Anomaly Detection using Network Metadata

Khaled Mutmbak[1], Sultan Alotaibi[2], Khalid Alharbi[3], Umar Albalawi[4], Osama Younes[5]

College of Computing and Information Technology, University of Tabuk, Tabuk, 71491, Saudi Arabia[1,2,3,4,5]
Faculty of Computers and Information, Menoufia University, Menoufia 32951, Egypt[5]

***Abstract*—The proliferation of numerous network function today gave rise to the importance of network traffic classification against various cyber-attacks. Automatic training with a huge number of representative data necessitates the creation of a model for an efficient classifier. As a result, automatic categorization requires using training techniques capable of assigning classes to data objects based on the activities supplied to learn classes. Predefined classes allow for the detection of new items. However, the analysis and categorization of data activity in intrusion detection systems are vulnerable to a wide range of threats. Thus, New methods of analysis must be developed in order to establish an appropriate approach for monitoring circulating traffic in order to solve this problem. The major goal of this research is to develop and verify a heterogeneous traffic classifier that can classify the collected metadata of networks. In this study, a new model is proposed, which is based on machine learning technique, to increase the accuracy of prediction. Prior to the analysis stage, the gathered traffic is subjected to preprocessing. This paper aims to provide the mathematical validation of a novel machine learning classifier for heterogeneous traffic and anomaly detection.**

***Keywords*—Anomaly detection; network metadata; packet analysis; intrusion detection system; machine learning; classification; heterogeneous traffic**

I. Introduction

As part of network forensics, network traffic and event logs are commonly referred to as being sniffed, recorded, acquired, and analyzed to investigate a network security incident. It enables the investigator to study network traffic and records to identify and locate the assaulting system. Computers, smartphones, tablets, and other network-connected devices continue to grow. As the frequency of assaults against networked systems increases, the criticality of network forensics grows. Most previous studies confront two fundamental issues in extracting external and internal data, making traffic flow prediction a difficult endeavour. Currently, available solutions do not completely use the fundamental properties of short-term nearby and long-term periodic temporal patterns in terms of their various roles. In terms of the extrinsic task, current work has primarily used hand-crafted fusion algorithms to incorporate external inputs, however, there are still challenges with generalization [1].

The examination of a traffic incident is divided into two stages: Appearance check is the initial stage in the process of determining the Bloom filter (period) including an excerpt. To find the flows that conveyed the excerpt, the second phase is termed "flow determination," and it involves combining the excerpt blocks with the flows found by the Bloom filter. It was key difficulties handled by HBF [2], such as ensuring that blocks were aligned and that they were consecutively placed.

Cybercrime is a constant danger to computer networks. No security mechanism can guarantee complete safety. Even the most advanced network security measures are unable to identify and prevent all assaults, particularly those that are new and unknown. In certain circumstances, preventing cybercrime is impossible. Suppose that confidential information about a company is leaked over its network. How can security specialists track down cybercriminals? Let us consider the following scenario: an organization's internal network has been infected by a worm, and the organization's Intrusion Detection and Prevention System (IDS/IPS) was unable to identify and block the worm's dissemination. How can you track down the person who spreads the virus or the afflicted systems? As a result, in addition to preventative security systems, tools and methodologies for investigating cybercrime after it has occurred are required. This is the function of network forensics and the tools that it provides [2].

Recording and storing raw network traffic is the most basic method of network research. Traffic recording makes it feasible to examine any networking event that occurs. It is possible to scan through the recorded traffic for the leaked information or the worm's signature to determine where it originated and where it ended up. "Attribution" is the term for this operation. The most difficult challenge with this system is the exceedingly costly storing of large amounts of data [3]. In addition, the invasion of privacy is a concern with traffic recording. By monitoring network traffic, it is possible to gain access to the personal information of users. As a result of the increasing difficulty in providing both privacy and network forensics, new Internet designs and protocols have been proposed [4]. However, implementing such modifications would be prohibitively expensive, making them impractical in practice.

In the field of traffic categorization, three groups of methodologies exist port-based, payload-based, and machine learning-based methods [5]. The identification of network traffic based on port numbers is a straightforward process that depends on mapping programs to well-known port numbers. Regrettably, port-based categorization algorithms have grown erroneous as a result of the increased use of dynamic port numbers by numerous apps. Payload-based approaches necessitate the analysis of the payload of each packet. Privacy regulations and encryption, on the other hand, may prevent traffic payloads from being accessed. As a result of this, deep packet inspection (DPI) is expensive in terms of both computation and signature maintenance [6].

II. Motivation

Machine learning-based solutions have the potential to overcome some of the restrictions associated with port- and payload-based systems. More precisely, machine learning approaches can classify Internet traffic based on application-neutral traffic data. When it comes to how long it takes to send and receive a particular message, there are several variables that may be taken into consideration. Furthermore, it has the potential to minimize computing costs while also making it easier to identify encrypted traffic.

There are two main applications for network forensics. The first, which focuses on network security, is keeping an eye out for unusual traffic patterns and spotting breaches. On a hacked system, an attacker may be able to delete all log files. Consequently, network-based evidence may be the sole evidence accessible for forensic investigation in this situation. Law enforcement can also take advantage of network forensics by interpreting human communication represented through e-mails or other forms of electronic correspondence and reassembling transmitted information, looking for keywords, and so on [7].

Today's world is evolving at a rapid pace, and the internet is critical for quicker communication between people or machines, faster transactions, and faster fulfillment of duties (tasks). However, the internet is also a major victim of cybercrime. Transactions over the internet are the main draw for attackers. To do this, we need a forensic technology known as "Network Metadata" to help us identify the perpetrators of cybercrime and their methods of attack. Network Metadata is a sub-field of digital forensics research that deals with computer networks. The collection of network traces from the victim system for examination is a common practice in network forensics, whether the crime has been discovered or after it has been committed. The evidence gathered can be used to bring the perpetrator to justice in a criminal court of law. While digital forensics involves the examination of static data, network forensics involves the examination of volatile and dynamic data [8].

III. Related Work

The study [9] establishes a network intrusion criminal system based on the switching scheme (NIFSTC) that may detect criminality in networked situations and identify digital evidence automatically. The advantage of NIFSTC is that it does not require a standard forensic network to be built, hence it has superior detection performance in practice than traditional approaches. For the most modern network forensic methodologies, the KDD Cup Experiment Series 1999 dataset shows NIFSTC's highest true positive (TP) and lowest rate false positive (FP) .

The authors [10] introduced SPIE (Source Path Isolation Engine) in this regard, which calculates the first eight bytes of the payload and packet digests (i.e. hashes) from the header. A brief period of time is allowed for the digestion of these digests in a bloom filter. If a third-party device, such as an IDS or a firewall, identifies suspicious activity, SPIE can be used to track down the source of a packet.

In the research [11], the focus is on the security risks of the botnet through which DDoS attacks, worms and spam attacks are implemented. For network security forensic investigation, the researchers recommended the design and implementation of a cloud-based security center. Also, cloud storage is used to store the acquired traffic data, which is then processed utilizing cloud computing.

A tool that explores the architecture of the network forensic is proposed in [12], which is called NetFo (Network Forensic) analysis tool. It captures packets using Winpcap technology and It can be used as a monitoring and management tool. NetFo can discover session information, keywords, bookmarks, hostnames, IP addresses, and other information.

As explained in [13], due to many requirements that were not addressed in this design space, developing a forensic network architecture is a complex task.

The authors [14] present a real-life case study in which they reconstruct a crime scene in relation to a victim's previous Facebook session using digital evidence collected and analysed via access to a desktop computer's RAM, with a focus on some distinct chains that could be used to reconstruct a previous Facebook session.

Huaxin et al. [15] developed a framework for extracting four types of characteristics from real-world Wi-Fi data, as well as supervised machine learning approaches for estimating user demographics. The study was based on Wi-Fi traffic information from 28,158 users during a five-month period. According to the testing results, the best accuracy in predicting gender and education level is 82% and 78%, respectively. Users' demographics may be predicted with a precision of 69% and 76% utilizing HTTPS traffic, even in encrypted transmission (i.e., across the internet). Being forensically prepared increases the degree of security in both cloud and on-premises computing. As a result, research in the fields of cloud and network security may also apply to IoT-centered forensics investigations. After all, traditional computer networks and Internet of Things (IoT) networks are also vulnerable to security flaws. Because IoT systems interact with the physical environment more frequently than traditional systems, they are susceptible to a greater number of physical and digital dangers. As a result, the work introduced in [16] was dedicated for securing the IoT domain.

The authors in [17] provided an overview of forensic advancements related to the IoT as well as the remaining hurdles. They focused on the taxonomy and criteria in the IoT Forensics. However, they did not discuss historical and current frameworks, standardization and certification difficulties in the IoT Forensics.

IV. The Proposed Model

The ML models are becoming widespread in recent years because of mitigating a variety of complex relationships and acquiring the most favorable solutions by general evolution. The ML models have the ability to discover nonlinear relationships and complex functions among independent and dependent variables based on processing and classifying the data through training. An ML technique is comprised of algorithms with many models based on artificial intelligence. In this paper, five ML classifiers are used and compared in terms of the highest accuracy.

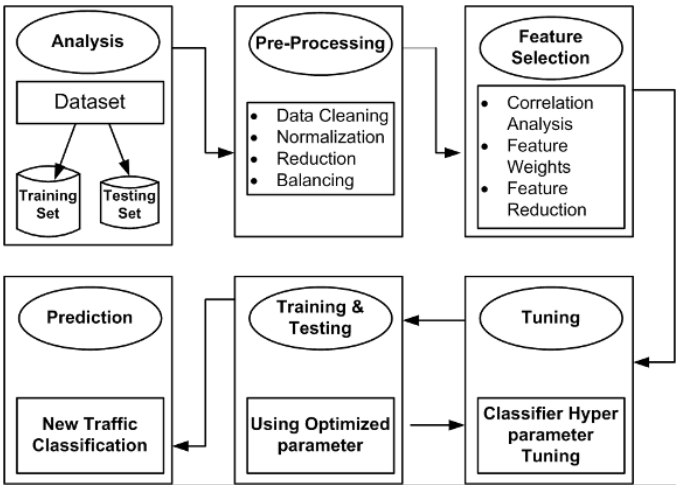

Fig. 1. The Proposed Model based on Machine Learning

Fig. 1 depicts the overall layout of the proposed framework based on machine learning approaches for network anomaly detection. It represents the phases that the model goes through and includes a large number of distinct processes. In the first phase, the dataset is analyzed and split into training and testing sets. For both training and testing, the attribute vectors are sliced in a 70:30 ratio. Next, in the pre-processing phase, the dataset is cleaned, features containing categorical data are normalized, and records including incorrect data are removed. Following, in the feature selection phase, the features are analyzed according to their weights and we choose most important features to define the attacks. Next, in the tuning phase, parameters of chosen classifier are tuned and optimized using a grid search. At the end, the optimized classifier is used for training and testing datasets, which are used for prediction of new traffic records.

A. Dataset

Data are the most valuable asset to develop an efficient intrusion detection system. CICIDS2017 [18] is the most recent intrusion detection evaluation dataset. It was created by the Canadian Institute for Cybersecurity at the University of New Brunswick. The CICIDS2017 dataset was constructed using the Network Traffic Flow analyzer. It was captured over a duration of 5 days over which 83 features and 15 classes were captured [19]. One of these classes represents the normal network traffic (defined as Benign) while the other 14 represent anomaly traffic (called Attacks). The names and numbers of these classes are shown in Table I. Compared to older and traditional datasets, such as KDD-99 [20], DARPA 98/99 [21] and ISCX2012 [22], CICIDS2017 dataset has the following advantages:

- Cover the current trends of attacks.
- Represent real-world data.
- Datasets are labelled.
- Attacks based on many protocols are included, such as HTTP, HTTPS, FTP, SSH, and email protocols.

For these reasons, the CICIDS2017 dataset is selected.

TABLE I. NUMBER OF STREAM RECORDS FOR ATTACK TYPES IN THE CICIDS2017 DATASET

Attack Type	Records
BENIGN	2359087
DoS Hulk	231072
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – Sql Injection	21
Heartbleed	11

B. Data Pre-Processing

As explained in the last section, the CICIDS2017 dataset contains 3119345 stream records and 83 features containing 15 class labels (one for normal traffic and 14 for attacks). To ensure that the dataset is ready to be trained, we need to clean and normalize it.

As most of the datasets, CICIDS2017 dataset contains some undesirable elements that must be removed. In CICIDS2017 dataset, because the network traffic was collected using the CICFlowMeter tool, some flag features have constant values (0 or 1), such as “Bwd URG Flags” and “Bwd P SH Flags”. These features were removed from the dataset because they have no impact on model results and to decrease the memory footprint of the dataset. Next step in the preprocessing phase is removing records that have missing class label, missing information, and invalid values such as “NaN” or “Inf”. After examining these records, 288602 records were removed.

If the dataset used for training of a classifier or detector suffers from high class imbalance problem, the classifier biases towards the majority class. As a result, the classifier shows lower accuracy with higher false alarm. Unfortunately, CICIDS2017 data set is prone to high class imbalance, as shown in Table 1. Therefore, to avoid this problem, the normal traffic records have been down sampled. In addition, to improve prevalence ratio and reducing class imbalance issue, few minority classes have been merged, such as Web Attacks. Therefore, the new dataset was partitioned into 70% for training (1571510 records) and 30% for testing (471453) sets.

C. Feature Selection

The goals of feature selection are to identify and remove unneeded, irrelevant and redundant features from the dataset. This help reduce the complexity of the predictive model without compromising its accuracy. Feature selection helps define most important features for detecting attacks on the dataset. First using correlation test, some features are removed from the dataset to reduce its size and enhance the performance.

Fig. 2 shows the correlation matrix, which shows the value of the correlation of variables and features with each other, which negatively or positively affects them. A correlation

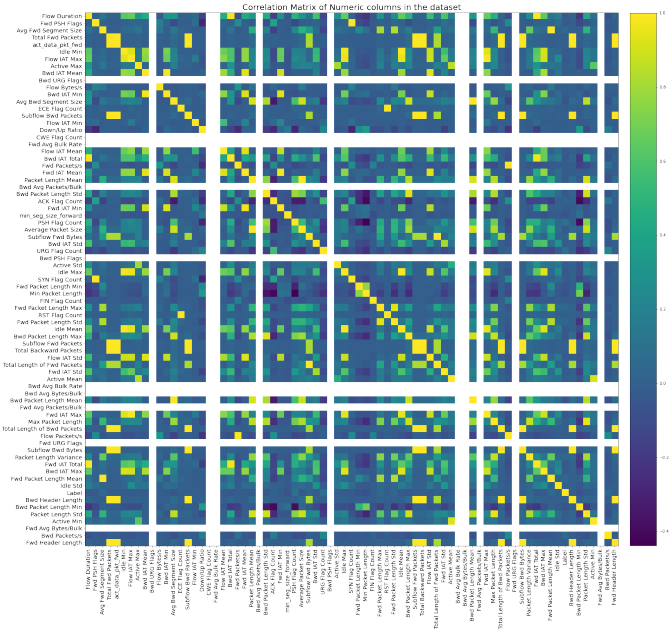

Fig. 2. Correlation Matrix

matrix is simply a table that displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and identify and visualize patterns in the given data. Each cell in the table contains the correlation coefficient between the features that scales from 0 to 1. If the coefficient approaches 1, it means that it is more positive, meaning that both features have an impact on the prediction process, and whenever the value approaches 0, it means a negative correlation that does not benefit us in the process of prediction, and they have no effect.

By analyzing the correlation matrix, we found a strong correlation between the following features: (Fwd IAT Std, BwdIATMean), (Bwd Header Length, Fwd Header Length) and (Bwd Header Length, Subflow Fwd packets). Therefore, we delete the features that are not needed.

After removing correlated features, we still have large number of features. We need to use feature selection methods to determine the importance of a certain features in the detection of anomalous traffic. There are several feature selection methods in the literature, such as Fisher Score, T-Score, chi-squared tests, random forest, or regression. Using these five feature selection methods, each feature is given a weight of importance as to how useful they are. These weights of features are compared and sorted. Fig. 3 shows the most 10 important features that are used for training and testing in the proposed model.

V. Result and Discussion

In this section, the results will be presented and discussed based on the proposed machine learning techniques.

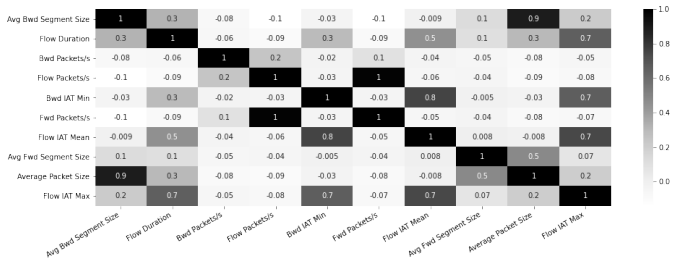

Fig. 3. Important Features

A. Data Classification Methods

In artificial intelligence, machine learning is regarded as a subfield. Automatic classification [[23], [24]] is one of interested subjects for machine learning. In order to handle classification difficulties, automatic learning employs a variety of methods that group homogenous classes of comparable data items together. In order to train the decision rule and develop a classifier, supervised learning is adopted. ML can be used to create a predictive model to detect unknown attacks in network traffic. However, one important problem in ML is to identify and select the most relevant feature characteristics, from which to build a specific model based on training data for a particular classification job [[24], [25],[26]].

Classification is a logical choice for doing predictions with discrete known outcomes when using a machine learning technique such as classification. Items are classified using a classification technique, which is a set of exact rules for categorizing objects based on the quantitative and qualitative factors that characterize the objects. There are a variety of goals for which data categorization is performed, the most prevalent of which is to assist with data security challenges, particularly in anomaly detection [[27], [28], [29]].

In this work, we adopted using five classifiers to categorize the network, which are: the Random Forest, Logistic Regression, Decision Tree Algorithm, SVM, and the k-nearest neighbors. The findings were then compared using performance metrics and classification reports. Through the optimization of the classifier, training and testing process are repeated, where the behavior of the classifier is changed until the intended behavior is accomplished.

B. Performance Measures

To evaluate the performance of the suggested classification methods for anomaly detection, we adopted the following measures: accuracy, recall, precision, and F1 Score. The confusion matrix is utilized to separate the prescient execution of the classification in the test data.

Fig. 4 shows a template for a binary confusion matrix that uses the four kinds of results: (true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN)) along with the positive and negative classifications. The following measurement metrics are used to measure the performance of a dataset:

1) Accuracy calculates predicted observation ratios for

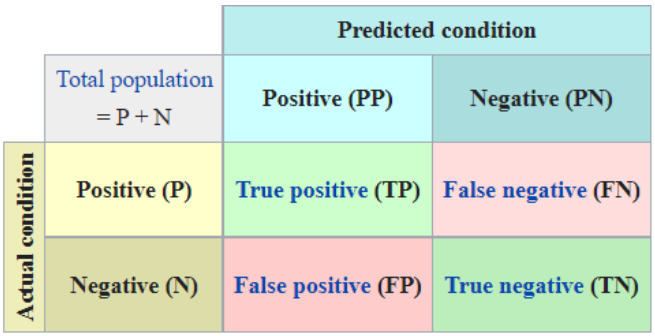

Fig. 4. Confusion Matrix

TABLE II. RESULT OF COMPARE BETWEEN CLASSIFIERS

Algorithm	Accuracy	Recall	Precision	F1 Score
K-Neighbors	95.84	97.51	95.84	97.84
Logistic regression	96.51	97.95	98.45	98.20
SVM	93.69	96.43	96.84	96.63
Decision Tree	97.11	98.31	98.69	98.50
Random Forest	98.63	98.82	99.80	99.31

total observations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2) Recall is the ability of the proposed model to detect the attacks. Recall can be calculated from the number of detected attacks rather than the number of actual attacks.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

3) Precision is the ratio of predicted positive to total positive observed predictions.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

4) F1 Score is the average of recall and precision values.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

C. Experimental Results

Table II and Fig. 5 show the results of applying five different machine learning techniques for classifying different types of attacks. Fig. 5 shows the confusion matrix for all algorithms. Table II shows the accuracy, precision, recall, F1 Score for each algorithm.

From the Table II, we can notice that the best algorithms are Random Forest and Decision Tree. This is because they have a high accuracy and precision rates. The worst algorithm is K-Neighbors because it had lower accuracy and precision rates.

VI. CONCLUSION AND FUTURE WORK

An intrusion detection system is an important protection tool for detecting complex network attacks. In this work we have developed a new model for network intrusion (anomaly) detection based on machine learning algorithms. The proposed

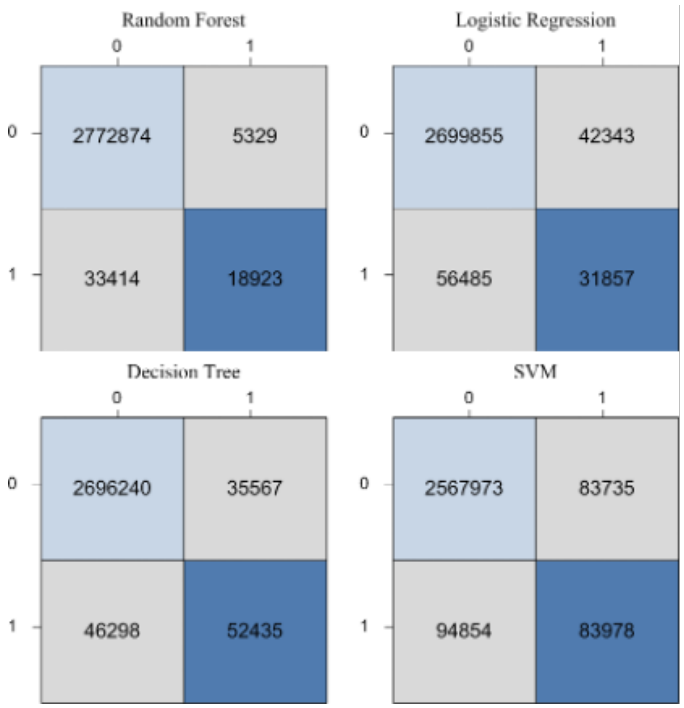

Fig. 5. Confusion Matrix for different Classification Methods

model consists of six phases: dataset analysis, pre-processing, feature selection, parameter tuning, training and testing. Using the proposed model, five machine learning algorithms have been investigated for classification of network anomaly detection, which are: K-neighbors, logistic regression, SVM, decision tree and random forest. The performances of these ML algorithms have been observed on the basis of their accuracy, recall, precision and F1 score. The dataset CICIDS-2017 has been used for training and testing, which consists of seven different types of attacks. According to results, compared to other ML algorithms, the performance of the random forest algorithm is better. This is because it has achieved the highest accuracy and precision rates for classification of anomaly detection, which are 98.63% and 99.80, respectively. Compared to related work, the performance of the proposed model is better. This is because of: (1) The dataset was carefully cleaned by removing noise and outlier data and solving imbalance issues. (2) The proposed feature selection technique removed correlated and irrelevant features from the dataset. (3) Parameters of chosen classifier are tuned and optimized using gride search. As a future work, we will investigate other machine learning and deep learning algorithms for network anomaly detection.

ACKNOWLEDGMENT

The authors would like to thank The University of Tabuk for providing research support and facilities.

REFERENCES

[1] S. Fang, X. Pan, S. Xiang, and C. Pan, "Meta-msnet: Meta-learning based multi-source data fusion for traffic flow prediction," *IEEE Signal Processing Letters*, vol. 28, pp. 6–10, 2020.

[2] S. M. Hosseini, A. H. Jahangir, and M. Kazemi, "Digesting network traffic for forensic investigation using digital signal processing techniques," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3312–3321, 2019.

[3] D. Quick and K.-K. R. Choo, "Impacts of increasing volume of digital forensic data: A survey and future research challenges," *Digital Investigation*, vol. 11, no. 4, pp. 273–294, 2014.

[4] M. Afanasyev, T. Kohno, J. Ma, N. Murphy, S. Savage, A. C. Snoeren, and G. M. Voelker, "Privacy-preserving network forensics," *Communications of the ACM*, vol. 54, no. 5, pp. 78–87, 2011.

[5] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE communications surveys & tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[6] W. Li and A. W. Moore, "A machine learning approach for efficient traffic classification," in *2007 15th International symposium on modeling, analysis, and simulation of computer and telecommunication systems*. IEEE, 2007, pp. 310–317.

[7] R. Hunt and S. Zeadally, "Network forensics: an analysis of techniques, tools, and trends," *Computer*, vol. 45, no. 12, pp. 36–43, 2012.

[8] G. Shrivastava, "Approaches of network forensic model for investigation," *International Journal of Forensic Engineering*, vol. 3, no. 3, pp. 195–215, 2017.

[9] Z. Tian, W. Jiang, and Y. Li, "A transductive scheme based inference techniques for network forensic analysis," *China Communications*, vol. 12, no. 2, pp. 167–176, 2015.

[10] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-based ip traceback," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 3–14, 2001.

[11] Z. Chen, F. Han, J. Cao, X. Jiang, and S. Chen, "Cloud computing-based forensic analysis for collaborative network security management system," *Tsinghua science and technology*, vol. 18, no. 1, pp. 40–50, 2013.

[12] M. H. Mate and S. R. Kapse, "Network forensic tool–concept and architecture," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, 2015, pp. 711–713.

[13] G. Shrivastava, "Network forensics: Methodical literature review," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2016, pp. 2203–2208.

[14] H.-C. Chu, D.-J. Deng, and J. H. Park, "Live data mining concerning social networking forensics based on a facebook session through aggregation of social data," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1368–1376, 2011.

[15] H. Li, H. Zhu, and D. Ma, "Demographic information inference through meta-data analysis of wi-fi traffic," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1033–1047, 2017.

[16] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[17] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[19] World BankThe World Bank, "Intrusion detection evaluation dataset (cicids2017)," 2018, accessed April 2022, http://www.unb.ca/cic/datasets/ids-2017.html.

[20] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[21] J. W. Haines, R. P. Lippmann, D. J. Fried, M. Zissman, and E. Tran, "1999 darpa intrusion detection evaluation: Design and procedures," MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, Tech. Rep., 2001.

[22] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[23] L. Igual and S. Seguí, "Introduction to data science," in *Introduction to Data Science*. Springer, 2017, pp. 1–4.

[24] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159–190, 2006.

[25] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, 1999.

[26] M. Rocha, P. Cortez, and J. Neves, "Evolution of neural networks for classification and regression," *Neurocomputing*, vol. 70, no. 16-18, pp. 2809–2816, 2007.

[27] S. Hao, J. Long, and Y. Yang, "Bl-ids: Detecting web attacks using bi-lstm model based on deep learning," in *International Conference on Security and Privacy in New Computing Environments*. Springer, 2019, pp. 551–563.

[28] A. Verma, A. Singh *et al.*, "An approach to detect packets using packet sniffing," *International Journal of Computer Science and Engineering Survey*, vol. 4, no. 3, p. 21, 2013.

[29] M. Idhammad, K. Afdel, and M. Belouch, "Detection system of http ddos attacks in a cloud environment based on information theoretic entropy and random forest," *Security and Communication Networks*, vol. 2018, 2018.