EURASIP Journal on Information Security

RESEARCH

Open Access

Peer-to-peer botnets: exploring behavioural characteristics and machine/deep learning-based detection

Arkan Hammoodi Hasan Kabla[1], Achmad Husni Thamrin[2], Mohammed Anbar[1], Selvakumar Manickam[1] and Shankar Karuppayah[1*]

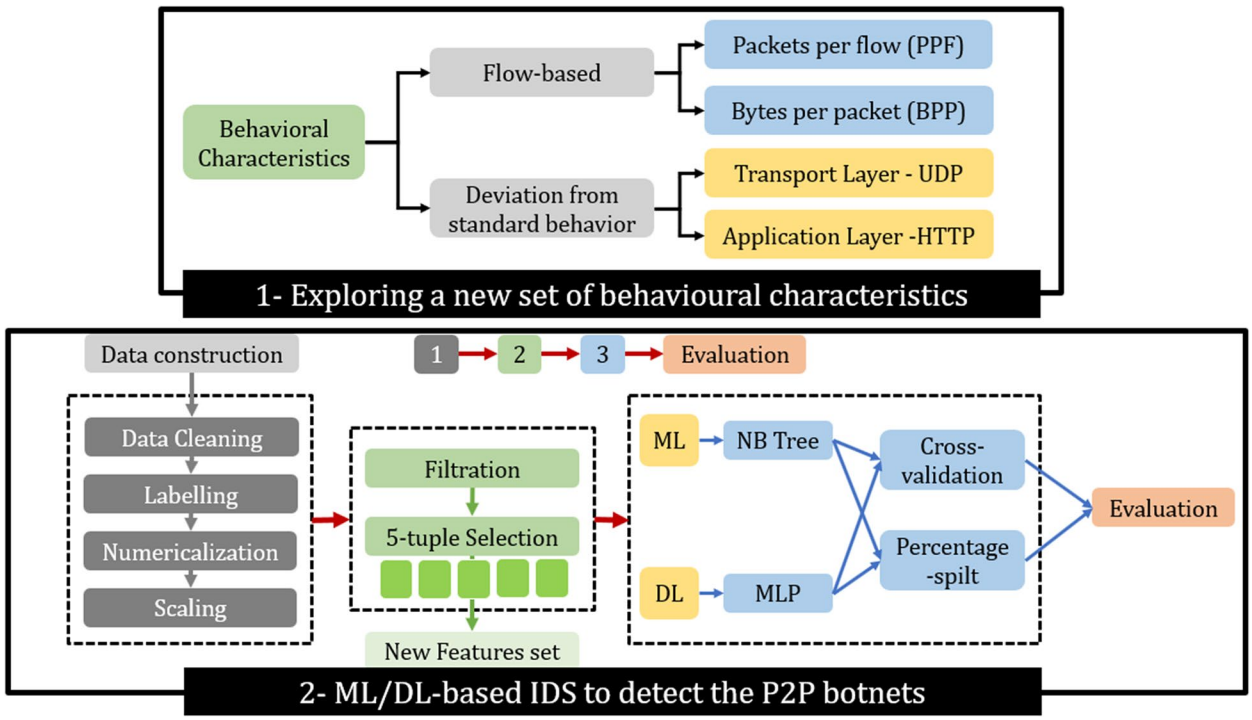
Abstract

The orientation of emerging technologies on the Internet is moving toward decentralisation. Botnets have always been one of the biggest threats to Internet security, and botmasters have adopted the robust concept of decentralisation to develop and improve peer-to-peer botnet tactics. This makes the botnets cleverer and more artful, although bots under the same botnet have symmetrical behaviour, which is what makes them detectable. However, the literature indicates that the last decade has lacked research that explores new behavioural characteristics that could be used to identify peer-to-peer botnets. For the abovementioned reasons, in this study, we propose new two methods to detect peer-to-peer botnets: first, we explored a new set of behavioural characteristics based on network traffic flow analyses that allow network administrators to more easily recognise a botnet's presence, and second, we developed a new anomaly detection approach by adopting machine-learning and deep-learning techniques that have not yet been leveraged to detect peer-to-peer botnets using only the five-tuple static indicators as selected features. The experimental analyses revealed new and important behavioural characteristics that can be used to identify peer-to-peer botnets, whereas the experimental results for the detection approach showed a high detection accuracy of 99.99% with no false alarms.

Keywords P2P botnets, Network traffic analysis, Intrusion detection system, Anomaly detection, Machine learning, Deep learning

*Correspondence:
Shankar Karuppayah
kshankar@usm.my
Full list of author information is available at the end of the article

Graphical Abstract



1 Introduction

The term "bot" refers to a compromised machine under the command of a botmaster, whereas the term "botnet" refers to a network of such compromised machines [1]. Typically, bots are exploited to perform various attacks, such as stealing data, launching distributed denial of service (DDoS) attacks, phishing, and spam [2]. Recently, botnets have led to huge threats to Internet infrastructure security in different scenarios. Therefore, managing and improving network security have become more challenging, especially since the attackers are also improving their tactics and capabilities to avoid the existing countermeasures against them. In the last decade, botmasters developed their tactics well by benefiting from several robust concepts, such as decentralisation [3]. The concept of decentralisation has been used to solve many of the biggest problems related to the Internet's network infrastructure, such as the single point of failure problem. However, it also brought new challenges when illegal intruders utilised the same strong points against the original purposes of those points. For example, peer-to-peer (P2P) botnets have been observed to adopt the P2P architecture, and these botnets are characterised by dispersion and distribution [4, 5]. Figure 1 shows the difference between the P2P botnets on the left side and the centralised botnets on the right side.

In addition, P2P botnets have no independent botnet mainframe, which eliminates the vulnerabilities that weaken other architectures [6]. Furthermore, P2P botnets are more resilient and stealthier than other types of botnets, which is another reason why they are very difficult to defeat or detect [7].

However, there are still several security countermeasures for botnets, and each countermeasure thwarts the botnets differently. For example, botnet monitoring provides information about most bots using monitoring mechanisms, such as honeypots, crawlers, and sensors [4]. These mechanisms assist to more behavioural understanding and analysis. Consequently, that leads to identify the botnets' characteristics and behaviours in the networks. Another effective security countermeasure is the intrusion detection and prevention system (IDPS); the purpose of these systems is to monitor network traffic to detect unauthorised access and take procedures to prevent it [8]. There are two main types of intrusion detection systems: anomaly based and signature based. The first type detects abnormal traffic based on deviations from the normal network traffic. The second type defines certain misbehaviours or signatures and then

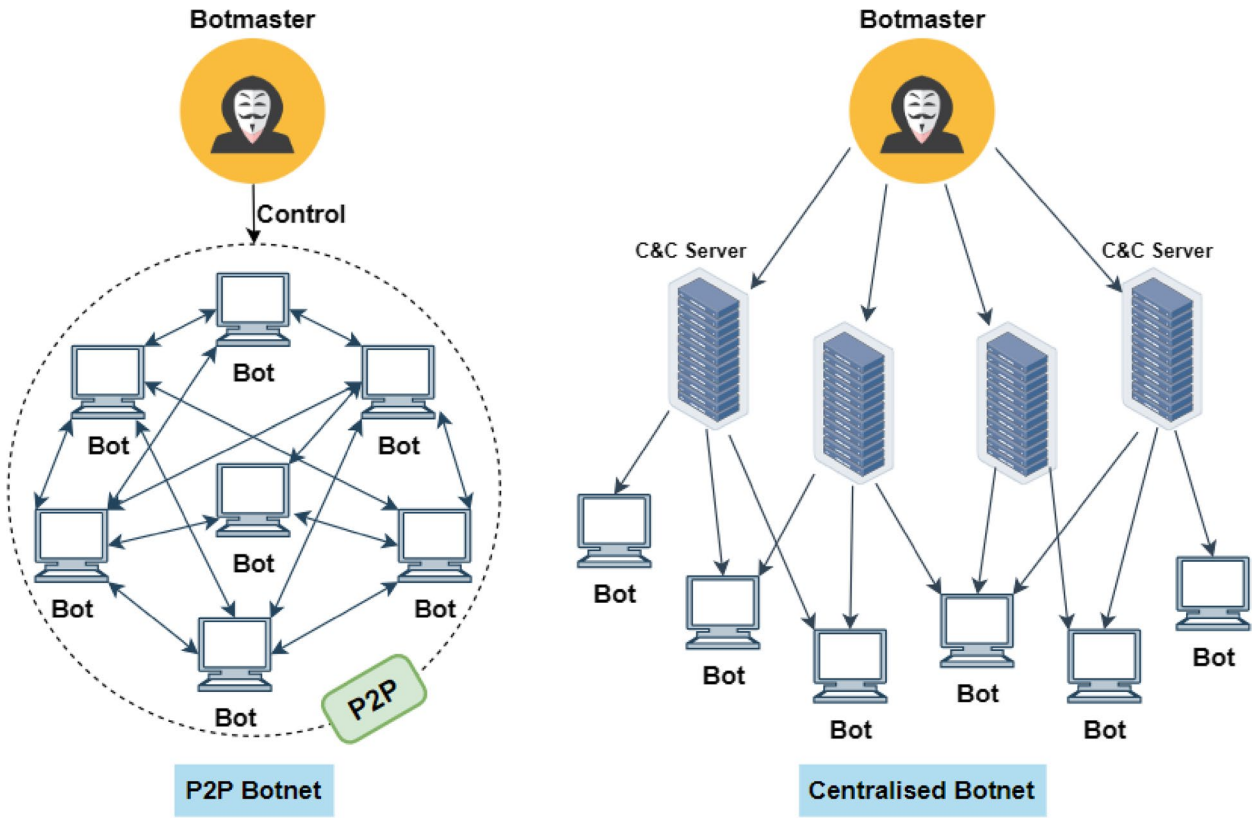


Fig. 1 Centralised botnet vs. P2P botnets

detects them once they happen. In terms of the location model, there are two main types of IDSs: host-based IDSs reside in the host, and network-based IDSs reside across the whole network [9, 10]. However, they are not foolproof and may not catch all the botnet instances especially botnets are constantly evolving and that what makes it challenging for even the effective IDPSs to keep up with the new tactics of botnets.

Although IDPSs are valuable security countermeasures, they are not a panacea, and they should be supplemented with other security practices to effectively mitigate the cyberthreats such botnets. This work aims to fill this gap by exploring more behavioural characteristics of one of the most serious and modern threats which P2P botnets. This paper proposes a new method of network traffic analysis that assists to identify new behavioural indicators of P2P botnets in the network. This method categorises the behavioural characteristics into two categories: (i) flow based has two norms to measure the packets per flow (PPF) and bytes per packet (BPP) as indicators and (ii) deviation from standard behaviour to measure the behavioural deviation from the transport layer and application layer as another indicator. Practically, these artefacts can be used as indicators of compromise (IOC) that can be leveraged by the network administrator to secure their networks from such threat.

Furthermore, this paper also proposes a novel approach to detect the P2P botnets using machine learning (ML) and deep learning (DL) techniques. The proposed approach utilises only the static indicators (five-tuple) including source and destination IP addresses, source and destination port numbers, and protocol identify number, as selected features.

In summary, this paper presents two security countermeasures for P2P botnets. First, we explore new behavioural characteristics/dynamic indicators (also known as IOCs) for P2P botnets to enable network administrators to distinguish the P2P traffic crossing the network boundaries via a network traffic analysis. Second, we utilise the static indicators (five-tuple) to detect P2P botnets using ML/DL techniques that have not yet been leveraged. For evaluation purpose, we utilise a recently published dataset by Kable et al. [11] that contained the P2P botnet scenario. To summarise, our contributions in this paper are as follows:

- Proposing a new method based on analysing the network traffic flow and deviation from the standard

protocols to identify the behavioural characteristics of P2P botnets
- Investing the newly discovered behavioural characteristics as IOCs to detect the P2P botnets
- Adopting ML/DL techniques to detect the P2P botnets using only the five-tuple static indicators

The paper is organised as follows. Section 2 conducts a comprehensive review of the related works and summarises the state of the art of ML/DL-based solutions. Section 3 defines the dataset used in this work. Section 4 lists the implementation prerequisites of this work. Section 5 describes the newly proposed method of exploring a new set of behavioural characteristics of P2P botnets. Section 6 presents the ML/DL-based proposed approach to detect the P2P botnets. Finally, Sect. 7 concludes this work and provide multiple future works.

2 Related works

The connection between compromised machines and the command and control (C&C) servers is an inevitable operation needed to call commands and updates. Consequently, some indicators always lead to the recognition of the botnets in a network [12]. In this section, we comprehensively review related work meant to identify botnet behaviour by analysing network traffic. Furthermore, this section also summarises the related works that have proposed IDSs to specifically detect P2P botnets, in a table at the end of this section (Table 1). Most of the effective IDSs proposed by related works are based on ML/DL techniques. In brief, ML and DL are subfields of artificial intelligence (AI), which can be defined as the capability of machines to learn and imitate intelligent human behaviour [13–15]. In addition, ML and DL techniques have shown promise as effective and efficient mechanisms for detecting anomalous behaviour [15, 16].

Lee et al. [17] used the degree of periodic repeatability to distinguish between malicious HTTP bots and benign nodes. The authors considered the repeatability standard deviation in the detection of HTTP botnets as the degree of periodic repeatability. The results showed that the flows from benign nodes and HTTP bots were distinguishable. However, this paper only dealt with a sample of malicious HTTP botnets, with the only feature vector being the degree of periodic repeatability, i.e. the authors only looked for malicious HTTP botnets by monitoring the relations between the HTTP servers and bots.

Strayer et al. [18] examined flow characteristics, such as the packet timing, burst duration, and bandwidth, and then considered various indicators as evidence of the existence of botnet command and control. The authors started by eliminating the traffic that was unlikely to represent the activity of a botnet. They then classified the remaining traffic into groups that were likely to represent botnet activities. Furthermore, the authors correlated the likely traffic to determine the common communication patterns used by the botnet activities. Ultimately, the authors showed that the evidence for botnets could be extracted from traffic traces. However, they only practically evaluated their work with IRC commands.

W. Lu et al. [19] presented a classification approach for the detection of botnets. The authors evaluated the proposed framework using the web and the IRC community; the evaluation results showed a high detection rate with a low false alarm rate. In addition, the authors formalized the botnet behaviour using the average standard deviation for the byte frequency (over 256 ASCII characters in the traffic payload). Then, they provided a botnet strategy, whereby a higher average deviation value represented a higher likelihood that the traffic was generated by human beings. This indication strategy is important when using unsupervised learning (e.g. clustering) to detect botnets. However, this approach requires a large number of bots in the network, and, intuitively, it is inefficient when there is a small-scale botnet.

Venkatesh et al. [20] proposed a method to detect HTTP-based botnets using the behaviour of bots in the network. The authors discovered that most web-based botnets' communications exploit TCP connections. The behaviours of the TCP connections were extracted as selected features to detect HTTP-based botnets using ML techniques, such as neural networks. This method demonstrated the capability to detect HTTP-based botnets with a high detection rate and low false alarm rate. However, the authors only evaluated the proposed method by using the Zeus and SpyEye bots, and both these bots are similar in their behaviour in network traffic.

G. Gu et al. [21] proposed a detection system based on the protocol and structure used by botnets. This system exploits the properties of botnets, as bots of each botnet utilise the same C&C communications, i.e. they have similar malicious behaviours.

Wang et al. [22] presented an approach for detecting web-based C&C bots by identifying their network behaviour in a supervised network. Modelling the essential network behaviour showed that the approach could be used to detect web-based C&C bots with a low false-positive rate. The authors noticed that the bots under the same botnet had similar connections when carrying out C&C communication. They therefore aimed to extract the common network behaviours used by web-based bots in order to automate the detection model. However, the authors neither consider group activities nor the payload information.

Table 1 Summary of related works

Article	Technique	ML/DL	Findings
[38]	SVM	ML	• Proposed an approach that achieved low misclassification when detecting three types of P2P botnets
			• The hosts used were running limited P2P applications, mostly Skype traffic
[39]	Nearest neighbour, Naive Bayes, J48	ML	• The authors experimented with different ML techniques for the detection of P2P botnets and compared their abilities in classifying this kind of botnet
			• Detection of legitimate traffic was very weak
[33]	Hierarchical clustering dendrogram	ML	• Detection of P2P botnets by discovering flow dependencies
			• The proposed approach could not detect the botnets that have irregularities in their traffic flow, such as storm, because the method was built based on the similarity of botnet traffic
[34]	Bayesian networks, Naive Bayes, J48	ML	• Proposed a methodology to detect P2P botnets using ML techniques and achieved a high detection rate
			• Research was only conducted for the LAN environment
[35]	Decision tree	ML	• Proposed a P2P detecting system involving the identification of malicious fast-flux networks
			• The system is based on low time to live; when the TTL reaches zero, packets are discarded. This leads to loss of some of the network information
[12]	Neural network	DL	• Based on a multilayer NN, the proposed method achieved a high detection rate of 99%
[6]	K-nearest, REP tree, SVM	ML	• Proposed a new feature extraction method using the graphic symmetry concept to detect P2P botnets
[36]	Decision tree	ML	• Proposed an approach based on an ML classifier to detect P2P botnets at the node level
			• Storage overheads and major computational resources were required to process the constant flows at the node without even feature engineering
[32]	MultiBoostAB, DecisionStump	ML	• Detection of parasite P2P botnets using machine-learning classifiers
			• The authors used the same dataset [31], which is small and limited in terms of the traffic type
[37]	DGA	Neither	• Development of a beneficial botnet as an anti-botnet measure
			• Use of the beneficial botnet to detect P2P communication by malware
[27]	Deep neural network	DL	• Proposing a new deep neural network-based approach to detect the P2P botnet using minimum number of features compared to other related works
[28]	Random forest, KNN, Naïve Bayes, SVM, decision tree	ML	• Proposing a Hadoop-based P2P botnet detection system to detect P2P botnet in local area network (LAN)
			• This paper introduced some of compromise indicators such count of unique destination hosts connected, total amount of data transferred from the source host, average of the TTL value of the packets transferred from the source host, and count of unique destination ports connected
[29]	SVM, K-means, decision tree, logistic regression	ML	• The authors experimentally examined some of feature selection algorithms to identify the most significant set of features
			• The authors applied four machine learning algorithms to detect the P2P botnet
[30]	ResNet convolution neural network	DL	• Proposing a ResNet CNN-based model to detect the P2P botnet by extracting important features from the traffic data. The idea of ResNet is to integrate the local connection and weight sharing in order to solve the problem of gradient explosion

Eslahi et al. [23] proposed low-access-rate and high-access-rate filters; these filters reduced the false-positive rate in HTTP-based botnet detection. The high-access-rate filter was proposed based on the fact that botnets do not generate bulk data. Therefore, this filter was designed to remove any traffic that generates a high rate of requests. Later, those high-rate requests are labelled as automatic software rather than bot communications. The low-access-rate filter ignores the traffic that appears to be low as bots are created to perform faster than humans, as well as to undertake larger tasks, i.e. bots do not generate brief traffic.

Table 2 Summary of the botnet datasets

Dataset	Description
P2P b**otnet** dataset — **PeerAmbush** [11]	The latest dataset that was published and contain P2P botnet scenario. This dataset is well-constructed, and it was used in a research to detect the P2P botnet using deep learning technique [11]
DCNDS [41]	Project dataset including a P2P botnet scenario. This dataset does not contain background flows, and no PCAP file is provided
CTU-13 [42]	Includes 13 scenarios with different botnet samples, such as the P2P botnet. Many protocols are considered, such as ICMP, TCP, and DNS. However, this dataset does not contain background flows
VHS-22 [43]	A CSV file that contains mixed flows of botnets from other datasets, such as ISOT, CICIDS, CTU-13, and MTA, with legitimate traffic
MTA-KDD-19 [44]	Malware Traffic Analysis Knowledge Dataset. However, only a small CSV file is provided
Trend Micro [45]	CTF Wildcard botnet dataset 400. It contains only the following features: timestamp, source, destination, port, and bytes. It is provided as a CSV file
P2P-BDS [46]	Based on the article "Peer-2-Peer botnet detection system", but it is no longer reachable
ISOR [47]	Based on [47], but it is no longer reachable
ISOT [48]	Botnet dataset that only contains traffic passed from/to DNS

Jang et al. [24] studied how to evade detection methods, and analysing the evasion technique was intended to contribute to detecting botnets.

AlAwadi et al. [25] proposed a multi-phase IRC botnet behaviour detection model. The authors used the C&C response messages and the malicious behaviours of IRC bots to identify botnets in the network environment.

Rostami et al. [26] provided an overview of the features and parameters utilised to detect HTTP botnets in order to propose a set of characteristics for the HTTP protocol that could be used to analyse and detect botnets. The authors presented various HTTP protocol attributes in order to facilitate better understanding and classification of HTTP packets, such as GET, POST, and the user agent.

As earlier stated, this section ends with a summary of related works that proposed IDS as a solution to detect P2P botnets [6, 12, 21, 27–37]. Table 1 summarises the related studies that proposed IDSs to specifically detect P2P botnets.

In sum, botnets quickly upgrade their functionalities and improve their methods to evade detection techniques. Consequently, the periodic tasks with C&C servers and the packet size can change, which can defeat current botnet detection systems based on these features. Therefore, studying other attributes based on traffic analyses might help to develop new indicators that can facilitate botnet detection by network administrators.

3 Dataset definition

For many reasons, such as privacy considerations, obtaining a real network dataset is difficult. We can see that most existing datasets are simulation-based datasets. We were not concerned about whether the dataset used here was a real network or a simulation-based one, but we were concerned about the method of construction. Thorough and adequate dataset construction is important since new IDSs should be evaluated before deployment in real networks using a robust dataset. Issues in the datasets may even be reflected in the final evaluation [40].

We comprehensively studied the existing datasets, and each one was found to have its limitations: some were small-size datasets, some were unknown-source datasets, and some were datasets that were no longer reachable. Table 2 summarises the information about the existing datasets that contain P2P botnet traffic flows.

The issue with most of the existing datasets is that they are incomplete datasets. For detection purposes, the dataset must contain attack traffic mixed with background traffic in order to allow the trained model to learn more about both normal and abnormal behaviour. For example, the CTU-13 dataset is the most widely used compared to others (for instance, Xing et al., 2022) because it is a reliable and well-constructed dataset. However, after we experimentally analysed this dataset, we found that no benign traffic was recorded from non-infected machines, i.e. once we blocked the IP addresses of the botmaster and the infected machines, no traffic was left. There was only one dataset that has a traffic contained of both P2P botnets and benign nodes which was published by Kabla et al. [11].

Another important point is that most of the datasets are provided as CSV files, and we counted this as a limitation since CSV files only reflect a limited image of network traffic. In addition, flow-based behavioural indicators and bias standard behavioural indicators cannot be derived from CSV files, but PCAP files give complete network information, allowing better understanding when devising new IOCs.

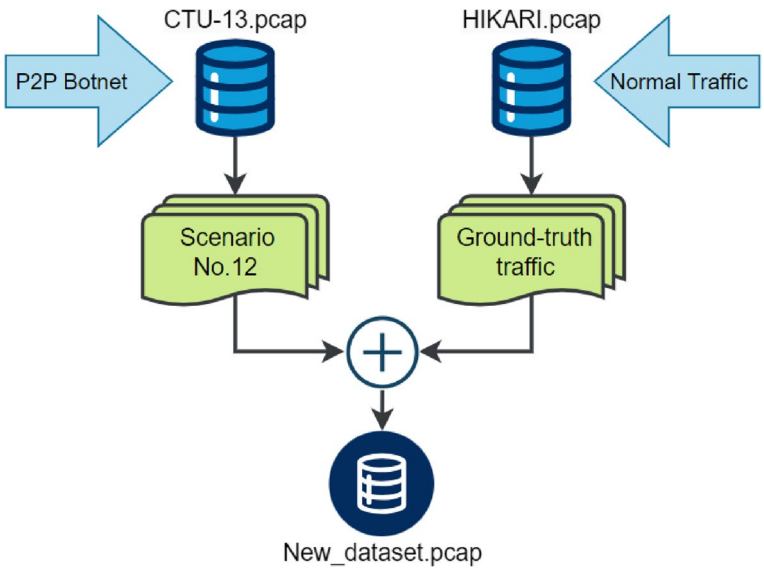


Fig. 2 Data construction process of the selected dataset

Table 3 Description of the selected dataset

Total number of records	**886,114**
Category	Multi-class
Classes	Botmaster, bot, normal
Number of botmaster/bot records	352,266
Number of normal records	533,848
Number of features	30

Given the above reasons, we selected the P2P botnet dataset (PeerAmbush) [11] to evaluate the two proposed methods. The selected dataset is available for other researchers at Kaggle,[1] namely: P2P botnet dataset — PeerAbmush.[2] Figure 2 shows the dataset construction process of the selected dataset [11].

The selected dataset was completed by including the traffic flows of the botmaster, bots/infected machines, and noninfected machines. Then, the selected dataset is used for two purposes: to explore a new set of behavioural characteristics for a P2P botnet and to train a detection model using the static indicators. Table 3 describes the selected dataset.

[1] Kaggle.com

[2] https://www.kaggle.com/datasets/arkantaha/p2p-botnet-dataset-peerabmush

Table 4 Hardware specifications

Specification	**Capacity**
CPU	AMD Ryzen 9 4900H
Memory (RAM)	16.0 GB
Hard drive	1 TB

Table 5 Software specifications

	Item	**Version**
Operating System	Windows enterprise	10 (64-bit)
Software	Python programming language	3.10.11
	Wireshark	3.6.6
	Microsoft Office	18
	WEKA	3.8.6
	Java	8

4 Implementation prerequisites

The implementation prerequisites include programming languages and software tools to experimentally implement the proposed methods as follows: (i) identify the behavioural characteristics of P2P botnets by analysing the network traffic flow, and (ii) detect the P2P botnets using ML/DL techniques. Tables 4 and 5 list the hardware and software specifications used, respectively.

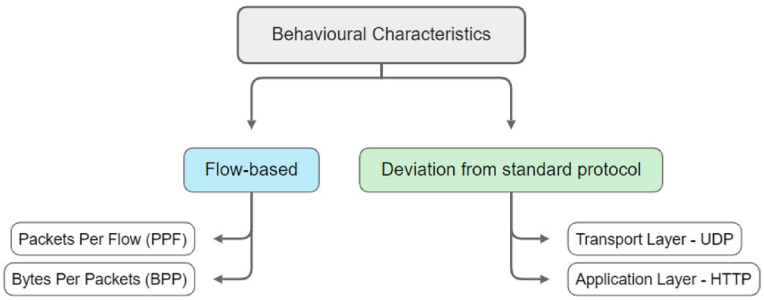


Fig. 3 The behavioural characteristics

5 Behavioural characteristics of peer-to-peer botnets

Typically, the main part of a botnet is the C&C channel. When we analysed network traffic, the behavioural indicators of C&C were also analysed. There may be some common features among the bots in network traffic, such as when botmasters are directly or indirectly informed about botnet detection or analysis activities. In addition, botmasters are required to periodically update the bots, which forces them to find a means of communication that, in the end, will be evidence of their presence. This kind of bot activity makes them recognisable and detectable. However, large-scale networks with extensive Internet bandwidth and administrative restrictions make it harder to monitor the whole network and accurately detect intrusions. Thus, this paper presents a new set of behavioural characteristics that can be used as IOCs to recognise the presence of P2P botnets in a network environment.

Unlike packet-based analysis, the behaviour level is related to higher-level features that are extracted from the traffic flow in order to help the network administrator recognise P2P botnets. In this study, we categorised the behavioural characteristics into flow-based characteristics and deviations from the standard behaviour of the network protocols. Noticeably, the experimental findings indicated deviations from standard behaviour in the transport layer (UDP) and the application layer (HTTP). Figure 3 summarises the categorisation of behavioural characteristics in this paper.

In other words, we depended on behaviour analysis and recognition using the standard protocol behaviours (i.e. the dynamic indicators), disregarding the port-based analysis undertaken by some researchers because there would be high false-positive rates. The reason behind high false-identification rates is that thousands of network applications do not use the registered TCP/UDP ports nowadays [49].

On the other hand, despite each botnet implementing its own C&C mechanism, such mechanisms exhibit distinguishable behaviours that can be captured by analysing the network behavioural indicators, allowing the network administrator to recognise anomalies easily. Furthermore, partially matching behaviours occur regularly in the lifetimes of botnets, which is another factor that makes it possible to capture them. For example, the botmaster may distribute scripts that automatically execute when certain events happen, such as new bots joining the botnet.

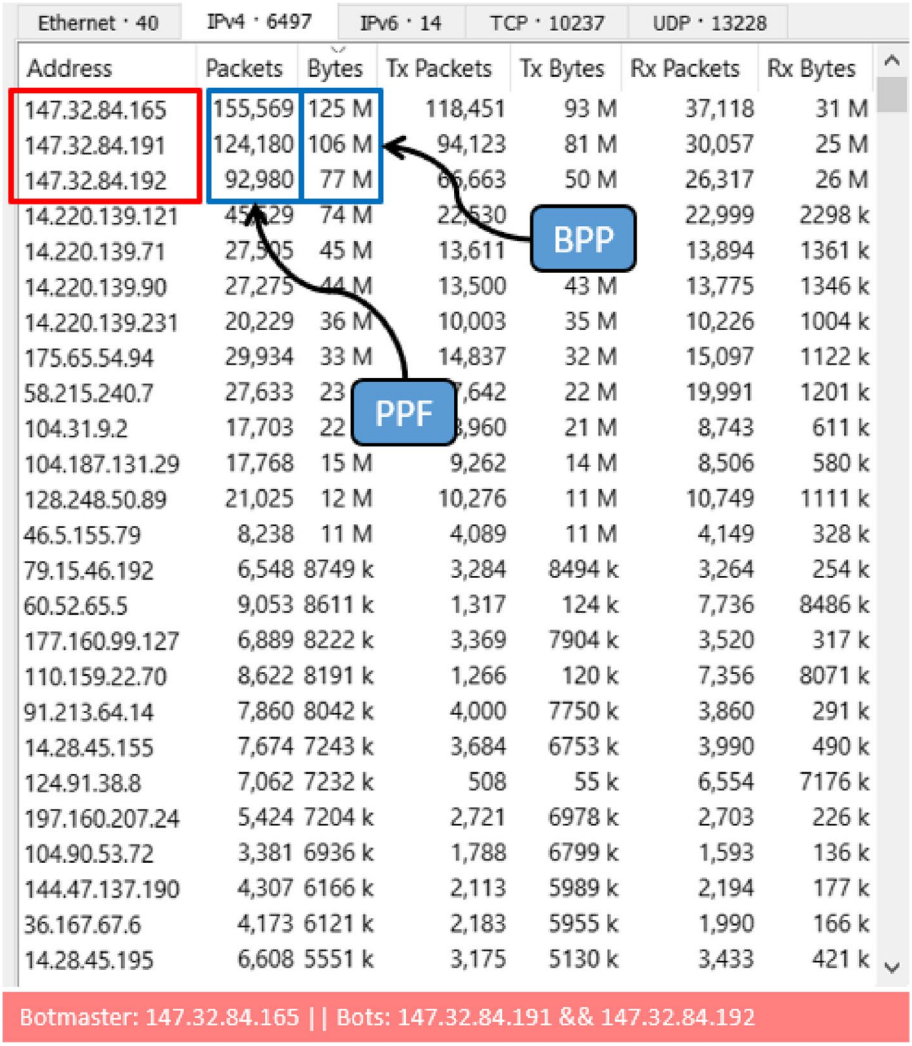
5.1 Flow-based behavioural characteristics

This category involved classifying distinctive network traffic behaviours as indicators of anomalies or benign node traffic. The analysis was based on the flow; a flow is a set of packets that belong to the same instance of communication with an application at the source and destination hosts. One of the most common ways of identifying a particular UDP or transmission control protocol (TCP) flow is by using the five-tuple features: source IP address, destination IP address, source port number, destination port number, and protocol identifier number [50]. The items in the five-tuple were used as static indicators to detect P2P botnets using ML/DL techniques (Sect. 6) in order to show how indicative these static indicators are in the detection of botnets. Nevertheless, no related work has yet leveraged the five-tuple for detection purposes.

However, to uniquely identify a flow, we must define it as something altogether different. Moreover, this analysis can work with encrypted traffic because it does not rely on the packet payload.

Flow-based indicators fall into two types: static indicators, which are not changeable over the flow's lifetime, and dynamic indicators, which are changeable as the flow progresses through time. As is known, the immutable information in the IP and TCP/UDP headers is a significant source of statistical indicators (Sect. 6 describes P2P botnet detection using static indicators). The static indicators include five-tuple values (as mentioned above).

Likewise, some dynamic indicators, such as the packet size values, may also be derived from the payload



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
147.32.84.165	155,569	125 M	118,451	93 M	37,118	31 M
147.32.84.191	124,180	106 M	94,123	81 M	30,057	25 M
147.32.84.192	92,980	77 M	66,663	50 M	26,317	26 M
14.220.139.121	45,529	74 M	22,530		22,999	2298 k
14.220.139.71	27,505	45 M	13,611		13,894	1361 k
14.220.139.90	27,275	44 M	13,500	43 M	13,775	1346 k
14.220.139.231	20,229	36 M	10,003	35 M	10,226	1004 k
175.65.54.94	29,934	33 M	14,837	32 M	15,097	1122 k
58.215.240.7	27,633	23	7,642	22 M	19,991	1201 k
104.31.9.2	17,703	22	8,960	21 M	8,743	611 k
104.187.131.29	17,768	15 M	9,262	14 M	8,506	580 k
128.248.50.89	21,025	12 M	10,276	11 M	10,749	1111 k
46.5.155.79	8,238	11 M	4,089	11 M	4,149	328 k
79.15.46.192	6,548	8749 k	3,284	8494 k	3,264	254 k
60.52.65.5	9,053	8611 k	1,317	124 k	7,736	8486 k
177.160.99.127	6,889	8222 k	3,369	7904 k	3,520	317 k
110.159.22.70	8,622	8191 k	1,266	120 k	7,356	8071 k
91.213.64.14	7,860	8042 k	4,000	7750 k	3,860	291 k
14.28.45.155	7,674	7243 k	3,684	6753 k	3,990	490 k
124.91.38.8	7,062	7232 k	508	55 k	6,554	7176 k
197.160.207.24	5,424	7204 k	2,721	6978 k	2,703	226 k
104.90.53.72	3,381	6936 k	1,788	6799 k	1,593	136 k
144.47.137.190	4,307	6166 k	2,113	5989 k	2,194	177 k
36.167.67.6	4,173	6121 k	2,183	5955 k	1,990	166 k
14.28.45.195	6,608	5551 k	3,175	5130 k	3,433	421 k

Fig. 4 PPF and BPP indicators

information and packet header. In contrast, the packet arrival and departure times represent dynamic indicators, but they are outside the packet. Further dynamic indicators can be derived, such as burst times, periodic throughput samples, and bytes per burst.

In our experimental analysis, we depend on two new and important indicators to distinguish the behavioural characteristics: packets per flow (PPF) and bytes per packet (BPP).

5.1.1 Packets per flow (PPF)

The PPF refers to how many packets uniquely represent a single flow. The PPF revealed that the greatest numbers of packets were transmitted (Tx packets) and received (Rx packets) by the botmaster IP in the first place and to/from the infected machine in the second place, as shown in the screenshot in Fig. 4.

5.1.2 Bytes per packets (BPP)

In the same way, the BPP revealed that the volume of data (Tx bytes, Rx bytes) sent to/from the botmaster was the greatest, followed by that to/from the infected machines, as shown in the screenshot in Fig. 4. The IP addresses of the botmaster and the bots are listed below the screenshot in Fig. 4.

5.2 Deviation from standard behavioural indicators of the protocols

The analysis of deviations from standard behavioural indicators is also known as protocol-based analysis. This analysis is based directly on the packet's payload. This analysis has a low false-positive rate compared to other analyses; thus, we worked with two different analysis directions in order to avoid a limited indication reading. However, there are two drawbacks to this method

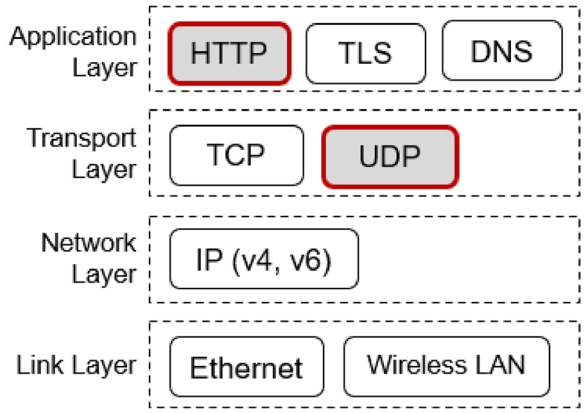


Fig. 5 Positions of deviations from the standard behaviours of protocols in the network layers

of analysis: it poses a possible threat to privacy, and it is computationally intensive.

In the analysis of deviations from standard behavioural indicators, the experimental findings showed deviations in two network layers: the transport layer and the application layer. The deviations were in two protocols: UDP and HTTP. Figure 5 shows the positions of the deviations from the standard behavioural indicators in the network layers.

5.2.1 Transport layer — UDP

For the CTU-13 botnet dataset, we realised that the botnet utilised the UDP protocol as the main carrier channel to infect computers. Compared to other protocols, UDP accomplishes this process in a simple fashion: it sends packets directly to a target computer without establishing a connection first and indicates the order of said packets or checks whether they have arrived as intended, unlike the TCP protocol, which completely relies on a handshaking-style connection. With many of the security mechanisms in other protocols, computers can drop suspicious requests; i.e. no acknowledgement is required. For example, we compare UDP connections to TCP handshaking in Fig. 6 to show the ease with which botnets can use UDP as a carrier channel.

The comparison reveals a valuable vision and provides a better understanding that can be used with indicators to recognise deviations from protocol standard behaviours. Our experimental analyses showed that the UDP protocol was more often leveraged by the P2P botnets than TCP, as shown in the screenshot in Fig. 7. The IP addresses of the botmaster and the bots are listed below the screenshot in Fig. 7.

5.2.2 Application layer — HTTP

Regarding the HTTP protocol and why it is preferable for exploitation by botnets, botmasters of P2P botnets might publish the commands on a certain website to update the bots. This process continues regularly at intervals predefined by the botmasters.

In recent years, HTTP has become the dominant protocol among the various protocols for Internet services as it provides a set of rules for the management of the data exchange between servers and browsers. Analysing HTTP traffic has thus become a common method in current HTTP-based botnet detection studies [17, 20, 23]. With the HTTP protocol, bots hide their communication flows within the normal HTTP flows, making them stealthy and difficult to detect. Monitoring and inspecting HTTP packets can reveal valuable information that can help network administrators analyse botnets' behaviour better and, ultimately, detect their presence in the network. In our experimental analyses, we identified several HTTP characteristics that were very helpful in distinguishing the bot traffic from the rest of the web network traffic. The screenshot in Fig. 8 clearly shows that the greatest numbers of packets were transmitted (Tx packets) and received (Rx packets) by the botmaster IP in the first place and sent to/from the infected machine in the second place, and there was a noticeable difference in

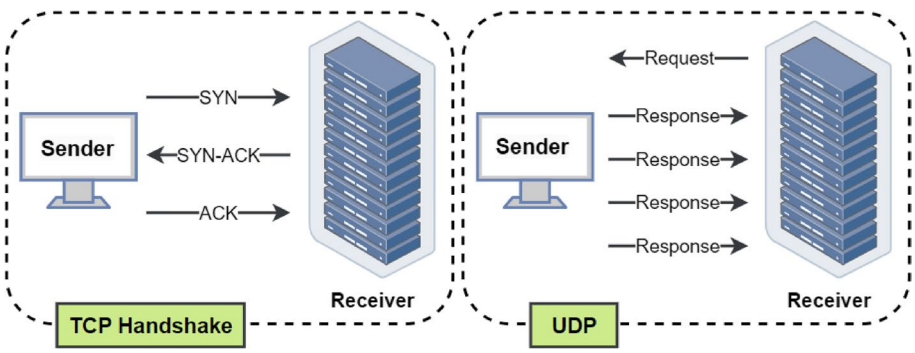


Fig. 6 TCP vs. UDP communications

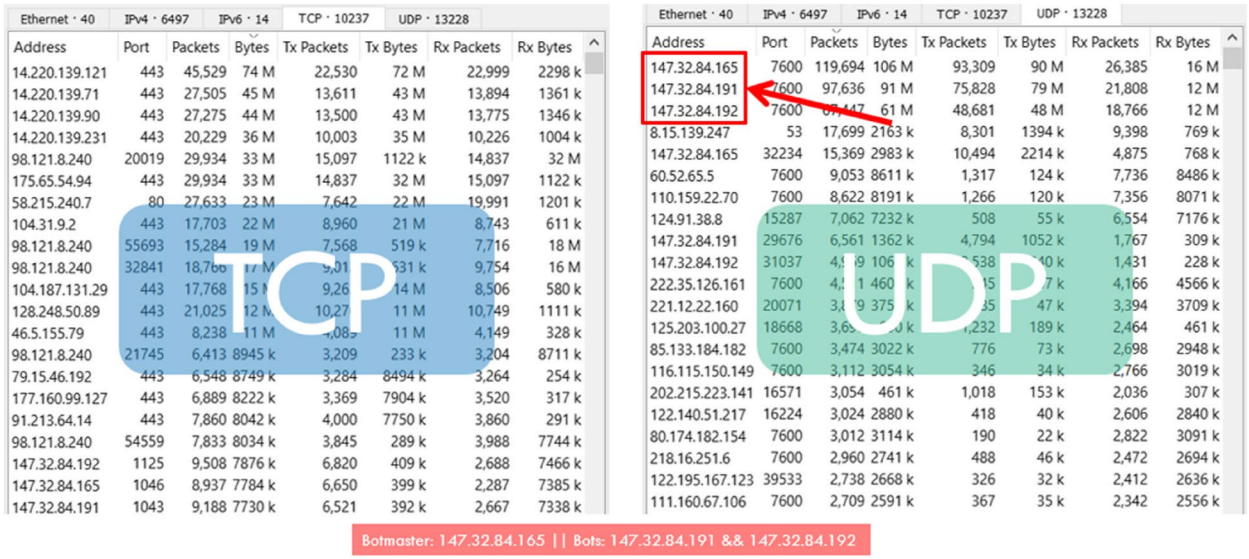


Fig. 7 TCP vs. UDP deviations from the standard behaviour

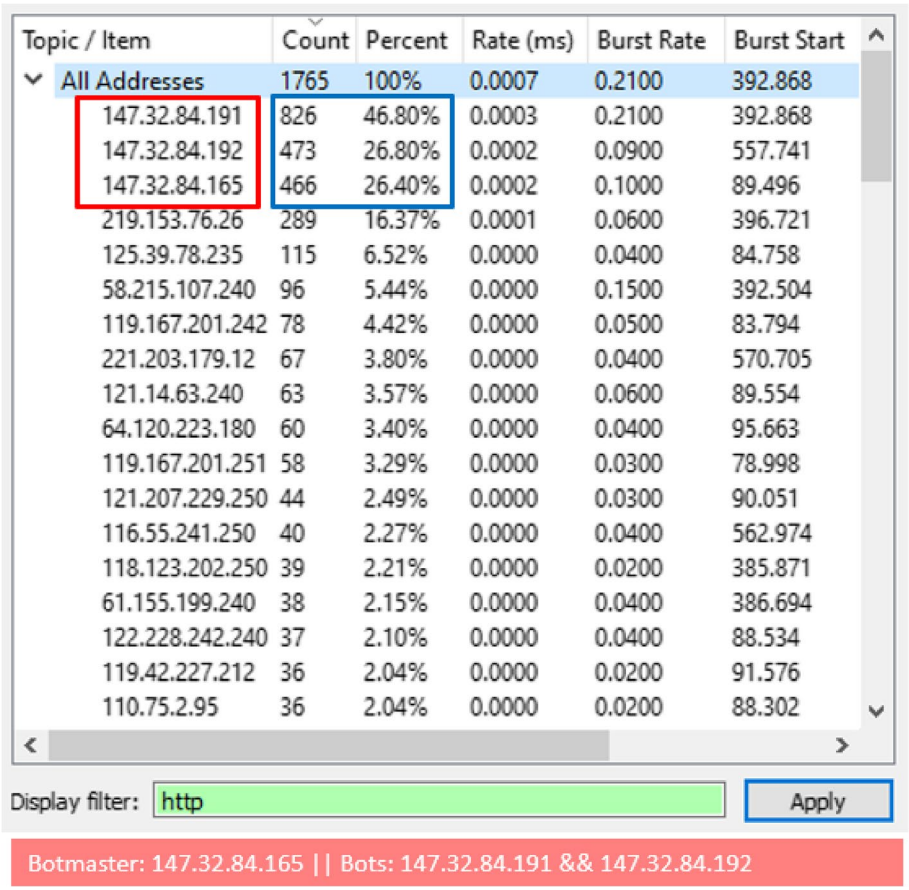


Fig. 8 HTTP bias standard behaviour

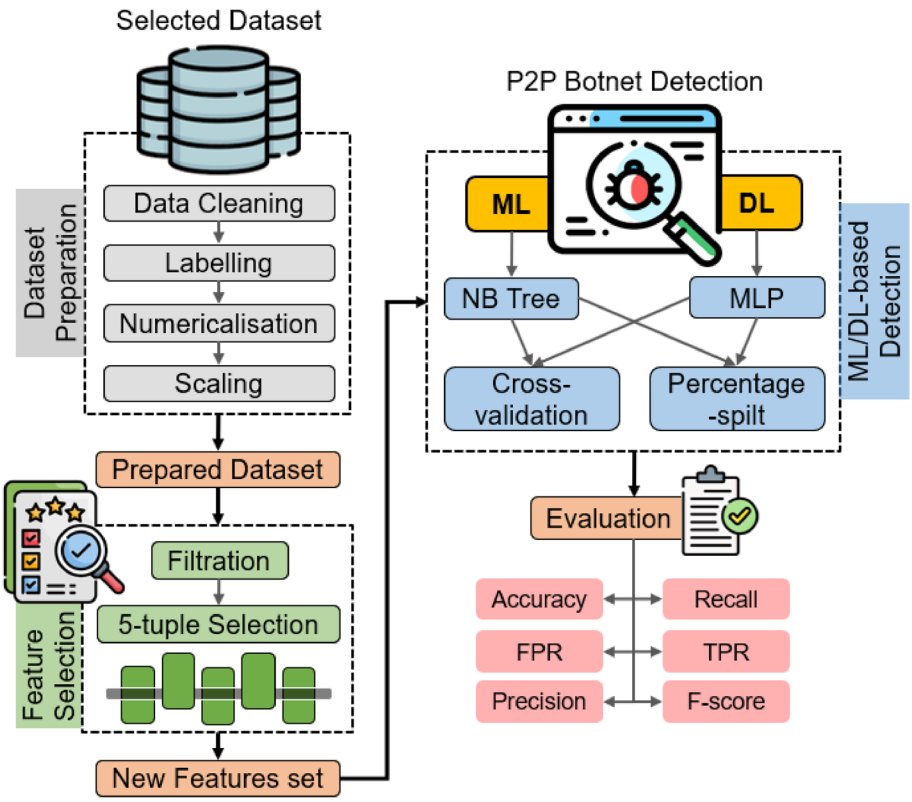


Fig. 9 The road map for the proposed approach

their percentages. The IP addresses of the botmaster and the bots are listed below the screenshot in Fig. 8. Keep in mind that the HTTP service is indispensable and widely used by many Internet applications, so it requires work to block it.

5.3 Detecting peer-to-peer botnets using the five-tuple static indicators

The rapid extension rates for network bandwidth are one of the most significant challenges for botnet detection systems. Thus, one of the critical assessment norms for IDS researchers is assessing the processing capability of IDSs. The well-known IDSs, such as Bro and Snort, nowadays consume large amounts of resources when they process a large amount of payload data over a high-speed network [51].

The orientation of the research shows the effectiveness of data mining and the adaptation of ML/DL techniques for detecting botnets [11, 51, 52]. For many reasons, such as the growing sizes of payload information streaming on the network and increasing network speeds, solutions that rely on learning-based techniques are preferable because these techniques can automate the processing of huge amounts of data. ML/DL technique-based solutions can save resources and time for systems, reduce the solution complexity, and make the process smoother. Moreover, data mining and ML/DL techniques are easy to apply to network flow information. In addition, the evaluation metrics are convenient indicators for the detection of botnets.

Given the above reasons, we experimentally examined two ML and DL techniques (NBTree and MLP) that have not previously been evaluated for the detection of P2P botnets using only the five-tuple features (previously mentioned in Sect. 5.1), i.e. the static indicators comprising the source IP address, destination IP address, source port number, destination port number, and protocol identifier number. The NBTree technique is a decision tree-based attribute-weighting technique with an adaptive Naïve Bayesian Tree [53]. The algorithm's pseudo-code and an analysis of NBTree can be found in [54], whereas the multilayer perceptron (MLP) is a deep neural network. Unlike other classification techniques, such as support vectors or the Naive Bayes classifier, MLP classifier relies on an underlying neural network to perform the task of classification [11]. The algorithm's pseudo-code can be found in [55].

The proposed approach consists of three major stages: data preparation, feature selection, and ML/DL-based

detection. Figure 9 shows the road map for the proposed approach.

5.4 Data preparation

The data preparation process entails the preparation of the selected dataset for the next stages through various steps that make it readable by the ML and DL algorithms. The first step after selecting the dataset was data labelling because we adopted supervised ML/DL techniques in the third stage. Thereafter, we labelled the dataset with multiple classes: botmaster, bot, and normal records. Data cleaning was necessary to remove the incorrectly formatted, incomplete, or corrupted data within the dataset because when merging multiple datasets (as described in the selected dataset [11]), as in the dataset construction, there are opportunities for data to be mislabelled or duplicated. Therefore, we converted the dataset into numerical data to make it understandable by the following algorithms. Finally, we scaled the numerical data to fit within a specific scale, such as 0–1 or 0–100. We scaled the dataset because of algorithms used in the third stage that are based on measuring how far apart the data points are, such as the ML algorithm [56]. The prepared dataset represented the input for the next stages.

5.5 Feature selection

As discussed previously in Sect. 5, we considered the static indicators—i.e. the five-tuple features comprising the source and destination IP addresses, source and destination port numbers, and protocol identify number—as selected features, in addition to the class, for the detection of the P2P botnets.

5.6 Machine and deep learning-based detection

The behaviour of P2P botnets is distinguishable from benign behaviour in a network. The P2P botnet detection issue could be modelled as a multi-class classification task, thanks to our previous labelling of the dataset into a botmaster, bots, and benign flows. In order to detect the P2P botnet, we used only the five-tuple features, as previously mentioned (Sect. 6.2). Accordingly, we adopted ML and DL techniques that have yet to be leveraged to detect the P2P botnets. Day by day, the relationship between cybersecurity and ML/DL techniques, such as AI applications, becomes stronger [57]. This interplay between cybersecurity and AI applications, such as ML, reflects the effectiveness of these solutions in defeating cyber threats [13, 52]. Although there are still some risks from AI in some fields (as discussed by Radanliev et al. [58]), it is efficient and effective in anomaly detection and worth investigating.

We used two different testing approaches: cross-validation and percentage splitting. The cross-validation testing approach splits the dataset into folds. For example, if there are 10-folds, 9 of them may be specified for training and evaluation purposes and only 1 for testing purposes. Percentage splitting splits the dataset into two different sets: the first comprises 80% of the original dataset and is for training purposes, while the other 20% of the original dataset is for testing purposes [59, 60].

5.6.1 Parameter settings

This section shows the parameter settings of the ML and DL classifiers used in this work. Two algorithms are used as classifiers, NBTree as ML classifier, and MLP as a DL classifier. As aforementioned, there are two testing approaches that are used in this stage: *cross-validation and percentage splitting*. The parameter settings that we set to MBTree are as follows. For cross-validation testing approach, the batch size was 100, where the number of decimal places to be used for the output of numbers in the model was 2. The number of folds that used to assess the performance and generalisation ability of NBTree was 10 in this experiment. For percentage splitting, the numbers of batch size and the decimal places are the same that were used in the cross-validation. In this testing approach, the dataset was sliced into 10-folds. This approach ensures that the proposed approach is trained on majority of the dataset while still retaining a portion for independent testing, helping to assess its generalisation to unseen data.

Whereas the parameter settings that we set to MLP are as follows. For cross-validation testing approach, the number of training instances utilised in one iteration is 100 (officially called as the batch size). In addition, there are 10 hidden layers in our proposed MLP. Furthermore, we set 0.3 as the learning rate for updating the weights of nodes, whereas the momentum that is applied to weight updates is 0.2. Last but not least, the number of folds that used to assess the performance and generalisation ability of MLP was 10 in this experiment. For percentage split, the number of training instances utilised in one iteration is also 100, when there are 10 hidden layers as well. Similarly, the learning rate and momentum are the same that are set to the cross-validation testing approach. In the testing approach, the dataset was divided into 80% for training and 20% for testing as performed by [11, 56].

5.6.2 Evaluation metrics

In general, there are many evaluation metrics that can be used to evaluate the performance of applied techniques, such as the false-positive rate (FPR) and true-positive rate (TPR). In this study, we evaluated our proposed approach using key metrics: accuracy, recall, precision, FPR, TPR, and F-score. Table 6 describes the evaluation metrics and the equations used to calculate those metrics [13].

Table 6 Evaluation metrics

Metric	Description	Equation
ACC	Accuracy is the standard measurement of an IDS. It refers to the percentage of records that are correctly classified	$ACC = \frac{TPR+TNR}{TPR+TNR+FPR+FNR}$
R	The ratio of correctly classified attack incidents to the number of real attacks	$R = \frac{TP}{TP+FN}$
P	The percentage of attack incidents correctly classified relative to the classified number of attacks	$P = \frac{TP}{TP+FP}$
FPR	The relative weaknesses of the proposed approach, in other words, it refers to the proportion of misclassifications	$FPR = \frac{FP}{TN+FP}$
TPR	The percentage of normal traffic that is classified as normal traffic	$TPR = \frac{TP}{TP+FN}$
FS	A combined measure of precision and recall	$F\text{-score} = \frac{Precision \times recall}{Precision+recall} * 2$

ACC, accuracy; *R*, recall; *P*, precision; *TP*, true positive; *TN*, true negative; *FP*, false positive; *FN*, false negative; *TPR*, true-positive rate; *FPR*, false-positive rate; *TNR*, true-negative rate; *FNR*, false-negative rate; *FS*, F-score

5.6.3 Experimental results

In this section, we compare the experimental results for our proposed approach to existing related work (see Table 1 in Sect. 2). As abovementioned, we applied ML and DL techniques to detect the P2P botnet: NBTree as a ML classifier and MLP as a DL classifier. Both classifiers surpassed the results of related work on evaluation metrics in terms of the accuracy, recall, precision, FPR, TPR, F-score, and even the time taken to build a model. NBTree as a ML technique has achieved a higher detection accuracy of 99.99% compared to the related works that adopted other ML techniques in their detection stages. In addition, NBTree also showed higher scores in terms of recall, precision, TPR, and F-score, compared to the related works. The experimental results of this ML technique showed its effectiveness in recognising the P2P botnets within a short record time taken to build a model of 53.68 s in cross-validation and 0.46 s in percentage split. Last but not least, this technique showed a superiority in terms of there was no FPR, which means this technique has very accurately recognised all the instances of P2P botnets as abnormal instances (attack) and recognised all the normal behaviour as such. In other words, this technique can accurately distinguish the behaviours of P2P botnets from the normal behaviours without any errors.

Meanwhile, MLP as a DL technique has also achieved a higher detection accuracy of 99.86% compared to all scores of detections in the related works. Moreover, MLP also achieved higher scores in terms of recall, precision, TPR, and F-score, compared to the related works. However, this technique took longer time to build a model compared to NBTree. The time taken to build a model using MLP was 269.43 s in cross-validation and 0.37 s in percentage split. According to [52], it is reasonable that DL techniques take longer time for training compared to ML techniques in case of exactly same experiment circumstances.

In general, the proposed approach using NBTree and MLP achieved higher detection accuracy compared to the related works by using only the static indicators (the five-tuple). The five-tuple represents five features, and this was the fewest number of features compared to other IDSs that have been proposed to detect P2P botnets. Initially, there were 30 features in the dataset, and after our analyses, we selected only 5 features to detect the P2P botnet. Relatively, we only used 16.6% of the original dataset to detect the P2P botnet and achieved very high detection accuracy. Technically, this saved around 84% of the time and resources normally consumed.

Achieving the highest detection accuracy using the fewest number of features can be advantageous for several reasons as follows: (i) *Simplicity*, where having smaller set of features can make the operation easier to understand and interpret, and that leads for a faster training [56]; (ii) *efficiency*, using fewer features may reduce the computational resources required to train the model, and that makes the detection process more efficient [56]; and (iii) *cost reduction*, collecting and preprocessing data for feature extraction can be resource-intensive, while using fewer features may reduce the cost associated with data collection and preprocessing. Taken together, the proposed approach showed its effectiveness and efficiency compared to the existing detection systems as discussed above.

Table 7 tabulates the experimental results for the proposed approach using two different testing approaches to evaluate NBTree and MLP as classifiers to detect P2P botnets.

It was challenging to conduct a fair comparison of the existing IDSs that have been developed to detect botnets and our proposed approach for many reasons, such as the following: (i) the fact that each approach/solution has been evaluated in a different environment [61], (ii) there are many different binary bots employed in the different experiments [61], and (iii) it is not trivial to obtain and execute the code for each solution [25]. Therefore,

Table 7 The evaluation metrics of the proposed approach using two testing approaches: cross-validation and percentage splitting

	Cross-validation		Percentage splitting	
Evaluation metric/technique	**NBTree (ML)**	**MLP (DL)**	**NBTree (ML)**	**MLP (DL)**
Accuracy (%)	99.99	99.68	99.99	99.96
Recall (%)	100	99.7	100	100
Precision (%)	100	99.7	100	100
FPR (%)	0.00	0.001	0.00	0.00
TPR (%)	100	99.7	100	100
F-score (%)	100	99.7	100	100
Time taken to build model (seconds)	53.68	269.43	0.46	0.37

Table 8 Comparison between the proposed approach and the related works in terms of accuracy, FPR, precision, recall, and F-score using ML techniques

Article	**Technique**	**Accuracy (%)**	**FPR**	**Precision (%)**	**Recall (%)**	**F-score (%)**
[6]	KNN	82.3	0.021	-	0.87	-
	REP Tree	96.7	0.00	-	0.99	-
	SVM	90.8	0.063	-	0.96	
[34]	J48	86.0	0.00	-	-	-
	Naïve Bayes	86.67	2.0	-	-	-
	BayesNet	93.33	1.0	-	-	-
[27]	Logistic regression	84.9	-	-	-	-
	Gaussian NB	76.3	-	-	-	-
	SVM	94.4	-	-	-	-
	Random forest	95.7	-	-	-	-
[28]	Random forest	97.45	-	97.27	96.67	96.98
	Naïve Bayes	42.45	-	42.34	99.84	59.46
	KNN	95.43	-	93.71	95.61	94.65
	Decision tree	97.06	-	96.39	96.67	96.53
	SVM	60.0	-	83.06	6.83	12.6
[29]	SVM	82.0	-	-	-	-
	Logistic regression	81.0	-	-	-	-
	KNN	97.0	-	-	-	-
	Decision tree	88.0	-	-	-	-
The proposed approach	**NBTree**	**99.99**	**0.00**	**100**	**100**	**100**

we undertook a traffic analysis to explore a new set of IOCs and then compared the performance of our detection approach to that found in the related work using the standard evaluation metrics. Table 8 compares the proposed approach to the related works in terms of accuracy, recall, precision, FPR, TPR, and F-score by using ML techniques. Take note, the comparison is exclusively performed to the related works that exactly proposed detection models/approaches/solutions to detect P2P botnets using either ML or DL techniques.

The above table shows that the proposed approach outperforms the ML-based-related works in terms of the standard evaluation metrics especially the detection accuracy. However, Table 9 compares the proposed approach to the related works in terms of accuracy, recall, precision, FPR, TPR, and F-score by using DL techniques.

Once again, the above table shows that the proposed approach outperforms the DL-based-related works in terms of the standard evaluation metrics. In addition, the proposed approach achieves the highest detection accuracy by using the fewest number of number features compared to the related works (five-tuple, static indicators). Figures 10 and 11 show the detection accuracy of

Table 9 Comparison between the proposed approach and the related works in terms of accuracy, FPR, precision, recall, and F-score using DL techniques

Article	Technique	Accuracy (%)	FPR	Precision (%)	Recall (%)	F-score (%)
[12]	Multilayer NN	99.20	0.75	-	-	-
[27]	Deep NN	97.36	0.96	-	-	96.93
[30]	ResNet	93.21	-	-	-	-
	CNN	88.87	-	-	-	-
The proposed approach	**MLP**	**99.96**	**0.00**	**100**	**100**	**100**

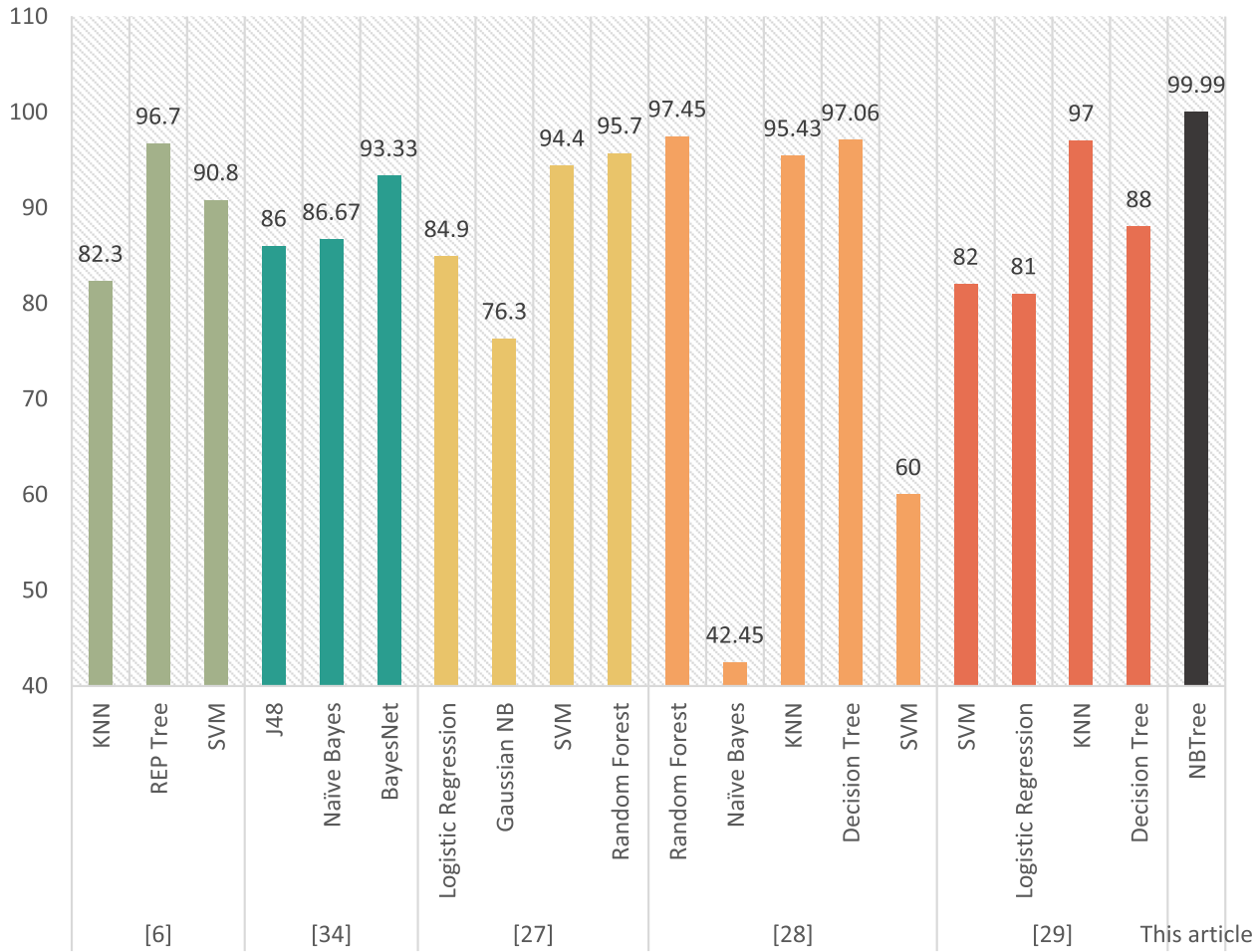


Fig. 10 The detection accuracy of the proposed approach (NBTree) compared to the ML-based related works

the proposed approach compared to the related works that based on ML and DL techniques, respectively.

The experimental results showed that the five-tuple features (static indicators) are enough to accurately detect P2P botnets using NBTree or MLP. In the above-mentioned comparison, the detection accuracy might show slight privilege, but considering the number of features used, this proposed approach outperforms the related works. Taken together, the proposed approach achieved the highest detection accuracy compared to the related works using the fewest number of features (five-tuple, static indicators). The performance reflects the effectiveness and efficiency of the proposed approach in detecting P2P botnets, showing that this

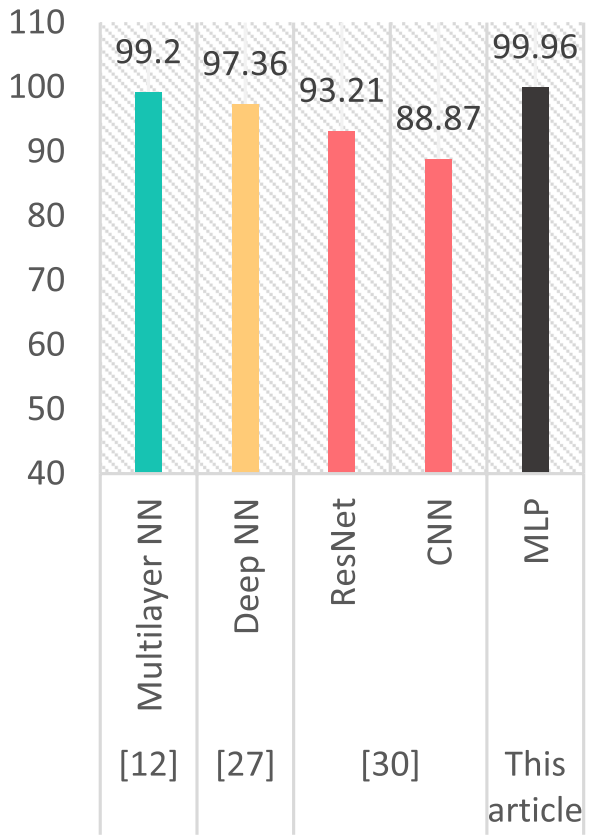


Fig. 11 The detection accuracy of the proposed approach (MLP) compared to the DL-based related works

approach is promising enough to depend and build on in future work.

6 Conclusion and future work

In this paper, we proposed two methods to detect P2P botnets. First, we analysed the traffic flow to develop a new set of behavioural characteristics as IOCs (or signs) of P2P botnets in two directions: flow-based indicators and indicators of deviations from standard protocol behaviour. Second, we proposed a new approach to detect P2P botnets using only static indicators (the five-tuple) using two ML/DL techniques as classifiers. The experimental results showed that these two methods are efficient security countermeasures to recognise and detect the P2P botnets. These two methods proved their efficiency to be adopted as a solid foundation for future research. To build upon this study, potential extensions of this research include dynamic analysis integration, i.e. incorporate dynamic indicators analysis techniques alongside adopting the static indicators to create a hybrid detection approach. In addition, enhancing the feature engineering, i.e. investigating in more sophisticated feature selection or feature ranking techniques to identify the most relevant indicators for ML/DL techniques. Finally, we encourage the upcoming researchers to approach and develop the real-time detection and response. In other words, it could be optimising the detection systems/approaches/models/solutions for real-time operation and allowing for immediate response to emerging threats.

Abbreviations

ACC	Accuracy
ACK	Acknowledgement
AI	Artificial intelligence
AMD	Advanced micro devices
ASCII	American Standard Code for Information Interchange
BPP	Bytes per packet
C&C	Command and control
CNN	Convolutional neural network
CPU	Central processing unit
CSV	Comma-separated value
DDoS	Distributed denial of service
DGA	Domain Generation Algorithm
DL	Deep learning
DNS	Domain Name System
FN	False negative
FP	False positive
FPR	False-positive rate
GB	Gigabyte
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDPS	Intrusion detection and prevention system
IDS	Intrusion detection system
IOC	Indicators of compromise
IP	Internet protocol
IRC	Internet relay chat
J48	Java implementation of C4.5 decision tree
KNN	K-nearest neighbour
LAN	Local area network
ML	Machine learning
MLP	Multilayer perceptron
NB	Naïve Bayes
NBTree	Naïve Bayes Tree
NN	Neural network
P	Precision
P2P	Peer to peer
PCAP	Packet CAPture
PPF	Packets per flow
R	Recall
RAM	Random-access memory
REP Tree	Reduced Error Pruning Tree
Rx	Received packet
SVM	Support vector machine
SYN	Synchronisation
TB	Terabyte
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TN	True negative
TP	True positive
TPR	True-positive rate
TTL	Time to live
Tx	Transmitted packet
UDP	User Datagram Protocol
WEKA	Waikato Environment for Knowledge Analysis

Authors' contributions

Conceptualization, Arkan Hammoodi Hasan Kabla; methodology, Arkan Hammoodi Hasan Kabla; software, Arkan Hammoodi Hasan Kabla; validation, Arkan Hammoodi Hasan Kabla, Achmad Husni Thamrin and Shankar Karuppayah; investigation, Achmad Husni Thamrin and Shankar Karuppayah; data curation,

Arkan Hammoodi Hasan Kabla; writing—original draft preparation, Arkan Hammoodi Hasan Kabla; writing—review and editing, Achmad Husni Thamrin and Shankar Karuppayah; supervision, Achmad Husni Thamrin, Mohammed Anbar, Selvakumar Manickam, and Shankar Karuppayah; project administration, Achmad Husni Thamrin and Shankar Karuppayah. All authors have read and agreed to the published version of the manuscript.

Funding
This work was supported in part by the Ministry of Higher Education Malaysia's Fundamental Research Grant Scheme under Grant FRGS/1/2021/ICT07/USM/03/1.

Availability of data and materials
The P2P botnet dataset used to evaluate this work is available at https://www.kaggle.com/datasets/arkantaha/p2p-botnet-dataset-peerabmush.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
The authors give permission for this work to be published in the *EURASIP Journal on Information Security*, and other publications produced by the *EURASIP Journal on Information Security*, in print and online.

Competing interests
The authors declare no competing interests.

Author details
[1]National Advanced IPv6 Centre, Universiti Sains Malaysia, 11800 Penang, Malaysia. [2]Graduate School of Media and Governance, Keio University, Fujisawa, Kanagawa 252-0882, Japan.

Received: 21 March 2024 Accepted: 16 May 2024
Published online: 27 May 2024

References

1. D.T. Son, N.T.K. Tram, P.M. Hieu, Deep learning techniques to detect botnet. J. Sci. Technol. Inf. Secur. **1**, 85–91 (2022). https://doi.org/10.54654/isj.v1i15.846
2. K.S.H. Ramos, M.A.S. Monge, J.M. Vidal, Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics. Sensors (Switzerland) **20**, 1–31 (2020). https://doi.org/10.3390/s20164501
3. Y. Zhong, A. Zhou, L. Zhang et al., Dustbot: a duplex and stealthy P2P-based botnet in the Bitcoin network. PLoS ONE **14**, 1–27 (2019). https://doi.org/10.1371/journal.pone.0226594
4. S. Karuppayah, Advanced Monitoring in P2P Botnets. (Singapore, Springer Singapore, 2018), p. XVII, 105. https://doi.org/10.1007/978-981-10-9050-9_7
5. D. Zhuang, J. Morris Chang, Enhanced PeerHunter: detecting peer-to-peer botnets through network-flow level community behavior analysis. IEEE Trans. Inf. Forensics Secur. **14**, 1485–1500 (2019). https://doi.org/10.1109/TIFS.2018.2881657
6. Z. Yang, B. Wang, A feature extraction method for P2P botnet detection using graphic symmetry concept. Symmetry (Basel) **11**, (2019). https://doi.org/10.3390/sym11030326
7. A. Hammoodi Hasan Kabla, M. Anbar, S. Manickam, et al., Monitoring peer-to-peer botnets: requirements, challenges, and future works. Comput. Mater. Contin. **75**:3375–3398 (2023). https://doi.org/10.32604/cmc.2023.036587
8. A.H.H. Kabla, M. Anbar, S. Manickam et al., Applicability of intrusion detection system on Ethereum attacks: a comprehensive review. IEEE Access **10**, 71632–71655 (2022). https://doi.org/10.1109/ACCESS.2022.3188637
9. R.Di. Pietro, L.V. Mancini, Intrusion Detection Systems, 1st edn. (Boston, Springer US, 2008). https://doi.org/10.1007/978-0-387-77265-3
10. M. Swarnkar, S.S. Rajput, *Artificial intelligence for intrusion detection systems, 1st Editio* (Chapman and Hall/CRC, Boca Raton, 2023)
11. A.H.H. Kabla, A.H. Thamrin, M. Anbar et al., PeerAmbush: multi-layer perceptron to detect peer-to-peer botnet. Symmetry (Basel) **14**, 2483 (2022). https://doi.org/10.3390/sym14122483
12. M. Alauthaman, N. Aslam, L. Zhang et al., A P2P botnet detection scheme based on decision tree and adaptive multilayer neural networks. Neural Comput. Appl. **29**, 991–1004 (2018). https://doi.org/10.1007/s00521-016-2564-5
13. A.H. Hasan, M. Anbar, T.A. Alamiedy, Deep learning approach for detecting router advertisement flooding-based DDoS attacks. J. Ambient. Intell. Humaniz. Comput. (2022). https://doi.org/10.1007/s12652-022-04437-0
14. M. Luqman, M. Faheem, W.Y. Ramay et al., Utilizing ensemble learning for detecting multi-modal fake news. IEEE Access **12**, 15037–15049 (2024). https://doi.org/10.1109/ACCESS.2024.3357661
15. Bibi M, Hussain Qaisar Z, Aslam N, et al., TL-PBot: Twitter bot profile detection using transfer learning based on DNN model. Eng Reports 1–25 (2024). https://doi.org/10.1002/eng2.12838
16. T.A. Al-Amiedy, M. Anbar, B. Belaton, A.H.H. Kabla, I.H. Hasbullah, Z.R. Alashhab, A Systematic Literature Review on Machine and Deep Learning Approaches for Detecting Attacks in RPL-Based 6LoWPAN of Internet of Things. Sensors. **22**(9):3400 (2022). https://doi.org/10.3390/s22093400
17. J.S. Lee, H.C. Jeong, J.H. Park, et al., The activity analysis of malicious http-based botnets using degree of periodic repeatability. Proc - 2008 Int. Conf. Secur. Technol. SecTech. **2008**, 83–86 (2008). https://doi.org/10.1109/SecTech.2008.52
18. W.T. Strayer, D. Lapsely, R. Walsh, C. Livadas, Botnet detection based on network behavior. Adv Inf Secur **36**, 1–24 (2008). https://doi.org/10.1007/978-0-387-68768-1_1
19. W. Lu, M. Tavallaee, A.A. Ghorbani, Automatic discovery of botnet communities on large-scale communication networks. Proc 4th Int Symp ACM Symp Information, Comput Commun Secur ASIACCS'09 1–10 (2009) . https://doi.org/10.1145/1533057.1533062
20. G. Kirubavathi Venkatesh, R. Anitha Nadarajan, HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 7322 LNCS:38–48 (2012). https://doi.org/10.1007/978-3-642-30955-7_5
21. G. Gu, R. Perdisci, J. Zhang, W. Lee, BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. USENIX Security Symposium. (2008)
22. B. Wang, Z. Li, D. Li, et al., Modeling connections behavior for web-based bots detection. 2010 2nd Int Conf E-bus Inf Syst Secur EBISS2010 141–144 (2010). https://doi.org/10.1109/EBISS.2010.5473532
23. M. Eslahi, H. Hashim, N.M. Tahir, An efficient false alarm reduction approach in HTTP-based botnet detection. IEEE Symp Comput Informatics, Isc **2013**, 201–205 (2013). https://doi.org/10.1109/ISCI.2013.6612403
24. D. Jang, K. Cho, M. Kim, et al., Evasion technique and detection of malicious botnet. In: IEEE Conf. Publ (2010). https://ieeexplore.ieee.org/document/5678101. Accessed 30 Oct 2022
25. A. AlAwadi Hasan, B. Belaton, Multi-phase IRC botnet and botnet behavior detection model. Int. J. Comput. Appl. **66**, 975–8887 (2013). https://doi.org/10.5120/11164-6289
26. M.R. Rostami, M. Eslahi, B. Shanmugam, Z. Ismail, Botnet evolution: network traffic indicators. Proc - 2014 Int Symp Biometrics Secur Technol ISBAST **2014** 274–279 (2015). https://doi.org/10.1109/ISBAST.2014.7013134
27. M. Alauthman, P2P bot detection using deep learning with traffic reduction schema. J. Theor. Appl. Inf. Technol. **98**, 2901–2912 (2020)
28. R. Lohiya, A. Thakkar, Intrusion Detection Using Deep Neural Network with AntiRectifier Layer. In: Thampi, S.M., Lloret Mauri, J., Fernando, X., Boppana, R., Geetha, S., Sikora, A. (eds) Applied Soft Computing and Communication Networks. Lecture Notes in Networks and Systems, vol 187. (Singapore, Springer, 2021). https://doi.org/10.1007/978-981-33-6173-7_7
29. A. Jaiswal, S. Tarar, Real-Time Biometric System for Security and Surveillance Using Face Recognition. In: Singh, M., Gupta, P., Tyagi, V., Flusser, J., Ören, T., Valentino, G. (eds) Advances in Computing and Data Sciences. ICACDS 2020. Communications in Computer and Information Science, vol 1244. (Singapore. Springer, 2020). https://doi.org/10.1007/978-981-15-6634-9_27
30. Z. Pei, G. Gan, Research on p2p botnet traffic identification technology based on neural network. IOP Conf Ser Earth Environ Sci **428** (2020). https://doi.org/10.1088/1755-1315/428/1/012011

31. B. Rahbarinia, R. Perdisci, A. Lanzi, K. Li, PeerRush: mining for unwanted P2P traffic. J Inf Secur Appl **19**, 194–208 (2014). https://doi.org/10.1016/j.jisa.2014.03.002
32. Priyanka, M. Dave, PeerFox: detecting parasite P2P botnets in their waiting stage. Proc 2015 Int Conf Signal Process Comput Control ISPCC **2015** 350–355 (2016). https://doi.org/10.1109/ISPCC.2015.7375054
33. H. Jiang, X. Shao, Detecting P2P botnets by discovering flow dependency in C&C traffic. Peer-to-Peer Netw. Appl. 7, 320–331 (2014). https://doi.org/10.1007/s12083-012-0150-x
34. W.H. Liao, C.C. Chang, Peer to peer botnet detection using data mining scheme. Int. Conf. Internet. Technol. Appl. ITAP 2010 - Proc 0–3 (2010). https://doi.org/10.1109/ITAPP.2010.5566407
35. D. Zhao, I. Traore, P2P botnet detection through malicious fast flux network identification. Proc - 2012 7th Int Conf P2P, Parallel, Grid, Cloud Internet Comput 3PGCIC **2012** 170–175 (2012). https://doi.org/10.1109/3PGCIC.2012.48
36. C. Yin, Towards accurate node-based detection of P2P botnets. Sci. World. J. **2014**, (2014). https://doi.org/10.1155/2014/425491
37. T. Yamanoue, A botnet detecting infrastructure using a beneficial botnet. Proc ACM SIGUCCS User Serv Conf 35–42 (2018). https://doi.org/10.1145/3235715.3235728
38. B. Rahbarinia, R. Perdisci, A. Lanzi, K Li, PeerRush: mining for unwanted P2P traffic. Lect. Notes. Comput. Sci. (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 7967 LNCS:62–82 (2013). https://doi.org/10.1007/978-3-642-39235-1_4
39. S. Garg, A.K. Singh, A.K. Sarje, S.K. Peddoju, Behaviour analysis of machine learning algorithms for detecting P2P botnets. 2013 15th Int Conf Adv Comput Technol ICACT 2013 0–3 (2013). https://doi.org/10.1109/ICACT.2013.6710523
40. M. Kuhn, K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, 1st edn. Chapman and Hall/CRC. (2019). https://doi.org/10.1201/9781315108230
41. S. Karuppayah, A. Jaisan, *DCNDS project dataset - P2P botnet detection using enhanced peer hunter*. (2021). https://doi.org/10.5281/ZENODO.5554851
42. CTU University, *The CTU-13 dataset*. (2013). https://www.stratosphereips.org/datasets-ctu13. Accessed 12 Oct 2022
43. P. Szumelda, N. Orzechowski, M. Rawski, A. Janicki, VHS-22-a very heterogeneous set of network traffic data for threat detection. ACM Int Conf Proceeding Ser 72–78 (2022). https://doi.org/10.1145/3528580.3532843
44. M. Aché, *MTA-KDD-19 | Kaggle*. (2019). https://www.kaggle.com/datasets/mathurinache/mtakdd19. Accessed 12 Oct 2022
45. P. Berba, *TrendMicro CTF Wildcard 400 | Kaggle*. (2019). https://www.kaggle.com/datasets/hawkcurry/2019-trendmicro-ctf-wildcard-400. Accessed 12 Oct 2022
46. N. Kaur, S. Behal, P2P-BDS: peer-2-peer botnet detection system. IOSR J Comput Eng **16**, 28–33 (2014). https://doi.org/10.9790/0661-16552833
47. A. Joshi, M.S. Chaudhary, Study of P2P botnet. IOSR J Comput Eng **16**, 35–42 (2014)
48. S. Saad, I. Traore, Ghorbani et al., *IMPACT - ISOT botnet dataset*. (2011). https://www.impactcybertrust.org/dataset_view?idDataset=1281. Accessed 12 Oct 2022
49. S. Saad, I. Traore, A. Ghorbani et al., Detecting P2P botnets through network behavior analysis and machine learning. 2011 9th Annu Int Conf Privacy. Secur Trust PST **2011**, 174–180 (2011). https://doi.org/10.1109/PST.2011.5971980
50. P. Narang, S. Ray, C. Hota, V. Venkatakrishnan, PeerShark: detecting peer-to-peer botnets by tracking conversations. Proc - IEEE Symp Secur Priv 2014-Janua:108–115. (2014). https://doi.org/10.1109/SPW.2014.25
51. E. Alparslan, A. Karahoca, D. Karahoc, BotNet detection: enhancing analysis by using data mining techniques. Adv Data Min Knowl Discov Appl (2012). https://doi.org/10.5772/48804
52. A.H.H. Kabla, M. Anbar, S. Hamouda, et al., Machine and deep learning techniques for detecting Internet Protocol version six attacks : a review. Int J Electr Comput Eng **13**:5617–5631. (2023). https://doi.org/10.11591/ijece.v13i5.pp5617-5631
53. A. Karahoca, (ed.), Advances in Data Mining Knowledge Discovery and Applications. InTech. (2012). https://doi.org/10.5772/3349
54. D.Y. Mahmood, M.A. Hussein. AnalyzingNB, DT and NBTree intrusion detection algorithms. J Zankoy Sulaimani - Part A **16**:69–76 (2014). https://doi.org/10.17656/JZS.10285
55. S. Mishra, H.K. Tripathy, B.K. Mishra, Implementation of biologically motivated optimisation approach for tumour categorisation. Int J Comput Aided Eng Technol **10**, 244–256 (2018). https://doi.org/10.1504/IJCAET.2018.090534
56. A.H.H. Kabla, M. Anbar, S. Manickam, S. Karuppayah, Eth-PSD: a machine learning-based phishing scam detection approach in Ethereum. IEEE Access **10**, 118043–118057 (2022). https://doi.org/10.1109/ACCESS.2022.3220780
57. P. Radanliev, D. De Roure, C. Maple, O. Santos, Forecasts on future evolution of artificial intelligence and intelligent systems. IEEE Access **10**, 45280–45288 (2022). https://doi.org/10.1109/ACCESS.2022.3169580
58. P. Radanliev, D. De Roure, C. Maple, U. Ani, Super-forecasting the 'technological singularity' risks from artificial intelligence. Evol. Syst. **13**, 747–757 (2022). https://doi.org/10.1007/s12530-022-09431-7
59. A. Saied, R.E. Overill, T. Radzik, Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing **172**, 385–393 (2016). https://doi.org/10.1016/j.neucom.2015.04.101
60. RMA Saad, A. Almomani, A. Altaher, et al., ICMPv6 flood attack detection using DENFIS algorithms. Indian. J. Sci. Technol. 7:168–173 (2014). https://doi.org/10.17485/ijst/2014/v7i2.5
61. W. Lu, G. Rammidi, A.A. Ghorbani, Clustering botnet communication traffic based on n-gram feature selection. Comput. Commun. **34**, 502–514 (2011). https://doi.org/10.1016/J.COMCOM.2010.04.007

Publisher's Note