014 Science and Information Conference*, IEEE, London.

Kim, M. and Anpalagan, A. (2018) 'Tor traffic classification from raw packet header using convolutional neural network', in *1st IEEE International Conference on Knowledge Innovation and Invention, ICKII 2018*.

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P. (2006) 'Data preprocessing for supervised leaning', *International Journal of Computer Science*, Vol. 1, No. 2, pp.111–117.

Lashkari, A.H., Gil, G.D., Mamun, M.S.I. and Ghorbani, A.A. (2017) 'Characterization of tor traffic using time based features', in *ICISSP 2017 – Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, January.

Mathur, P. (2019) 'Overview of machine learning in finance', in *Machine Learning Applications Using Python*, pp.259–270, Springer.

Mitchell, T.M. (1997) *Machine Learning*, McGraw-Hill, New York.

Montieri, A., Ciuonzo, D., Bovenzi, G., Persico, V. and Pescapé, A. (2019) 'A dive into the dark web: hierarchical traffic classification of anonymity tools', *IEEE Transactions on Network Science and Engineering*, Vol. 7, No. 3, pp.1043–1054.

Orebaugh, A., Ramirez, G., Burke, J., Pesce, L., Wright, J. and Morris, G. (2006) 'Chapter 4 – using wireshark', in *Wireshark & Ethereal Network Protocol Analyzer Toolkit*, Rockland, Syngress, pp.133–220.

Panch, T., Szolovits, P. and Atun, R. (2018) 'Artificial intelligence, machine learning and health systems', Journal of Global Health, Vol. 8, No. 2, pp.1–8.

Peng, K. (2014) *Anonymous Communication Networks: Protecting Privacy on the Web*, pp.1–10, Auerbach Publications, USA.

Pescape, A., Montieri, A., Aceto G. and Ciuonzo, D. (2018) 'Anonymity services tor, I2P, JonDonym: classifying in the dark (web)', *IEEE Transactions on Dependable and Secure Computing*, Vol. 17, No. 3, pp.662–675.

Rao, Z., Niu, W., Zhang, X.S. and Li, H. (2018) 'Tor anonymous traffic identification based on gravitational clustering', *Peer-to-Peer Networking and Applications*, Vol. 11, No. 3, pp.592–601.

Rigby, K. (1995) 'Anonymity on the internet must be protected', *Ethics and Law on the Electronic Frontier*.

Shahbar, K. and Zincir-Heywood, A.N. (2018) 'How far can we push flow analysis to identify encrypted anonymity network traffic?', *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World*, *NOMS 2018*, pp.1–6.

Supported Applications (2020) [online] https://geti2p.net/en/docs/applications/supported, (accessed February 2020).

The Invisible Internet Project (2020) [online] https://geti2p.net/en/ (accessed February 2020).

Tor: Overview (2019) [online] https://.www.torproject.org/about/overview.html.en (accessed February 2020).

Tor: Pluggable Transports (2019) [online] https://.www.torproject.org/docs/pluggable-transports.html.en (accessed February 2020).

Tranalyzer2 (2020) [online] http://tranalyzer.com (accessed February 2020).

Wang, M., Cui, Y., Wang, X., Xiao, S. and Jiang, J. (2017) 'Machine learning for networking: workflow, advances and opportunities', *IEEE Network*, Vol. 32, No. 2, pp.92–99.

Watson, R., Chillingsworth, B., Tarbett, S., Rafferty G. and Carter, G. (2020) *Freenet* [online] http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p7.html (accessed February 2020).

Wuest, T., Weimer, D., Irgens, C. and Thoben, K-D. (2016) 'Machine learning in manufacturing: advantages, challenges, and applications', *Production & Manufacturing Research*, Vol. 4, No. 1, pp.23–45.

Zuev, D. and Moore, A.W. (2005) 'Traffic classification using a statistical approach', *International Workshop on Passive and Active Network Measurement*, Berlin.