International Journal of Information Management Data Insights 1 (2021) 100048

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification

Onkar Thorat, Nirali Parekh*, Ramchandra Mangrulkar

Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

ARTICLE INFO

Keywords:
Information management security
Machine learning
Distributed denial of service
Reflection attack
Exploitation attack

ABSTRACT

Distributed Denial of Service (DDoS) attack is one of the most dangerous attacks that result in bringing down the server(s) and it is essential to classify the exact attack to implement robust security measures. In this work, we present an approach for detecting the prominent DDoS attacks that can be carried over Transport Layer protocols. Four different levels are taken into consideration which helps to classify one of the 11 different attacks. A bigger problem is divided into smaller ones and then conquered. This approach, called TaxoDaCML - Taxonomy-based Divide and Conquer approach using ML minimizes computational cost and at the same time maintains the required accuracy. Results prove that our approach achieves 99.9% accuracy for DDoS attack detection and more than 85% for DDoS attack classification. Comparison of TaxoDaCML is done with the previous works and is found to perform better for DDoS attacks classification.

1. Introduction

In the recent times, the information security infrastructures such as payment gateways, search engines, banks, and social websites, are growing immensely to provide online services to countless users over the world. It has led to an exponential rise in network traffic which has further induced rise in misuse of the Internet. The security and privacy issues create a fear in the mind of the end-users, thus deterring them from using the online services (Chatterjee, Kar, Dwivedi, & Kizgin, 2019; Singhal & Kar, 2015). Information security management works have been proposed, and implemented, but there are significant challenges when it comes to its technical solutions (Soomro, Shah, & Ahmed, 2016). Of the various types of attacks present on the Internet, Distributed Denial of Service (DDoS) attack give rise to severe threats to the organizations, their architecture, and impairs user from accessing information and data. DDoS attacks have been dominant in the last few decades and with the growth of developing applications, they are likely to increase in the future. For example, in 2018, Github had undergone one of the largest DDoS attacks in history with a traffic volume of around 1.35 Tbps (Newman, 2018). In addition, monitoring network threats have become one of the greatest challenges for most organizations and they spend exorbitant amounts of their money to secure the landscapes, yet large scale cyber-attacks take place as the attackers become advanced, and tools to counter the attacks become obsolete (Ariyaluran Habeeb et al., 2019).

The fundamental principle behind any DDoS attack is the same. The attacker exploits some network protocol and floods the server with spoofed request packets. The victim server cannot discriminate between these packets and starts serving all of them. In an attempt to satisfy all the requests, the server is congested and goes down. As a result, an attacker manages to deplete all available server resources, creating a denial of service. Since the packets are sent from different devices rather than a single source machine, it is called a Distributed Denial of Service attack.

The majority of the DDoS attacks are reported to be carried over the User Datagram Protocol (UDP), but in the past few years, attacks via the Transmission Control Protocol (TCP) have been on the rise (Jiao et al., 2017). These attacks are executed by exploiting network protocols. For instance, Domain Name Server (DNS) and Lightweight Directory Access Protocol (LDAP) can operate over both TCP and UDP. An example of a DDoS attack carried over TCP is the SYN attack, that overwhelms the server by flooding it with TCP SYN packets.

Sharafaldin, Lashkari, Hakak, & Ghorbani (2019) have generated the dataset CICDDoS2019 that contains the most recent DDoS attacks of the time and hence, is used by us. They also proposed a new taxonomy for DDoS attacks based on TCP and UDP that is used by us with a modification. They had trained various machine learning algorithms during their experimentation on their CICDDoS2019 dataset and concluded that the Iterative Dichotomiser 3 (ID3) algorithm worked the best with an accuracy of 70%. But an evaluation of their model shows that there is still a

* Corresponding author.
E-mail address: nirali25parekh@gmail.com (N. Parekh).

https://doi.org/10.1016/j.jjimei.2021.100048
Received 22 June 2021; Received in revised form 12 November 2021; Accepted 13 November 2021

chance for improving the accuracy and adjusting the tradeoff between precision and recall.

This work has attempted to classify the 11 most common and modern types of DDoS attacks (Sharafaldin et al., 2019) in the management of information security. Hence, it aims to detect and classify larger number of DDoS attacks that can be carried over TCP, UDP or both through the approach named TaxoDaCML - divide using taxonomy and conquer using machine learning. We divide a bigger classification problem into sub-problems using taxonomy of the DDoS attacks, and conquer these sub-problems using different machine learning algorithms. Hence, the objectives of this work are:

- Detection and classification of larger number of modern-day DDoS attacks that are carried over the IT infrastructure by the attackers.
- Detection of the onset of a DDoS attack as soon as possible by keeping the computational cost as well as time of detection minimum, thus providing a quick redressal to secure information management infrastructure.

The rest of the paper is organized as follows. In Section 2, the previous related work is being reviewed and shortcomings are identified. Section 3 provides the importance of this work and shortcomings in the previous works. In Section 4, the TaxoDaCML approach is presented. In Section 5, the Experimentation carried out by us is described. In Section 6, the Results and their Analysis is provided. Section 7 is the Discussion section. Finally, in Section 8, this work is concluded.

2. Background and related work

Modern Intrusion Detection Systems mostly utilize machine learning approaches for detecting anomalous behavior using statistical modeling (Li, Deng, Lee, & Wang, 2019). Recent works exploring machine learning based solutions for information management problems like Botnet, ransomware attacks have been proposed by Reshmi (2021), Singh, Guntuku, Thakur, & Hota (2014). Securing the IT landscapes from various threats is extremely important as seen in various works. Chatterjee, Kar, & Gupta (2018) address the issue of cybersecurity awareness and training in their work. System security and information security systems are discussed in this work by considering the well-being and security of citizens of smart cities. It stresses the importance of knowledgeable IT authority and information management for a successful smart city. Batra, Jain, Tikkiwal, & Chakraborty (2021) proposed the use of meta-heuristic algorithms for the classification of spam emails from legitimate emails. They utilized the biological structure-based optimized techniques and integrated with the KNN algorithm. Their experiments led to an f1-score of 74.46% with a classification time of 99.67 seconds using the whale optimization approach.

According to Kushwaha, Kar, & Dwivedi (2021), penetration of the internet in the last few years has given rise to smart devices and various kinds of data in large quantities are being generated. Such data may be analyzed for valuable insights by the businesses. Such a growth in IoT devices does come with certain issues required to be tackled. Chatterjee & Kar (2018) have discussed various security threats posed by the IoT enabled devices in their article and identified the privacy of a user as a major threat. Use of such IoT enabled smart devices may also be used to carry out DDoS attacks. For example, the cyber attack that exploited millions of internet-connected devices by infecting them with botnet, blocked few of the most popular websites like Twitter, Spotify, and CNN in 2016, producing a DDoS attack (Thielman & Hunt, 2016). Thus, securing the networks against DDoS attacks is of cardinal importance for developing the smart cities. At the same time it is also important to spread awareness and make the public technically sound about various attacks. For this, Mittal, Gupta, Chaturvedi, Chansarkar, & Gupta (2021) proposed a Serious Game approach that can strengthen cybersecurity in information management systems by permeating the blockchain technology. This approach has resulted in the development of six Serious Games and the DDoS attacks are mainly related to the Firewall game.

Detecting DDoS attacks for combating threats to information security using machine learning has proven to be effective in various works (Das, Mahfouz, Venugopal, & Shiva, 2019; Hou, Fu, Cao, & Xu, 2018; Patil, Rama Krishna, Kumar, & Behal, 2019; Wang, Lu, & Qin, 2020; Zargar, Joshi, & Tipper, 2013). Zargar et al. (2013) provided a survey report of defense mechanisms against DDoS attacks. According to their review, the DDoS attacks are categorized in two categories based on the protocol - Network/Transport level and Application level flooding attacks. They suggested that defense mechanisms can be deployed as centralized or distributed. Source-based, network-based, and destination-based are three types in centralized deployment while hybrid is the only approach in a distributed deployment. Destination-based centralized deployment provides high defense strength while source-based and network-based centralized deployment provides low defense strength.

The CICDDoS2019 dataset, generated by Sharafaldin et al. (2019), contains 12 attack classes and one benign class. In their work, they also studied and reviewed some other available datasets such as CAIDA “DDoS Attack 2007” and DARPA. A new taxonomy for DDoS attacks is proposed by them which is later used in this work with a modification.

CICDDoS2019 dataset contains an imbalance in the class distribution and a large number of features. Thus, it becomes necessary to select relevant features for the effective working of the machine learning algorithms. A few studies specifically focusing on feature selection techniques were found, and insights were drawn from them. Wang et al. (2020) combined sequential feature selection with Multilayer Perceptron. This was used to select the optimal features during the training phase. They also designed a feedback mechanism to dynamically reconstruct the detector during considerable detection errors. Das, Venugopal, Shiva, & Sheldon (2020) proposed an ensemble model for the feature selection. They used the majority voting technique to combine seven feature selection methods and generated a superior set of features. He, Zhang, & Lee (2017) proposed a machine learning based source side DDoS attack detection system and analyzed statistical features of different kinds of attacks in their framework. Sen, Gupta, & Ahsan (2020) detected DDoS attacks using AdaBoosting and received 93% detection accuracy with a low false positive rate.

Deep learning is a subset of machine learning that eases out the prior tasks of data pre-processing and feature selection with no effect on the algorithmic performance. Such models are inspired from human brains and may be considered as secondary ML algorithms capable of performing variety of tasks (Aggarwal, Mittal, & Battineni, 2021). Yuan, Li, & Li (2017) worked on the ISCX2012 dataset and proposed a Bi-directional Recurrent Neural Network architecture to detect DDoS attacks. Their DeepDefense approach included Convolutional and Recurrent Neural Networks and fully connected layers, which allowed them to lower the error rate to 2.103% which was previously 7.517%. Elsayed, Le-Khac, Dev, & Jurcut (2020) proposed a deep learning model that combined Recurrent Neural Network and Autoencoder. Their model received an accuracy of greater than 99% for detecting DDoS attacks. However, their work did not classify the types of DDoS attacks. Cil, Yildiz, & Buldu (2021) in their work suggested using Deep Neural Network (DNN) to detect DDoS attacks. They conducted their experiments on the CICDDoS2019 dataset and detected network attacks with 99.99% accuracy. The types of attacks were classified as ‘Reflection’ and ‘Exploitation’ with an accuracy of 94.57%.

Das et al. (2019) built an ensemble machine learning model of MLP, KNN, SVM, and Decision Tree for DDoS attack detection. They used the NSL-KDD dataset in their approach and achieved an accuracy of 99.1% for the ensemble model. Hou et al. (2018) detected DDoS attacks based on machine learning through network flow analysis. They extracted the flow and pattern based features from Network Flow data. They then built the Random Forest model using their extracted features and achieved more than 99% accuracy.

Thus, machine learning has proven to be powerful in real-world scenarios for detecting and classifying DDoS attacks, if trained on high-quality datasets. Feature selection can reduce computational calcula-

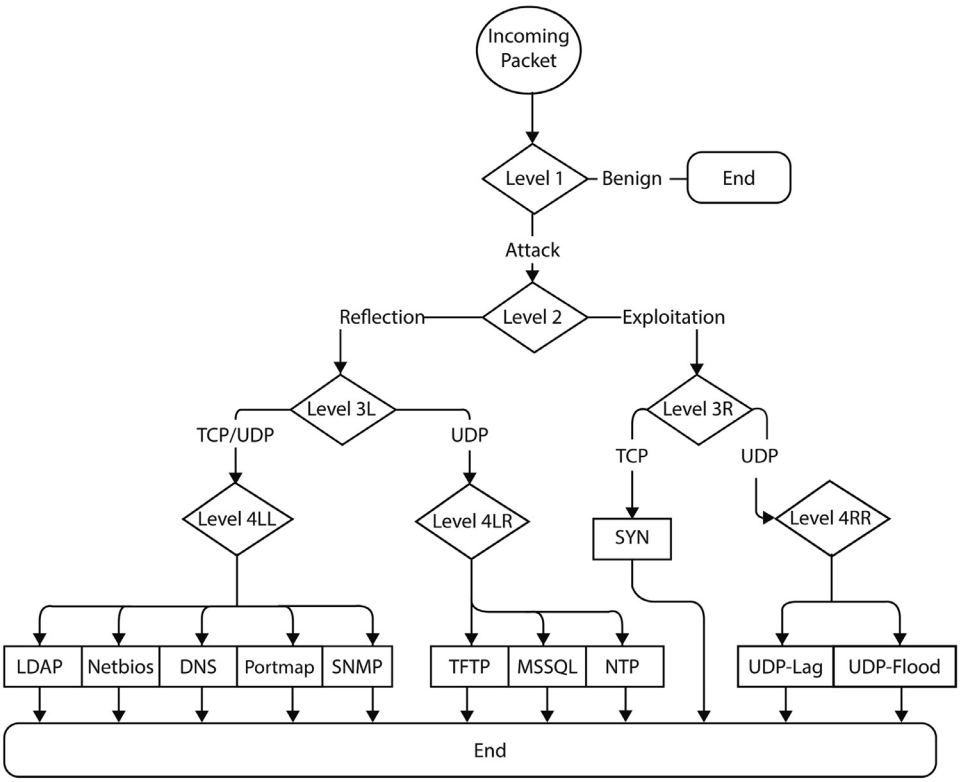


Fig. 1. The TaxoDaCML approach.

tions and improve the speed and accuracy of detection. Deep learning models are popular because they need a minimal feature selection and yet give a lot more reliability in detection. Thus, they have a substantial role in strengthening the information security management of the organizations.

3. Motivations

Most of the previous research has addressed the DDoS attack detection problem as a binary classification with benign and attack as two classes. However, it is unrealistic to develop defenses in an organization's security infrastructures that is an all-in-one solution for all possible DDoS attacks. The detection of the exact type leads to determination of the specific category of DDoS attack and helps to design customized responses (Mirkovic & Reiher, 2004). The exploited characteristic in the TCP SYN attack, for example, is the instantaneous allocation of significant space in a connection queue after receiving the TCP SYN request. Here, by knowing the exact protocol the intruder is trying to exploit, customizable threshold alerts and traffic diversion using more specific routing ensures robust and timely mitigation and restorations (Lee, Jun 29, 2020).

The research works that have attempted to classify the DDoS attacks has had one or more following shortcomings:

1. They have not been able to get a reliable accuracy. For instance, Sharafaldin et al. (2019) achieved an accuracy of 70% and a significant tradeoff between precision and recall.
2. They have classified only a small number of attacks or made a broader classification. For instance, Cil et al. (2021) detected and classified the DDoS attacks, but only into two broad categories, namely 'Reflection' and 'Exploitation'. Also, they used deep learning models for the detection of DDoS attacks which in real-world scenarios might be time-consuming.
3. They have used an outmoded dataset that does not contain modern attacks. For instance, Das et al. (2019) used the NSL-KDD dataset that is very old and contains attacks that may not be relevant to current attacks.

Initially, the attackers believed that TCP-based DDoS attacks could not generate amplifications at the scale of UDP-based attacks. This, combined with the complexity of the three-way TCP handshake, has caused the attackers to prefer DDoS attacks based on UDP. However, the idea that TCP-based attacks are less effective than UDP-based attacks is starting to change, and modern DDoS attacks are beginning to take place on TCP. The previous research has only focused on TCP-based SYN attacks.

4. The TaxoDaCML approach

Sharafaldin et al. (2019) put forward a taxonomy for Distributed Denial-of-Service attacks in their work. The original taxonomy is based on the transport layer protocols TCP and UDP. The MSSQL attack is assigned as a Reflection attack transmitted over TCP. In our work, we modified this and assigned it as a Reflection attack transmitted over UDP since the MSSQL server is primarily designed to accept UDP requests (Hefley, 2003).

To identify the exact DDoS attack class out of the 11 attack classes, the overall classification problem is divided into seven smaller classification problems. The wrong classes are kept on eliminating with the help of rules defined by the taxonomy of DDoS attacks. Each of the seven smaller classification problems are solved using machine learning algorithms. The seven smaller problems are distributed at four levels overall. The schematic flowchart in Fig. 1 describes the overall structure.

Consider an incoming network packet p and the following sets.

$$Level1 = \{benign, ddos\}$$

$$Level2 = \{ref, exp\}$$

$$Level3L = \{tcpudp\_r, udp\_r\}$$

$$Level3R = \{tcp\_e, udp\_e\}$$

$Level4LL = \{ldap, netbios, dns, portmap, snmp\}$

$Level4LR = \{tftp, mssql, ntp\}$

$Level4RL = \{udplag, udpflood\}$

where the following symbols resembles the corresponding attack type:
$benign$ = Benign Packet
$ddos$ = DDoS attack
ref = Reflection DDoS attack,
exp = Exploitation DDoS attack
$tcpudp\_r$ = Packet that works over TCP or UDP for Reflection DDoS attack
udp_r = Packet that works over UDP for Reflection DDoS attack
tcp_e = Packet that works over TCP for Exploitation DDoS attack also termed as SYN DDoS attack
udp_e = Packet that works over UDP for Exploitation DDoS attack
$ldap$ = LDAP attack
$netbios$ = NetBios attack,
dns = DNS attack
$portmap$ = Portmap attack
$snmp$ = SNMP attack
$tftp$ = TFTP attack
$mssql$ = MSSQL attack
ntp = NTP attack
$udplag$ = UDP-Lag attack
$udpflood$ = UDP-Flood attack
The goal is to find following,

1. Whether $p \equiv benign$ or $p \equiv ddos$
2. If $p \equiv ddos \Rightarrow$ classify p into one of the 11 attack classes - $ldap$, $netbios$, dns, $portmap$, $snmp$, $tftp$, $mssql$, ntp, syn, $udplag$, $udpflood$

Given packet p,

1. Find $x | x \in Level1$ and $x \equiv p$ using Decision Tree algorithm.
   (a) If $x \equiv benign \Rightarrow p \equiv benign$ & no need to check further.
   (b) If $x \equiv ddos \Rightarrow$ find $y \mid y \in Level2$ & $y \equiv p$ using Random Forest algorithm.
      (1) If $y \equiv ref \Rightarrow$ find $z \mid z \in Level3L$ & $z \equiv p$ using Decision Tree algorithm.
         (1) If $z \equiv tcpudp\_r \Rightarrow$ find $t \mid t \in Level4LL$ & $t \equiv p$ using ANNs. t = target ddos attack class & no need to check further.
         (2) If $z \equiv udp\_r \Rightarrow$ find $t \mid t \in Level4LR$ & $t \equiv p$ using Random Forest algorithm. t = target ddos attack class & no need to check further.
      (2) If $y \equiv exp \Rightarrow$ find $z \mid z \in Level3R$ & $z \equiv p$ using KNN algorithm.
         (1) If $z \equiv tcp\_e \Rightarrow p \equiv syn$ & no need to check further.
         (2) If $z \equiv udp\_e \Rightarrow$ find $t \mid t \in Level4RL$ & $t \equiv p$ using Random Forest algorithm. t = target ddos attack class.

Let us understand this approach better with an example. Consider a scene where an incoming packet is a TFTP attack in nature. Now, let us see how the TaxoDacML framework deals with this packet. Initially, the incoming packet passes through Level 1 to assess whether it is malicious or benign. The trained ML model would classify it as malicious and thus, it now passes further to discern the exact attack. On passing through Level 2, the trained ML model at this level would classify it as a Reflection based DDoS attack. The packet now enters in the penultimate Level 3L, where it is classified as a Reflection based DDoS attack carried over UDP. Now, the packet, finally enters in the last Level (Level 4LR in this case) where the exact attack is classified by the ML model, that is, DDoS attack exploiting TFTP. If the models are trained well and demonstrate a good performance, this framework is sure to detect the exact type of DDoS attack.

The features required to train different machine learning algorithms are described in Table 2. The choice of the final machine learning algorithm for each level is made based on experimentation. The experimentation carried out and various factors used to finalize the algorithm for each problem are mentioned in subsequent sections.

Thus, a hierarchical approach is incorporated in which a bigger problem is divided into smaller problems based on the taxonomy. These smaller problems are solved using different machine learning algorithms. This approach does has its pros and cons. Below are the reasons why this method is expected to work better than other methods proposed in the literature:

1. It offers the flexibility to choose different features and algorithms for various smaller problems and helps to make the overall prediction stronger. It is possible to work on the level which performs poorly and analyze the reason at a more granular level.
2. Increases the likelihood of correct prediction since most of the problems become of binary classification. At the same time, this approach seeks to eliminate the possibilities of wrong classes at every level.
3. Multiple smaller and lightweight models connected are better than a large bulky model.

At the same time, there are some drawbacks to this method, for example, if a prediction goes wrong at a certain stage like level 2, the overall prediction would go wrong. For this reason, it becomes critically important to have good accuracy at every level.

Fig. 2 gives an overview in graphical form for the process used in training machine learning models.

4.1. Dataset description

The CICDDoS2019 (Sharafaldin et al., 2019) consists of benign and modern DDoS attacks resembling real world scenario. There are total 12 attack classes present in this dataset. The overall distribution of these classes is highly imbalanaced and needs proper preprocessing and cleaning. Moreover, the data contains large dimensions and is necessary to either lower the dimensions or perform feature extraction. The attack classes and their description is as mentioned below and summarized in Table 1.

1. LDAP - On corporate internal networks, this protocol is extensively used for services based on the directory. Because of the amplification of traffic, using the LDAP protocol is harmful to the victim. Small searches by the attacker might result in large answers from Internet servers, flooding the victim.
2. NetBIOS - Allows programs on different machines to interact and utilize common resources through sessions. An attacker might compel a target system to delete its own valid name from its name table and not respond to other NetBIOS requests by delivering faked ”Name Surrender” or “Name Dispute” signals to it.
3. DNS - When an attacker sends spoofed DNS queries that are exact copies of legitimate DNS queries from a large number of source IPs, this is known as a DNS spoofing attack. The server uses up all of its capacity in order to fulfill all of the requests. As a result, the assault uses up all of the available bandwidth until it is exhausted.
4. Portmap - Portmapper is a service that uses TCP or UDP port 111 to guide clients to the correct port number. By transmitting packets with a faked Internet address, the Portmap attack initiates volumetric assaults, culminating in resource magnification.
5. SNMP - This attack is done by delivering a large number of tiny packets with the target’s faked IP address. These faked requests are then utilized to deliver UDP floods to the target as answers from these devices.
6. TFTP - An attacker uses a TFTP amplification attack to transmit a specifically altered TFTP route request to a TFTP server with a faked source IP address for the file. This results in more message packets being sent to the victim.
7. NTP - A network protocol for synchronizing computer clocks to a certain period. When an attacker sends tiny packets with a faked IP of the target to internet-connected devices running NTP, this is known as an NTP Flood attack.

Table 1
Prominent DDoS attacks.

Attack name	Protocol exploited	Amplication Factor	Transport Layer Protocols used	Reference(s)
LDAP	Lightweight Directory Access Protocol	46–70	TCP and UDP	(Tuttle et al., 2004)
NetBios	Network Basic Input/ Output System	2.5–3.8	Mainly TCP but can work on UDP with less accuracy	(Schwaderer, 1988), (Thomas, 1987)
DNS	Domain Name System	28–54	TCP and UDP	(Alieyan, Kadhum, Anbar, Rehman, & Alajmi, 2016), (Arends, Austein, Larson, Massey, & Rose, 2005)
Portmap	Portmapper	7–28	TCP and UDP	(Labs, 2015), (Srinivasan, 1995)
SNMP	Simple Network Management Protocol	6.3	Originally made for UDP but can operate over TCP	(Hare, 2002)
TFTP	Trivial File Transfer Protocol	60	UDP	(Sollins, 1992), (Sieklik, Macfarlane, & Buchanan, 2016)
NTP	Network Time Protocol	556.9	UDP	(Czyz et al., 2014), (Kawamura, Fukushi, Hirano, Fujita, & Hamamoto, 2017)
MSSQL	Microsoft SQL Server Resolution Protocol	25	Mainly UDP	(Hefley, 2003)
SYN	A packet part of TCP three-way handshake	770	TCP	(Bogdanoski, Suminoski, & Risteski, 2013)
UDP-Lag	UDP	-	UDP	(Kotey, Tchao, & Gadze, 2019), (Sharafaldin et al., 2019)
UDP-Flood	UDP	4–350	UDP	(Xiaoming, Sejdini, & Chowdhury, 2010), (Singh & Juneja, 2010)

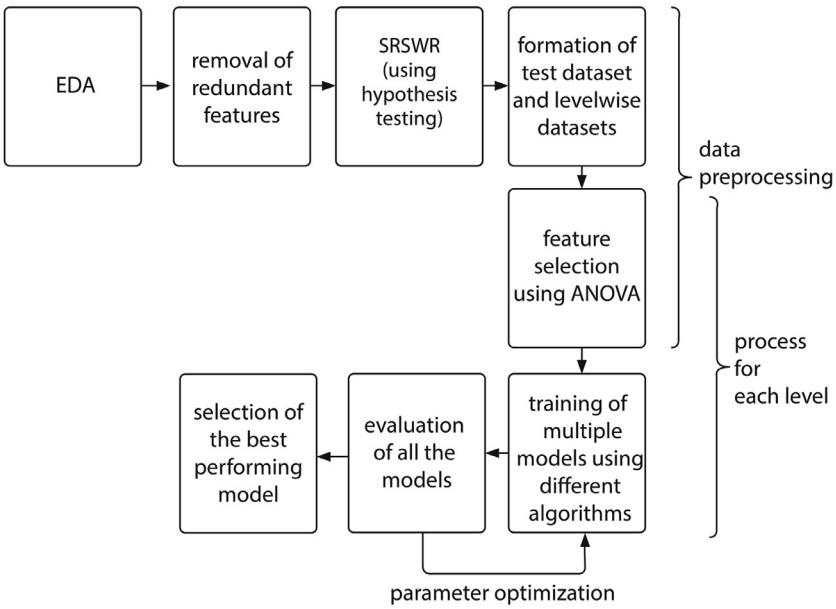


Fig. 2. Overview process of TaxoDaCML.

8. MSSQL - When a Microsoft SQL Server replies to a client request trying to exploit the SQL Database Resolution Protocol on UDP port 1434, this attack takes place.
9. SYN - When an intruder exploits a sequence of SYN requests to a target device, it is called a SYN attack. The system's resources have been depleted, and it no longer responds to valid traffic.
10. UDP-Lag - The link between both the client and the server is broken as a result of this assault. This tactic is most commonly utilized in online games when players try to stifle or halt the movement of other players. This attack may be carried out in two ways: with a lag switch or with an application that consumes other users' network bandwidth.
11. UDP-Flood - When an attacker is able to overwhelm the target computer with a large number of faked data packets across the UDP connection, this is known as a flood attack. As a result, all channel capacity has been consumed and drained.

4.2. Data cleaning

The raw datasets usually are not correct, consistent or complete, so they need to be processed and cleaned before applying any ML technique (Hassan et al., 2021). On performing the Exploratory Data Analysis (EDA) of the CICDDoS2019 dataset using pandas (Wes McKinney, 2010; pandas development team, 2020) in Python, it was noticed that the dataset contained a few features that had constant values or values belonging to the data type 'object', thus making those features redundant. These redundant features would have made it harder and more time-consuming to detect attacks and were hence removed.

4.3. Dataset creation by SRSWR

The CICDDoS2019 dataset consists of data captured from two days, each day having different attack categories and a highly imbalanced class distribution. As a result, we decided to create subsets by undersampling each class using appropriate hypothesis test. This resulted in a balanced class distribution and elimination of bias (if any).

The following points provide a more in-depth overview of the process of subset formation.

4.3.1. Hypothesis testing

Statistical Hypothesis Testing is a means of testing the results of a survey or experiment to check if they are meaningful. Hypothesis Testing was used while forming the test dataset and at each level, to make sure that the overall distribution in each dataset and subset remain proportional to each other and also to the original CICDDoS2019 dataset. Because of the huge number of instances to be tested, it is implicit that

Table 2
Details and ANOVA scores of prominent features.

		Anova Scores						
Feature	**Feature Details**	**Level 1**	**Level 2**	**Level 3L**	**Level 3R**	**Level 4LL**	**Level 4LR**	**Level 4RR**
Source Port	Origin Port	46976	54076	118209	5201	161	279373	15244
Flow IAT Mean	Average time between two flows	708	5476	13896	24	68	81684	14041
Flow IAT Std	Std. deviation time between two flows	3254	5489	13812	2	72	80387	13776
Flow IAT Max	Maximum time between two flows	5862	6451	13857	232	69	76511	14340
Fwd IAT Mean	Mean time of 2 packets in forward direction	1592	6389	13852	3637	68	76983	14575
Fwd IAT Std	Std. deviation time between 2 packets sent in forward direction	3719	5764	13768	11	72	76696	13896
Fwd IAT Max	Maximum time between 2 packets sent in forward direction	5377	6454	13847	231	69	76438	14341
Fwd Packets per second	No. of forwarding packets per second	66928	6766	26173	209	5722	94941	2055
Total Fwd Packets	Total packets in the forward direction	1	3805	155	1426	3	73350	3788
Total Length of Fwd Packets	Total packets size in the forward direction	274	12621	14789	47078	38227	70466	62399
Average Packet Size	Average size of packets	79989	107748	54026	80523	844955	74843	138504
Subflow Fwd Packets	Average no. of packets in a sub-flow in forward direction	1	3804	155	1427	3	73350	3788
Subflow Fwd Bytes	Average no. of bytes in a sub-flow in forward direction	274	12621	14789	47078	38227	70466	62399
act_data_pkt_fwd	No. of data packets in forward direction	614	7192	17986	631	139	73342	37326
Protocol	Network protocol (UDP/TCP)	61111	226614	221	116052	145	323	514973
Fwd Packet Length Max	Maximum packets size in forward direction	39162	118674	42556	90042	850193	20465	182982
Fwd Packet Length Mean	Average packets size in the forward direction	93660	120184	43170	87899	849495	23492	170786
Fwd Packet Length Min	Minimum packets size in the forward direction	100750	113191	47066	85391	846187	44107	158921
Max Packet Length	Maximum length of a flow	4838	118657	42524	89750	850344	20247	182080
Min Packet Length	Minimum length of a flow	101815	113371	47066	85911	846156	44108	164118
Packet Length Mean	Mean length of a flow	78909	120761	43112	87820	849680	23222	171644
ACK Flag Count	No. of packets with ACK	71	226702	193	115585	138	472	512571
init_Win_bytes_forward	No. of bytes sent in the first window in forward direction	13577	201908	46	110790	77	58	480844
Avg Fwd Segment Size	Average size in forward direction	93660	120184	43171	87899	849495	23492	170786
Destination Port	Target port	57590	5	0	16	1	4	12
URG Flag Count	No. of packets with URG	75106	3	40	12	21	143	47
Down/Up Ratio	Download and upload ratio	62963	19458	31	83	98	93	6383
Inbound	Details coming-in to a network	381379	13	0	0	75	85	122
Bwd Packet Length Min	Minimum packets size in backward direction	27509	3117	32	6667	107	2	54
Fwd Header Length	Length of the header in forward direction	196	183	107	989	108	156	850
min_seg_size_forward	Minimum size of segment in forward direction	2590	2332	84	1193	936	3953	1111
Total features		12	11	9	11	20	14	5

each feature follows an approximate normal distribution for the statistical Z-test. We conducted Multiple-univariate tests and tested each feature in a sample with corresponding feature in the population. Hypothesis Testing was conducted at a significance level of 95%.

The Null Hypothesis (denoted by H_0) stated that the mean of the population and the sample is equal while the Alternate Hypothesis (denoted by H_A) stated that the mean of the population and the sample is not equal. Thus, it formed a two-tailed test.

$$H_0 : \mu 1 = \mu 2 \quad (1)$$

$$H_A : \mu 1 \neq \mu 2 \quad (2)$$

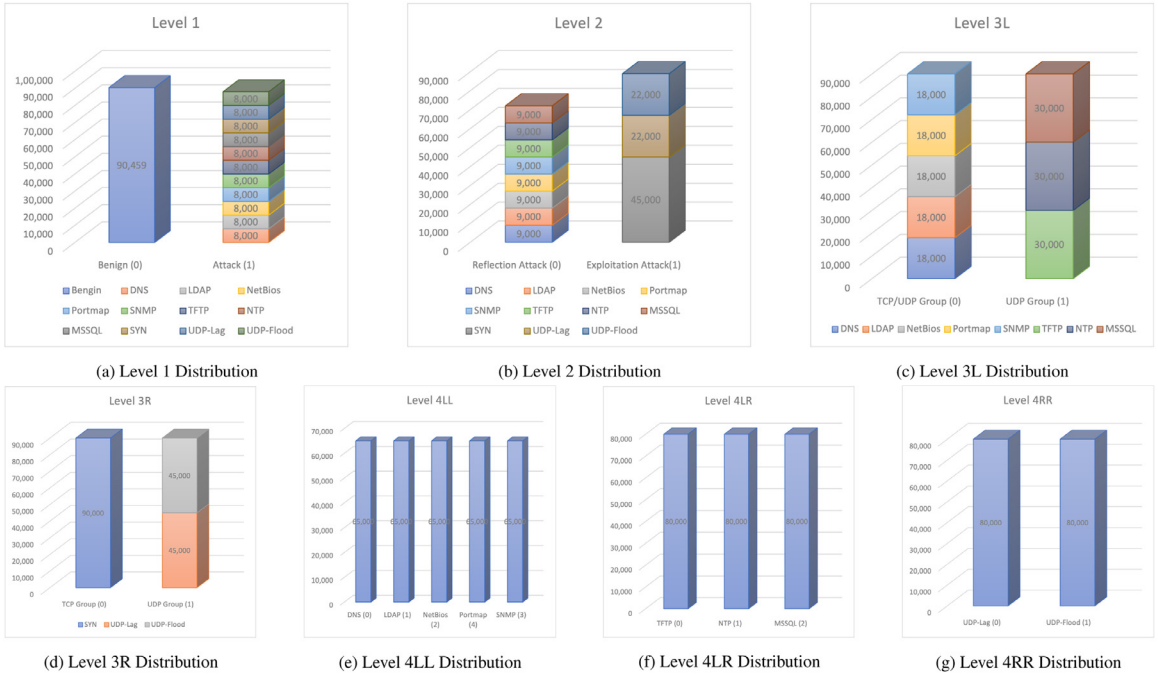
$where, \mu 1 = mean\ of\ the\ sample$

$\mu 2 = mean\ of\ the\ population$

In case we received such a feature that rejected the Null Hypothesis, we re-sampled the dataset until each feature accepted the Null Hypothesis. The Python statsmodels module (Seabold & Perktold, 2010) was used to perform this analysis.

4.3.2. Formation of test dataset

Before starting with any preprocessing or training, the instances of the benign class and each of the 11 attack classes were sampled with the Hypothesis Testing. These instances were also removed from the original dataset so that they remain unseen during the training phase. This test dataset comprised of 22,606 data points from the benign class and 22,000 data points from each of the 11 classes. Therefore, the test



(a) Level 1 Distribution (b) Level 2 Distribution (c) Level 3L Distribution

(d) Level 3R Distribution (e) Level 4LL Distribution (f) Level 4LR Distribution (g) Level 4RR Distribution

Fig. 3. Distribution for each level after performing sampling. The numbers in the brackets provide the label assigned that is used while training and testing.

dataset included a total of 264,606 observations. This dataset was used in the end to verify the accuracy of the entire framework as a whole.

4.3.3. Formation of other levelwise datasets

At each level, we formed the main training datasets, as per the attack classes required for the classification task at that level, by sampling the CICDDoS2019 with Hypothesis Testing as mentioned earlier. The attack classes required at any level are as per the Section 4. Following sampling at each level, the sampled instances were also removed from the training dataset. Hence, Simple Random Sampling with Replacement was performed (SRSWR). The class distribution for each level and labels assigned to classes is depicted in Fig. 3.

4.4. Feature Selection

Feature Selection is one of the most significant stages that contribute to the result and performance of machine learning models. Selecting and reducing the features from datasets increases the training and testing speed, the accuracy of the classifier, and the computational cost of modeling.

Out of the 67 features and 1 label remaining after the data cleaning process, various Statistical Feature Selection techniques such as Analysis of Variance (ANOVA) and Mutual Information were tried. The ANOVA technique was more suitable for the case since it is widely used when one variable is numeric and one is categorical (Kuhn & Johnson, 2019, Chapter 10). The ANOVA algorithm, often known as the ANOVA f-test comes under F-statistic metrics. The results of this test can be used for Feature Selection where the features that are less dependent on the target variable can be removed from the dataset.

Table 2 provides the ANOVA scores of the chief features along with their description. The underlined figures denote the ANOVA scores of the shortlisted features that were used to train the respective models.

5. Experimentation

All the experiments were performed on a 2.3 GHz 8-core Intel i9 processor with 16 GB of RAM. For each level, we experimented by training and testing different parametric and non-parametric machine learning algorithms to find the best one for each level. These algorithms with their different parameters are mentioned below. The training dataset for each level was split into training and validation sets in the ratio 3:1. The validation set was used to test the model at each individual level and finalize the best suited algorithm for that level. The selected final algorithms are mentioned in the Section 6 section with the appropriate reasons. All these algorithms were implemented using scikit-learn (Buitinck et al., 2013) and keras (Chollet et al., 2015) in Python.

5.1. K-Nearest Neighbors

K-Nearest Neighbors (KNN) (Altman, 1992) is a lazy learning algorithm, that is, it does not learn until it is asked to predict. KNN is a non-parametric algorithm, based on similarity of the features. A data point is classified based on the resemblance of that point to the training set. A majority vote of its neighbors is used to assign an object to the most common class among the K nearest neighbors. Different distance metrics can be used to compute the similarity between the data points. A few of them are mentioned below.

$$Euclidean\ Distance = \left(\sum_{i=1}^{k}(x_i - y_i)^2\right)^{\frac{1}{2}} \tag{3}$$

$$Minkowski\ Distance = \left(\sum_{i=1}^{k}(x_i - y_i)^p\right)^{\frac{1}{p}} \tag{4}$$

$$Manhattan\ Distance = \sum_{i=1}^{k}|x_i - y_i| \tag{5}$$

$where,\ x_i,\ y_i = x,\ y\ coordinates\ for\ ith\ point$

$k = no.\ of\ nearest\ neighbors\ to\ be\ considered$

$p = power\ parameter$

We experimented with various parameters such as different values of k (5 to 11) and distance metrics (Euclidean, Minkowski and Manhattan Distance) to find the ones that best suited the problem at hand.

5.2. Decision trees and random forest

Decision Trees (Breiman, Friedman, Stone, & Olshen, 1984) is a non-parametric machine learning algorithm and can be used for classification as well as regression tasks. This algorithm breaks a dataset into subsets and develops a tree-structured model for the same. Decision Trees is lightweight and thus make the classification process extremely fast. Decision Trees can be constructed using ID3 algorithm or CART algorithm. The splitting of nodes could be done using the Gini Index (when using Gini Impurity) or Entropy (when using Information Gain).

Gini index and Entropy is given as:

$$Gini = 1 - \sum_{j=1}^{c} (p_j)^2 \tag{6}$$

$$Entropy = -\sum_{j=1}^{c} p_j log_2(p_j) \tag{7}$$

$where,\ p_j = proportion\ of\ samples\ belonging\ to\ class\ j$

$c = number\ of\ unique\ labels$

We experimented with various parameters such as the depth of the tree and a choice between Gini Index and Entropy while constructing a Decision Tree. The depth of the tree is an important parameter to adjust to avoid overfitting and hence, the authors tested a variety of depths.

The Random Forest algorithm (Tin Kam Ho, 1995) is an ensemble of unpruned classification trees and is sturdy against overfitting. Randomness can be introduced at two places in this algorithm. First, a bootstrapped dataset is created randomly to train a particular tree, and second, a subset of the variables of the bootstrapped dataset is created randomly to choose a node from that subset for the tree. While training, this algorithm constructs numerous Decision Trees and the predictions from these trees are pooled to provide a final prediction.

Most of the parameters to consider when forming a Random Forest model remain similar to Decision Trees. One of the important parameters to experiment with Random Forest is the number of trees to be constructed in the forest. An optimum number of trees avoids overfitting and keeps computational cost in check.

5.3. Artificial Neural Networks (ANN)

Artificial Neural Networks (Hopfield, 1982) are deep learning algorithms built like the human brain, with neuron nodes interconnected like a web. These networks are made up of fully connected layers having multiple nodes. Each node has certain associated weights that are learnt during training. ANN consists of forward propagation, back propagation and the weight updation as three major steps while learning optimum weights for prediction. These steps continue in a cycle for a decided number of epochs or until the error is within desirable limits.

We used a four-layer Neural Network model and experimented with various hyperparameters such as the batch size and the learning rate. Batch sizes 32, 64, 128 and 256 and the learning rates 0.001, 0.01 and 0.1 were experimented. Different activation functions that include Rectified Linear Unit (ReLU) (Agarap, 2019) and Parametric Rectified Linear Unit (PReLU) (He, Zhang, Ren, & Sun, 2015) were used for the hidden layers, and Softmax (Liu, Wen, Yu, & Yang, 2017) was used for the last layer as it being a classification problem. We experimented Adam and Adamax optimizers (Kingma & Ba, 2015). Adam optimization uses the Stochastic Gradient Descent (SGD) with adaptive estimation of first-order and second-order moments while Adamax optimization is a variant of Adam optimizer based on the infinity norm. After relevant experiments, it was decided to keep the hyperparameters of this optimization function ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) to their default values.

6. Results

The machine learning models were tested on data packets that they have never seen before. At each level, the performance of different algorithms discussed in Section 5 were analyzed. While choosing the algorithm for each smaller problem at every level, two factors were taken into consideration in the following order of priority. First, the performance of the algorithm on the four evaluation metrics mentioned below was evaluated. Second, the computational cost of the algorithm was reviewed.

The evaluation metrics considered for choosing the appropriate algorithm:

1. Precision: The proportion of accurately predicted positive cases to all anticipated positive cases.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

2. Recall: The ratio of accurately predicted positive occurrences compared to the total number of examples in the class.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

3. F1-score: Using precision and recall, F1 score is a measure of quality of model's classifications.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

4. Accuracy: Proportion of accurately predicted instances with the entire set of instances.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

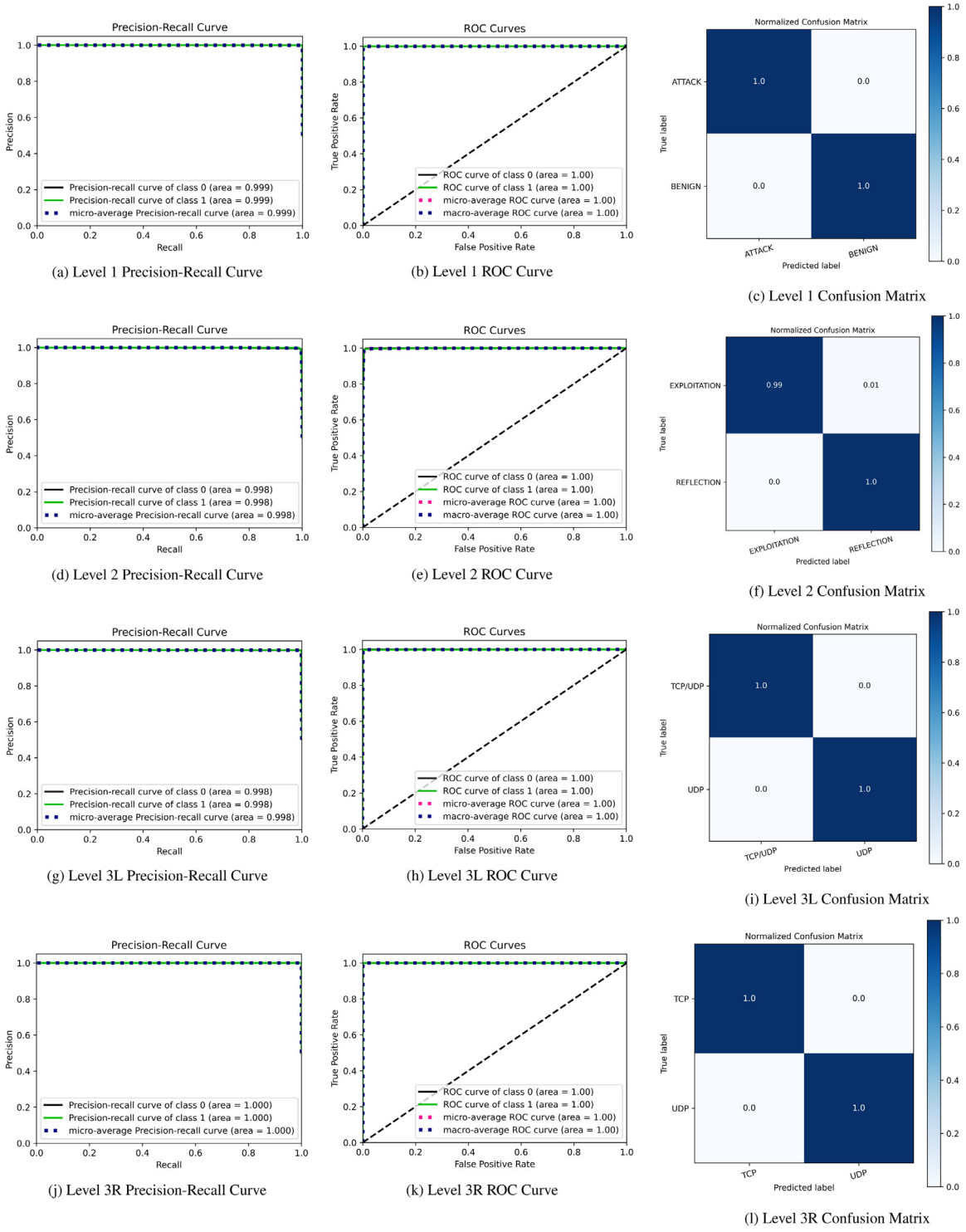
Figs. 4 and 5 provide the Receiver Operating Characteristic curve (ROC-curve) and Precision-vs-Recall curves for the models chosen for each problem at different levels. The classes mentioned in these curves correspond to the assigned labels as mentioned in Fig. 3.

Table 3 provides the evaluation metrics of the chosen algorithms at each level on the respective validation sets.

Now, the most efficient models at each level (as mentioned in Table 3) were combined together according to the TaxoDaCML approach explained in Section 4. This final and compiled framework was tested on the testing dataset that was sampled before sampling the training datasets from original dataset. It had the instances unseen by all the models in the framework. After randomly sampling into four subsets, the framework was individually tested on all the four subsets. It ensured that the testing done was more reliable and unbiased. Furthermore, the inference time to detect and classify each attack was also explored. These results are mentioned in Table 4.

7. Discussions

It is evident from Table 3 that Level 1 which is responsible for differentiating a DDoS attack from a benign packet has an accuracy of 99.9% and hence, it would be difficult for any packet responsible for DDoS attack to escape through it. Most of the levels had a reliable performance as evident from the evaluation metrics, Receiver Operating Characteristic (ROC) curve and Precision-Recall Curve. The level 4LL received low accuracy with conventional machine learning algorithms and hence, an Artificial Neural Network (ANN) was used for this level. But, the improvement received by the ANN was still not adequate. As a result, we used Principal Component Analysis (PCA) to



(a) Level 1 Precision-Recall Curve (b) Level 1 ROC Curve (c) Level 1 Confusion Matrix

(d) Level 2 Precision-Recall Curve (e) Level 2 ROC Curve (f) Level 2 Confusion Matrix

(g) Level 3L Precision-Recall Curve (h) Level 3L ROC Curve (i) Level 3L Confusion Matrix

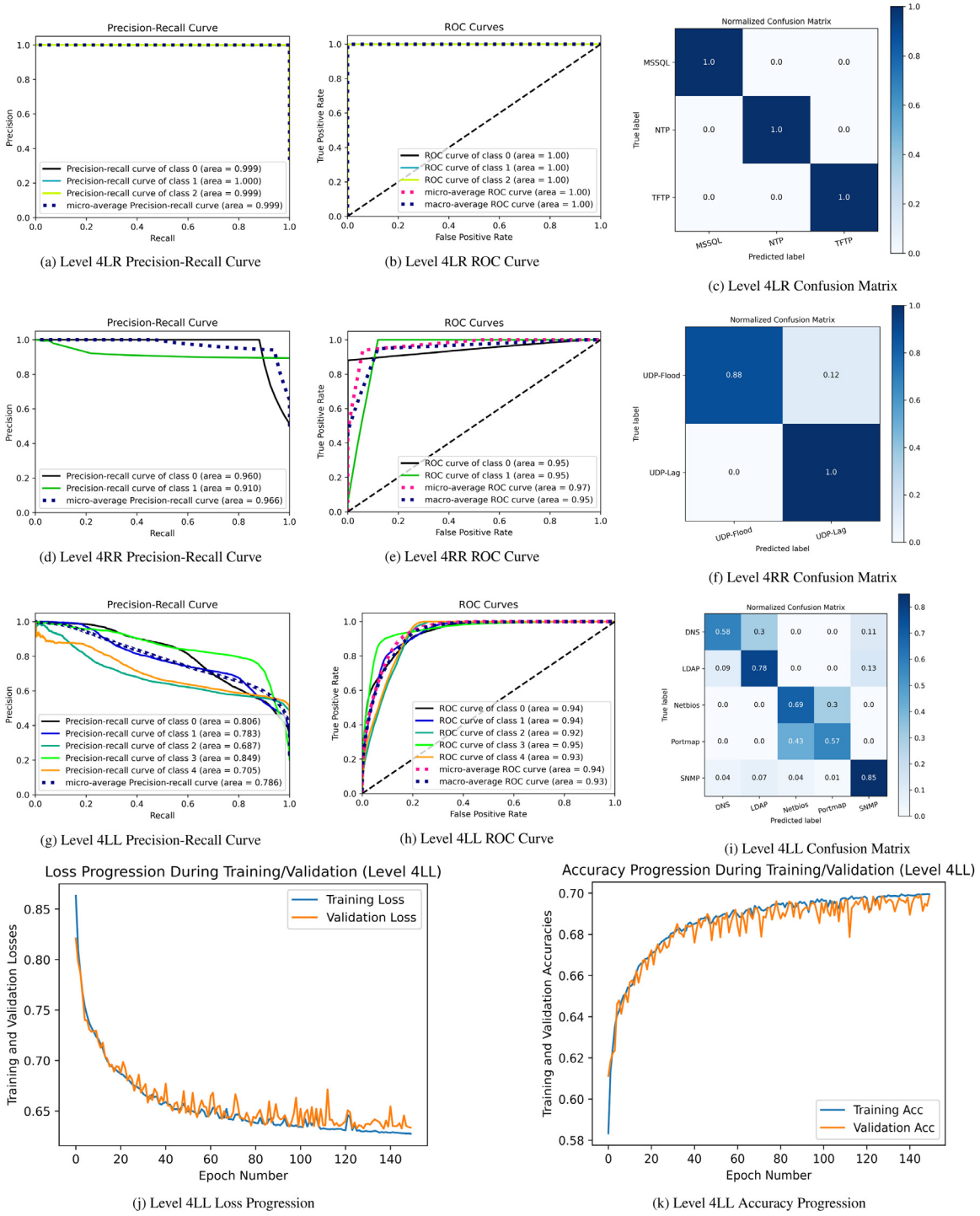
(j) Level 3R Precision-Recall Curve (k) Level 3R ROC Curve (l) Level 3R Confusion Matrix

Fig. 4. Precision-Recall, ROC Curves and Confusion Matrix for Level 1, 2, 3L and 3R.

minimize the data's dimensionality (Hotelling, 1933) and plotted the features in 3-dimensional space to get more insights about the data. It was observed that the five attack classes present in the level 4LL had multicollinearity present among them. This made generalizing capabilities of any machine learning or deep learning algorithm challenging for this level. Thus, a possible future improvement in this approach could be to classify these five attacks present in this level more accurately.

The true positive (TP) values, which reflect the number of packets correctly classified, are represented by the major diagonals in each confusion matrix. The Fig. 4 show that True Positive values are considerably higher than other values in the same rows and columns, indicating that the models are performing efficiently.

Our experiments led to a response time of 2.59 milliseconds when separating an attack packet from benign network packets. In a real-world scenario, the threats can mutate within minutes necessitating the



(a) Level 4LR Precision-Recall Curve (b) Level 4LR ROC Curve (c) Level 4LR Confusion Matrix

(d) Level 4RR Precision-Recall Curve (e) Level 4RR ROC Curve (f) Level 4RR Confusion Matrix

(g) Level 4LL Precision-Recall Curve (h) Level 4LL ROC Curve (i) Level 4LL Confusion Matrix

(j) Level 4LL Loss Progression (k) Level 4LL Accuracy Progression

Fig. 5. Precision-Recall, ROC Curves and Confusion Matrix for Level 4LR, 4RR and 4LL.

ability to quick and synchronous responses to threats and intrusion detection (Chatterjee, Kar, & Gupta, 2017). This can prove to be a substantial improvement over the existing systems, and for large scale information security systems.

The TaxoDaCML approach was also compared with the previous related works and Table 5 gives this comparison. It can be seen that this approach achieved best accuracy for attack detection. For attack classification, considering the large number of attack classes classified in this work, the accuracy can be seen at par or even above par than other works.

7.1. Contributions to literature

This research majorly extends the work of Sharafaldin et al. (2019) who have provided a dataset of modern-day DDoS attacks and proposed a taxonomy for the same. However, they achieved a relatively less accuracy for attack classification and a significant tradeoff between precision and recall. In detecting the attacks over IT infrastructures, the metrics such as precision and recall form a vital source of information to assess the performance. Using our approach that combines the taxonomy proposed by them with ML, the accuracy

Table 3
Performance of chosen algorithms for each problem at different levels.

Level & Algorithm selected	class	precision	recall	f1	acc
Level 1 (Decision Tree,criterion - entropy, random split)	Benign	0.999	0.999	0.999	0.999
	Attack	0.999	0.999	0.999	
Level 2 (Random Forest,criterion - entropy, n_estimators = 30)	Reflection	0.992	0.997	0.994	0.999
	Exploitation	0.997	0.992	0.994	
Level 3L (Decision Tree,criterion - entropy, best split)	TCP/UDP Group	0.997	0.998	0.998	0.998
	UDP Group	0.998	0.997	0.998	
Level 3R (KNN, k=7,distance_metric- manhattan)	TCP Group	0.999	0.999	0.999	0.999
	UDP Group	0.999	0.999	0.999	
Level 4LL(ANN,optimizer-adamax,lr=0.001,epochs=150)	DNS	0.809	0.584	0.678	0.698
	LDAP	0.685	0.777	0.728	
	NetBios	0.614	0.632	0.623	
	SNMP	0.772	0.854	0.811	
	Portmap	0.631	0.635	0.633	
Level 4LR (Random Forest,criterion - entropy,n_estimators = 10)	TFTP	0.998	0.999	0.998	0.999
	LDAP	0.999	0.999	0.999	
	MSSQL	0.999	0.999	0.998	
Level 4RR (Random Forest,criterion - entropy, n_estimators = 30)	UDP-Lag	0.999	0.881	0.937	0.941
	UDP-Flood	0.894	0.999	0.944	

Table 4
Final results obtained. P - Precision (in %), R - Recall (in %), F1 - F1 Score (in %), S - Support, T - Time to detect a single packet (in ms).

class	Run 1					Run 2					Run 3					Run 4				
	P	R	F1	S	T	P	R	F1	S	T	P	R	F1	S	T	P	R	F1	S	T
Benign	99	98	98	5652	2.68	98	99	98	5652	2.54	99	99	99	5652	2.64	99	99	99	5650	2.56
DNS	78	59	67	5500	44.64	77	60	67	5500	43.82	79	57	66	5500	43.61	79	58	67	5500	44.2
LDAP	69	75	72	5500	42.31	69	76	72	5500	43.45	70	75	72	5500	41.65	68	76	72	5500	41.98
NetBios	63	58	60	5500	47.61	63	57	60	5500	46.85	62	58	60	5500	46.53	63	57	60	5500	46.18
SNMP	76	84	80	5500	43.71	76	85	80	5500	42.8	77	83	80	5500	42.54	77	84	80	5500	42.19
Portmap	62	68	65	5500	44.27	63	67	65	5500	42.29	62	69	65	5500	42.43	61	69	65	5500	43.82
NTP	99	99	99	5500	13.90	99	99	99	5500	12.29	99	99	99	5500	12.52	99	99	99	5500	12.34
TFTP	99	99	99	5500	12.95	99	99	99	5500	12.36	99	99	99	5500	12.65	99	99	99	5500	12.01
MSSQL	97	98	97	5500	12.97	97	97	97	5500	13.08	97	98	97	5500	12.41	98	97	97	5500	12.39
SYN	98	99	98	5500	11.12	99	99	99	5500	10.51	99	99	99	5500	10.97	99	99	99	5500	10.63
UDP-Lag	99	87	93	5500	12.42	98	89	93	5500	12.82	99	86	92	5500	11.97	99	87	93	5500	12.09
UDP-Flood	88	98	93	5500	12.15	88	98	93	5500	11.63	89	97	93	5500	11.88	88	98	93	5500	11.34
accuracy			0.857					0.858					0.859					0.858		

Table 5
Comparison of TaxoDaCML with previous related works.

Approach/Reference	Datatset Used	Detection Accuracy	Classification Accuracy	No. of attack classes considered
TaxoDaCML	CICDDoS2019	99.9%	85.7%	11
(Sharafaldin et al., 2019)	CICDDoS2019	-	70%	12
(Das et al., 2019)	NSL-KDD	99.1%	-	-
(He et al., 2017)	Self-collected data	99.7%	-	-
(Hou et al., 2018)	ISP China Unicom	99.5%	-	-
DeepDefense (Yuan et al., 2017)	ISCX2012	98.4%	-	-
(Elsayed et al., 2020)	CICDDoS2019	99%	-	-
(Cil et al., 2021)	CICDDoS2019	99.9%	94.57%	2

increased to more than 85% and at the same time reduced the tradeoff between the precision and recall. Moreover, the time to detect a DDoS attack and computational cost remains in check with the help of our method.

In addition, the work by Jiao et al. (2017) mentions that the DDoS attacks executed by exploiting the TCP are on the rise. These attacks are capable of forming big threats by executing a denial of service to the information and data of millions of end-users. Our approach takes into consideration the various modern-day DDoS attacks carried over TCP as well and thus, reliably detects and classifies those attacks. Soomro et al. (2016) points out that human deficiencies in monitoring the network traffic lead to most data breaches and attack incidences in organizations. Reshmi (2021) states that anomaly based analysis for detecting malicious activity in the network is prone to high false positive rates and thus, a large number of normal traffic is classified as malicious. The author also states that machine learning is gaining popularity in detecting the malicious activities in the networks. Thus, by utilizing machine learning algorithms, we have automated the attacks' detection that has tackled the two main problems discussed in both the above works. Firstly, machine learning has reduced the human error and intervention. Secondly, the very high attack detection accuracy of our approach has substantially reduced the false positive rate.

Batra et al. (2021) indicate about the possibilities and potentials of using the hybrid and multi-algorithm systems as a part of their future directions. Inspired from this work, we have incorporated various ML algorithms in our framework to detect and classify DDoS attacks. Our system may also be termed as a hybrid system that combines ML with taxonomy of the DDoS attacks. Furthermore, some of the existing literature (Cil et al., 2021; Elsayed et al., 2020; He et al., 2017) propose solutions of DDoS classification, but only into a few broad high-level categories. This work extends it by further classifying into more exact DDoS attacks, thus improving the efficiency and reliability of the in-

formation security management systems detection system. At the same time, the bulky deep learning models that these works utilize leads to higher computation and decision-making time. Such issues are resolved in this work by the usage of lightweight models such as Decision Tree and Random Forests.

7.2. Practical implications

The TaxoDaCML technique for mitigating DDoS attacks has a wide range of applications in information management across a variety of disciplines. With the increasing use of IoT devices to construct smart cities, the threat posed by DDoS attacks is expected to exponentially rise. Cyber-criminals exploit these IoT devices by affecting them with malware. These devices act as botnets and facilitate DDoS attacks over major websites and infrastructure. The TaxoDaCML approach can thus be used to secure the IoT devices from being injected with malware.

Moreover, this approach can also be used to secure the Gaming industry which is majorly prone to DDoS attacks executed by players to gain a competitive advantage over each other. Players' data and achievements are compromised via such attempts.

The industries that provide financial and softwares as a service (SaaS) are also prone to DDoS attacks namely NTP, DNS, and SYN. Securing such industries and keeping their services up is of paramount importance. If the attackers are successful in breaching their infrastructures, it can lead to financial losses and harm the reputation and trust of the companies. The TaxoDacML approach is a reliable and robust solution to secure the infrastructure against DDoS attacks. Since the approach comprises multiple high-performing models at different levels, it can prove extremely effective to discern the intruders trying to harm the infrastructure. At the same time, the computational cost and time to detect a packet remain in check, decreasing the chances of false negatives. Moreover, the approach also classifies 11 major DDoS attacks. Since TaxoDaCML classifies 11 major attack types, it enables the industries to track the most exact type of DDoS attack which can further help them secure their IT infrastructure.

7.3. Future research directions

In future, this work may be extended to detect and classify attacks other than DDoS by devising specific taxonomies. Moreover, this may develop further into a complete Intrusion Detection System for detecting mutiple attacks like the Man-in-the-middle, Denial-of-Service, and IoT attacks. Protecting and securing large-scale information management systems from such attacks could be of utmost importance as we live in the era of data-driven decisions. As correctly indicated by Chatterjee & Kar (2018) that IoT enabled devices may infringe the privacy of the users, the future researchers may make use variety of ML algorithms in a network to protect these devices from hackers trying to steal the information. Morover, in the future it would also be worth working to solve the problem of multicollinearity discovered during the classification of Reflection-based DDoS attacks in Level 4LL.

Another future direction for proposed architecture is to train the models using a variety of datasets, thus providing a complete guard for the organizations' information security infrastructures. In this work, we have considered network analysis of two-day log files and in future, this work can be extended to consider real-time network packets for a more versatile and robust machine learning model.

8. Conclusion

This work extends the binary classification problem of detecting a DDoS attack to a multi classification problem of finding the protocol exploited by the DDoS attack. Knowing the exact protocol exploited by the attacker immensely helps to execute stronger security management measures and provide a quicker response to malicious packets coming into the network. By using the taxonomy to divide a bigger classification problem into seven smaller classification problems and then conquering them with machine learning algorithms, the TaxoDaCML approach obtains an aggregate accuracy of 85.8% for detecting and classifying the 11 prominent DDoS attacks. The results show that this approach helps in increasing the accuracy as compared to the previous approaches. Moreover, the approach uses a computationally light Decision Trees algorithm to detect a DDoS attack with a 99.9% recall, thus ensuring reliability in the attack detection without any delays. The attacks over Transport Layer protocols viz. either TCP or UDP or both are detected with the usage of taxonomy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarap, A. F. (2019). Deep learning using rectified linear units (ReLU). Accessed May. 11, 2021.

Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights, 1*(1), 100004. 10.1016/j.jjimei.2020.100004.

Alieyan, K., Kadhum, M. M., Anbar, M., Rehman, S. U., & Alajmi, N. K. (2016). An overview of DDoS attacks based on DNS. In *Proceedings of the international conference on information and communication technology convergence (ICTC)* (pp. 276–280). IEEE.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*, 175–185.

Arends, R., Austein, R., Larson, M., Massey, D., & Rose, S. (2005). DNS security introduction and requirements. *Technical Report, RFC 4033*. (Proposed Standard).

Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management, 45*, 289–307. 10.1016/j.ijinfomgt.2018.08.006.

Batra, J., Jain, R., Tikkiwal, V. A., & Chakraborty, A. (2021). A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques. *International Journal of Information Management Data Insights, 1*(1), 100006. 10.1016/j.jjimei.2020.100006.

Bogdanoski, M., Suminoski, T., & Risteski, A. (2013). Analysis of the SYN flood DoS attack. *International Journal of Computer Network and Information Security (IJCNIS), 5*(8), 1–11.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. In *Proceedings of the ECML PKDD workshop: Languages for data mining and machine learning* (pp. 108–122).

Chatterjee, S., & Kar, A. K. (2018). *Regulation and governance of the Internet of Things in India* (20, pp. 399–412). Digital Policy, Regulation and Governance. 10.1108/DPRG-04-2018-0017.

Chatterjee, S., Kar, A. K., Dwivedi, Y. K., & Kizgin, H. (2019). Prevention of cybercrimes in smart cities of India: from a citizen's perspective. *Information Technology & People, 32*, 1153–1183.

Chatterjee, S., Kar, A. K., & Gupta, M. (2017). Critical success factors to establish 5G network in smart cities: Inputs for security and privacy. *Journal of Global Information Management (JGIM), 25*(2), 15–37.

Chatterjee, S., Kar, A. K., & Gupta, M. P. (2018). Alignment of it authority and citizens of proposed smart cities in india: System security and privacy perspective. *Global Journal of Flexible Systems Management, 19*(1), 95–107. 10.1007/s40171-017-0173-5.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications, 169*, 114520. 10.1016/j.eswa.2020.114520.

Czyz, J., Kallitsis, M., Gharaibeh, M., Papadopoulos, C., Bailey, M., & Karir, M. (2014). Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks. In *Proceedings of the 2014 conference on internet measurement conference* (pp. 435–448).

Das, S., Mahfouz, A. M., Venugopal, D., & Shiva, S. (2019). Ddos intrusion detection through machine learning ensemble. In *Proceedings of the IEEE 19th international conference on software quality, reliability and security companion (qrs-c)* (pp. 471–477). 10.1109/QRS-C.2019.00090.

Das, S., Venugopal, D., Shiva, S., & Sheldon, F. T. (2020). Empirical evaluation of the ensemble framework for feature selection in DDoS attack. In *Proceedings of the 7th IEEE international conference on cyber security and cloud computing (CSCloud)/2020 6th IEEE international conference on edge computing and scalable cloud (EDGECOM)* (pp. 56–61). 10.1109/CSCloud-EdgeCom49738.2020.00019.

Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020). Ddosnet: A deep-learning model for detecting network attacks. In *Proceedings of the IEEE 21st international symposium on "a world of wireless, mobile and multimedia networks" (WOWMOM)* (pp. 391–396). 10.1109/WoWMoM49955.2020.00072.

Hare, C. (2002). What's not so simple about SNMP? *Information Security Management Handbook, Volume 4*.

Hassan, S.-U., Shabbir, M., Iqbal, S., Said, A., Kamiran, F., Nawaz, R., & Saif, U. (2021). Leveraging deep learning and SNA approaches for smart city policing in the developing world. *International Journal of Information Management, 56*, 102045. 10.1016/j.ijinfomgt.2019.102045.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Accessed May. 8, 2021.

He, Z., Zhang, T., & Lee, R. B. (2017). Machine learning based DDoS attack detection from source side in cloud. In *Proceedings of the IEEE 4th international conference on cyber security and cloud computing (CSCloud)* (pp. 114–120). 10.1109/CSCloud.2017.58.

Hefley, D. (2003). Udp port 1434 - services, vulnerabilities and exploits. Accessed May. 12, 2021.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America, 79*. 10.1073/pnas.79.8.2554. 2554–8

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, 24*, 498–520.

Hou, J., Fu, P., Cao, Z., & Xu, A. (2018). Machine learning based DDoS detection through netflow analysis. In *Proceedings of the Milcom 2018 IEEE military communications conference (MILCOM)* (pp. 1–6). 10.1109/MILCOM.2018.8599738.

Jiao, J., Ye, B., Zhao, Y., Stones, R. J., Wang, G., Liu, X., … Xie, G. (2017). Detecting TCP-based DDoS attacks in Baidu cloud computing data centers. In *Proceedings of the IEEE 36th symposium on reliable distributed systems (SRDS)* (pp. 256–258). 10.1109/SRDS.2017.37.

Kawamura, T., Fukushi, M., Hirano, Y., Fujita, Y., & Hamamoto, Y. (2017). An NTP-based detection module for DDoS attacks on IoT. In *Proceedings of the IEEE international conference on consumer electronics-Taiwan (ICCE-TW)* (pp. 15–16). IEEE.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR, abs/1412.6980*.

Kotey, S. D., Tchao, E. T., & Gadze, J. D. (2019). On distributed denial of service current defense schemes. *Technologies, 7*(1). 10.3390/technologies7010019.

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models (1st ed.)*. Chapman and Hall/CRC. 10.1201/9781315108230.

Kushwaha, A. K., Kar, A. K., & Dwivedi, Y. K. (2021). Applications of big data in emerging management disciplines: A literature review using text mining. *International Journal of Information Management Data Insights, 1*(2), 100017. 10.1016/j.jjimei.2021.100017.

Labs, B. L. (2015). A new DDoS reflection attack: Portmapper; An early warning to the industry. Accessed May. 12, 2021.

Lee, W. (Jun 29, 2020). DDoS protection for networks: Divert traffic using more specific routing. https://www.imperva.com/blog/ddos-protection-for-networks-divert-traffic-using-more-specific-routin/, Accessed May. 10, 2021.

Li, D., Deng, L., Lee, M., & Wang, H. (2019). IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *International Journal of Information Management, 49*, 533–545.

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2017). Large-margin softmax loss for convolutional neural networks. Accessed May. 10, 2021.

Wes McKinney (2010). Data structures for statistical computing in python. In Stéfan van der Walt, & Jarrod Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 56–61). 10.25080/Majora-92bf1922-00a.

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. In *Proceedings of the SIGCOMM computer communication review: 34* (pp. 39–53). 10.1145/997150.997156.

Mittal, A., Gupta, M., Chaturvedi, M., Chansarkar, S. R., & Gupta, S. (2021). Cybersecurity enhancement through blockchain training (CEBT) – a serious game approach. *International Journal of Information Management Data Insights, 1*(1), 100001. 10.1016/j.jjimei.2020.100001.

Newman, L. H. (2018). Github survived biggest DDoS attack ever recorded. https://www.wired.com/story/github-ddos-memcached, Accessed May. 15, 2021.

Patil, N. V., Rama Krishna, C., Kumar, K., & Behal, S. (2019). E-had: A distributed and collaborative detection framework for early detection of DDoS attacks. *Journal of King Saud University - Computer and Information Sciences*. 10.1016/j.jksuci.2019.06.016.

Reshmi, T. (2021). Information security breaches due to ransomware attacks-a systematic literature review. *International Journal of Information Management Data Insights, 1*(2), 100013.

Schwaderer, W. D. (1988). *Programmer's guide to NETBIOS*. SAMS.

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th python in science conference*.

Sen, S., Gupta, K. D., & Ahsan, M. M. (2020). Leveraging machine learning approach to setup software-defined network(SDN) controller rules during DDoS attack. In *Algorithms for intelligent systems* (pp. 49–60)). Springer, Singapore. 10.1007/978-981-13-7564-4_5.

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *Proceedings of the international Carnahan conference on security technology (iccst)* (pp. 1–8). 10.1109/CCST.2019.8888419.

Sieklik, B., Macfarlane, R., & Buchanan, W. J. (2016). Evaluation of TFTP DDoS amplification attack. *Computers & Security, 57*, 67–92.

Singh, A., & Juneja, D. (2010). Agent based preventive measure for UDP flood attack in DDoS attacks. *International Journal of Engineering Science and Technology, 2*(8), 3405–3411.

Singh, K., Guntuku, S. C., Thakur, A., & Hota, C. (2014). Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences, 278*, 488–497. 10.1016/j.ins.2014.03.066.

Singhal, H., & Kar, A. K. (2015). Information security concerns in digital services: Literature review and a multi-stakeholder approach. In *Proceedings of the international conference on advances in computing, communications and informatics (ICACCI)* (pp. 901–906). 10.1109/ICACCI.2015.7275725.

Sollins, K. (1992). The TFTP protocol (revision 2). *Technical Report*. STD 33, RFC 1350, MIT.

Soomro, Z. A., Shah, M. H., & Ahmed, J. (2016). Information security management needs more holistic approach: A literature review. *International Journal of Information Management, 36*(2), 215–225.

Srinivasan, R. (1995). Binding protocols for ONC RPC version 2. *Technical Report*. RFC 1833, August.

pandas development team, T. (2020). Pandas-dev/Pandas: Pandas. 10.5281/zenodo.3509134

Thielman, S., & Hunt, E. (2016). Cyber attack: Hackers 'weaponised' everyday devices with malware,. https://www.theguardian.com/technology/2016/oct/22/cyber-attack-hackers-weaponised-everyday-devices-with-malware-to-mount-assault.

Thomas, S. (1987). Netbios for ISO networks. In *Proceedings of the ACM SIGCOMM computer communication review: 17* (pp. 21–29).

Tin Kam Ho (1995). Random decision forests. In *Proceedings of the 3rd international conference on document analysis and recognition: vol. 1* (pp. 278–282 vol.1). 10.1109/ICDAR.1995.598994.

Tuttle, S., Ehlenberger, A., Gorthi, R., Leiserson, J., Macbeth, R., Owen, N., … Yang, C. (2004). *Understanding Ldap - design and implementation*. IBM Redbooks.

Wang, M., Lu, Y., & Qin, J. (2020). A dynamic MLP-based DDoS attack detection method using feature selection and feedback. *Computers & Security, 88*, 101645. 10.1016/j.cose.2019.101645.

Xiaoming, L., Sejdini, V., & Chowdhury, H. (2010). *Denial of service (DoS) attack with UDP flood*. School of Computer Science, University of Windsor, Canada.

Yuan, X., Li, C., & Li, X. (2017). Deepdefense: Identifying DDoS attack via deep learning. In *Proceedings of the IEEE international conference on smart computing (SMARTCOMP)* (pp. 1–8). 10.1109/SMARTCOMP.2017.7946998.

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys Tutorials, 15*(4), 2046–2069. 10.1109/SURV.2013.031413.00127.