Soft Comput (2017) 21:1315–1326
DOI 10.1007/s00500-015-1863-6

METHODOLOGIES AND APPLICATION

P2P and P2P botnet traffic classification in two stages

Wujian Ye[1] · **Kyungsan Cho**[1]

Published online: 19 September 2015

Abstract Nowadays accurate P2P traffic classification has become increasingly significant for network management. In addition, it is important to distinguish P2P botnet traffic from normal P2P traffic in order to find P2P malware and to immediately detect P2P botnets. Several approaches including port-based, signature-based, pattern-based, and statistics-based methods have been proposed to classify P2P and P2P botnet traffic. However, a single method alone cannot accurately classify both P2P and P2P botnet traffic. In this paper, we propose a hybrid traffic classifier that is composed of two stages. The first stage consists of a P2P traffic classifier that works in two steps. In the first step, a signature-based classifier is combined with connection heuristics, and in the second step, a statistics-based classifier is compensated by pattern heuristics. The statistics-based classifier is built using REPTree, a decision tree algorithm. The second stage is comprised of a P2P botnet traffic classifier that distinguishes P2P botnet traffic from other P2P traffic. The verification analysis and experiments using real datasets reveal that the proposed scheme provides a low overhead and achieves a high flow and byte accuracy of 97.70 and 97.06 % to classify P2P and P2P botnet traffic.

Keywords P2P traffic · P2P botnet traffic · Two-stage classification · Heuristic rules · Machine learning · Class imbalance problem

Communicated by V. Loia.

✉ Kyungsan Cho
kscho@dankook.ac.kr

[1] Department of Software Science, Dankook University, Yongin-si, Gyeonggi-do, Korea

1 Introduction

Peer-to-peer (P2P) technology allows any computer (referred to as a peer) to interact directly with other computers on the network, and in recent years, P2P file sharing has gained widespread use. P2P users on the edge of the network can use P2P file sharing application, such as eMule, Gnutella, and Kazaa, to share files that contain music, images, and video with each other (Tran et al. 2005). The rapid development of P2P applications has caused an explosive growth in Internet traffic, excessive bandwidth consumption, copyright violations, and security concerns. To address these problems and to control the quantity of P2P traffic in a network, it is necessary to accurately classify P2P traffic.

A botnet is a network of infected computers (referred to as bots) that are controlled by an attacker (referred to as a botmaster) in order to undertake malicious activities, such as Distributed Denial of Service (DDoS) attacks, phishing, and spamming (Elhalabi et al. 2013). In comparison with the centralized IRC and HTTP botnets, decentralized P2P botnets are more resilient against detection and takedown since they avoid a single point of failure. P2P botnets have become a serious threat to network security. Their traffic is buried within enormous, normal P2P traffic, which makes it harder to detect them and keeps P2P botnet stealthier (Kheir and Wolley 2013). Thus, it is necessary to separate P2P botnet traffic from other P2P traffic, in order to find P2P malware and to detect P2P botnets before they have completed their missions during command and control (C&C) or attack phases.

Several approaches have been proposed to classify P2P and P2P botnet traffic, but port-based and signature-based methods, for example, have not been effective since a great number of P2P network applications and P2P botnet malware have adopted the port disguise and payload encryption techniques. Although pattern-based and statistics-based methods

are effective against encrypted and unknown traffic well, pattern-based methods cannot detect a single flow, and statistics-based methods cannot detect untrained flows and suffer from a class imbalance problem.

A review of related works indicates that a single method is not sufficient to classify P2P traffic and to detect P2P botnet traffic accurately. Thus, we propose an improved hybrid traffic classification scheme that consists of two stages. In the first stage, a P2P traffic classifier executes two steps. In the first step, a packet-level signature-based classifier is combined with connection heuristics to classify P2P traffic. In the second step, a statistics-based classifier and pattern heuristics are applied to classify the remaining unknown traffic at the flow level. The statistics-based classifier is implemented by using REPTree, a decision tree algorithm. In the second stage, a P2P botnet traffic classifier distinguishes P2P botnet traffic from normal P2P traffic. The first stage filters most non-P2P traffic and accelerates the classification in the second stage. Our two-stage scheme exhibits a reduced error rate when detecting P2P botnet traffic, and it overcomes the class imbalance problem since the two stages have a low error correlation each other.

The rest of this paper is organized as follows. In Sect. 2, we analyze P2P and P2P botnet technologies and review related works. In Sect. 3, we propose an improved hybrid scheme to classify P2P and P2P botnet traffic in two stages. In Sect. 4, we present the performance evaluation of our proposed scheme. Finally, we conclude our research in Sect. 5.

2 Related works

2.1 P2P network

In recent years, P2P technology has become widely used, and in a P2P network, peers are connected to each other via the Internet. In such networks, files can be shared directly between peers without a central server. Each peer becomes both a file server and a client (Ye 2012; Valdés et al. 2015). Table 1 presents the comparison of P2P network with client–server network.

In P2P networks, there is no single point of failure that can happen in client–server networks. In addition, P2P prevents a network bottleneck since P2P can distribute data and can balance requests across the network without using a central server. Better scalability can be provided by a decentralized control, and unused resources, such as computing power and storage capacity, can be completely utilized in a P2P network. However, many applications need a high standard of security that has not yet been satisfied by current P2P solutions (Maly et al. 2003).

Table 1 Comparison of P2P and client–server networks

Features	P2P network	Client–server network
Architecture type	Decentralized	Centralized
Single point of failure	No	Yes
Bottleneck	Small	Large
Scalability	Easy	Hard
Resource sharing	Yes	No
Security	Bad	Good
Data management	Hard	Easy
Response speed	Slow	Fast
Cost	Cheap	Expensive
Maintenance	Few	Much

2.2 P2P botnet

A botnet is a network of compromised computers (bots) running malicious software that is usually installed through the use of different attack vectors, such as worms, viruses, and Trojan horses. Bots are remotely controlled by a botmaster, and they respond to the botmaster's commands to initiate various malicious activities (Elhalabi et al. 2013; Chiou et al. 2014). The main types of attacks carried out by botnets are DDoS attacks, spamming, click fraud, key logging and adware, stealing personal information, distribution of pirated media, and so on (Tyagi and Aghila 2011; Castiglione et al. 2014).

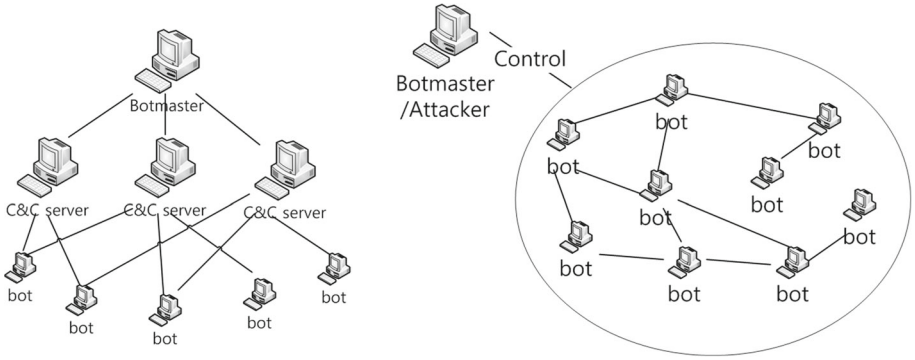
A C&C channel is the most critical component of a botnet because it allows the distribution of any command from a botmaster to any bot. This channel typically serves as the only way to control bots within the botnet (Silva et al. 2013). According to the C&C channel, botnets can be categorized into centralized and decentralized types. Their architectures are shown in Fig. 1.

In a centralized botnet, the botmaster usually chooses a high bandwidth computer as a central point (C&C server) for all bots (Vania et al. 2013). This C&C server exchanges commands and data between the botmaster and the bots (Tyagi and Aghila 2011).

In a decentralized botnet, attackers exploit P2P communications to proxy commands (Tyagi and Aghila 2011). In this case, a bot keeps some connections open with other bots in the botnet, acting as both a client and a server (Vania et al. 2013). As a result of the huge popularity of P2P file sharing systems, P2P bots can spread very quickly in P2P networks. Moreover, their traffic is buried within an enormous amount of normal P2P traffic, which makes it more difficult to detect botnets since these can be quite stealthy (Kheir and Wolley 2013).

P2P botnets are also more difficult to be disarticulated because detecting several or even many bots does not nec-

Fig. 1 Architecture of botnets



(a) Centralized IRC/HTTP botnet **(b)** Decentralized P2P botnet

essarily mean that the entire botnet has been lost, as there is no central C&C server (Silva et al. 2013). If nodes are taken offline, the gaps in the network are closed and the network continues to operate under the control of the botmaster. One more problem posed by P2P botnets to security specialists is the difficulty in estimating the size of the P2P botnet (Dittrich and Dietrich 2008).

2.3 Classification of P2P traffic

The four major methods that have been proposed to classify P2P traffic include port-based, signature-based, pattern-based, and statistics-based methods. Single-step methods involve one of these, while multi-step methods are the combination of the above four single-step methods.

It is impossible to classify all P2P traffic by using port-based and signature-based methods because a great number of P2P applications and P2P botnet malware, such as Nugache, Storm, Waledac, and Conficker, dynamically use arbitrary available port numbers and employ encryption mechanisms to transfer messages (Jiang and Shao 2012).

Pattern-based and statistics-based methods have been proposed to overcome the limitations of port-based and signature-based methods. The basic idea of using a pattern-based method is to look at the communication pattern that is generated by a particular host and to compare it to behavior patterns representing different activities or applications (Szabó et al. 2008). Karagiannis et al. (2004) detect P2P traffic according to two P2P behavior patterns in terms of an {IP, port} pair and a UDP/TCP pair. However, their method is not able to classify a single flow. Lu et al. (2012) use port association to assist traffic classification, which can speed up traffic classification, but is an auxiliary method that cannot classify all traffic. He et al. (2014) consider the aggregation flows as the patterns of network activities to build profiles for given P2P applications and perform the traffic classification. Wang et al. (2009) present a detection approach for P2P Storm botnets that is based on the stability of the C&C traffic. Their method is able to classify 98 % of strom C&C traffic as 'stable' with a false-positive (FP) rate of 30 % (Zhao et al. 2012). Jiang and Shao (2012) identify C&C communications from P2P bots by discovering flow dependencies in the C&C traffic, but this method may have difficulty in discovering the flow dependency when these flows rarely occur.

Statistics-based methods classify Internet traffic according to statistical features extracted from traffic traces, such as packet size, packet inter-arrival time, and flow duration. Nonlinear features are also proposed to characterize Internet traffic by applying recurrence quantification analysis technique recently (Palmieri and Fiore 2009). However, the increase in the number of features has made it more difficult to manually specify a mapping between the features and the respective traffic classes. Hence, machine learning (ML) algorithms are employed to classify traffic by applying different algorithmic procedures to automatically construct a statistics-based classifier model from a pre-labeled training dataset. Then, this classifier is used to group flow instances into different classes based on the values of their features (Soysal and Schmidt 2010; Narudin et al. 2014). The ML algorithms that are commonly used are the k-nearest neighbors (KNN), artificial neural network (ANN), support vector machine (SVM), decision tree (DT), rule learner (RL), and Nave Bayes (NB) algorithms. Based on our previous studies, DT shows high tolerance to missing values and noise due to its special mechanisms, such as its pruning strategies, and it has a high comprehensibility since it produces results that can be communicated very well in symbolic and visual terms and is easy to understand and use. Furthermore, DT provides a higher accuracy than the other five ML algorithms for P2P traffic classification (Ye and Cho 2014a). Three DT algorithms REPTree, CART, and C4.5 are analyzed in our previous study, and the performance of C4.5 is found to be the lowest because its pruning algorithm is prone to under-pruning, and the overfitting of C4.5 is more serious than that of REPTree and CART. The pruning algorithm of CART

results in more over-pruning than REPtree. Thus, REPTree provides higher accuracy than CART (Ye and Cho 2014b).

Recent works have mainly focused on achieving a high flow accuracy rather than a high byte accuracy (He et al. 2008). However, both flow accuracy and byte accuracy are important because a few misclassified flows could result in many bytes being incorrectly classified (Erman et al. 2007b).

ML algorithms have been widely applied to Internet traffic classification. However, due to the imbalance in the number of traffic flows (also referred to as class imbalance problem), classifiers are prone to misclassify flows as the traffic type that occupies the majority of flows on the Internet (Zhang et al. 2012). Thus, without considering the class imbalance problem, classifier built using ML algorithms, such as DT and NB, may produce a high flow accuracy but a low byte accuracy (Erman et al. 2007a).

In order to overcome the class imbalance problem, Erman et al. (2007a) propose semi-supervised and sampling techniques. The results indicate that the flow accuracy is around 90 %, and the byte accuracy is between 60 and 85 %. However, their sampling methods have been criticized mainly because they alter the original class distributions (Zhang et al. 2012). He et al. (2008) combine ensemble learning with a cost-sensitive algorithm, and they obtain a flow accuracy of 94 % as well as a byte accuracy of 81 %. However, the cost matrix used by cost-sensitive algorithms for real-world problems may contain uncertainty throughout both training and testing (Wang and Tang 2012). Zhang et al. (2012) propose two feature selection algorithms WSU_AUC and SRSF to select the optimal features that should be applied in practice. They can achieve a flow accuracy of above 94 % and a byte accuracy of above 80 %, on average. Statistics-based methods can classify unknown or encrypted P2P traffic, but their accuracy is not high and they cannot correctly identify untrained flows. Their FP rates are high with respect to P2P botnet traffic detection. Furthermore, they do not work well in online situations and result in a large quantity of computation.

As a result, it is hard to apply a single method to classify P2P traffic and to detect P2P botnet traffic completely. Thus, multi-step classifiers that combine several methods have been proposed. Chen et al. (2009) design a classifier composed of a static feature-based hardware classifier and a Flexible Neural Tree-based software classifier. This combined method achieves an accuracy of 95.67 %. Li et al. (2009) propose a two-step classifier that consists of coarse-grain classification and fine-grain classification, and the accuracy of this method is as high as 96.03 %. Keralapura et al. (2010) also propose a multi-step method that uses a time correlation metric (TCM), and its accuracy is of 95 %. In our previous study, we apply a signature-based classifier at the packet level combined with connection heuristics and a statistics-based classifier built using C4.5 algorithm at the flow level (Ye and Cho 2013). This method achieves a higher accuracy of 97.46 %. Zeng and Shin (2013) propose a two-step distributed approach to detect Storm botnets. Their method includes a set of heuristics and port numbers in the first step and a SVM classifier in the second step. It turns out that they can pinpoint more than 95 % of P2P traffic with 8–12 % FP rates, and this scheme works well with 0 % FP rate and 8 % false-negative (FN) rate to detect Storm botnet hosts. Zhang et al. (2014) identify stealthy P2P botnets by deriving statistical fingerprints of the P2P communications to first detect P2P clients and further distinguishing between those that are part of legitimate P2P networks and P2P bots. The evaluation results demonstrate that they can accomplish high accuracy and great scalability.

3 Proposed schemes

3.1 Motivation of two-stage scheme

Figure 2 shows the overlapping characteristics of the three traffic types: normal P2P traffic, P2P botnet traffic, and non-P2P traffic. According to Fig. 2, non-P2P traffic, such as HTTP, shows similar characteristics to P2P botnet traffic, which mainly includes small flows. Thus, non-P2P traffic can be easily classified as P2P botnet traffic. In addition, large flows of normal P2P traffic can be misclassified as P2P botnet traffic or non-P2P traffic due to the class imbalance problem.

If the three forms of traffic are classified in a single-stage scheme, as shown in Fig. 3a, a considerable part of the P2P botnet traffic sharing common characteristics with non-P2P traffic may be easily misclassified, and a small number of large P2P flows can be misclassified as a large number of small flows, such as P2P botnet traffic or HTTP traffic due to the class imbalance problem. Therefore, a single-stage scheme is not sufficient to classify the three traffic types shown in Fig. 2 at the same time.

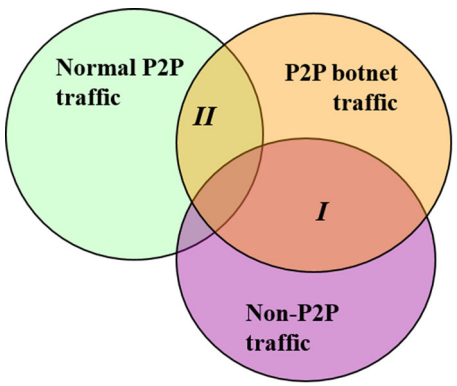


Fig. 2 Overlapping characteristics of the three traffic types

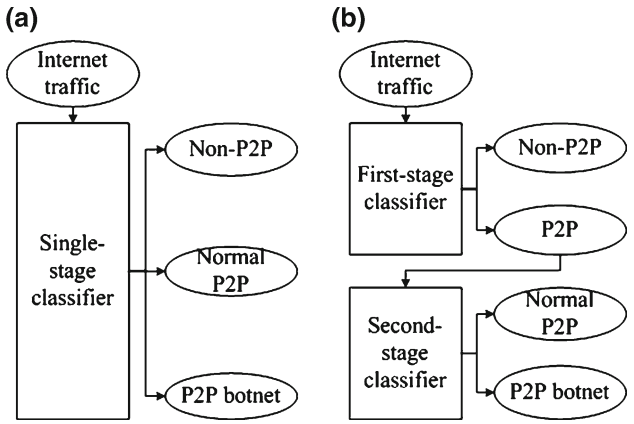


Fig. 3 Comparison of two schemes

Thus, a two-stage scheme is necessary in order to solve the above limitations. Although many possible two-stage schemes exist, we propose a two-stage scheme as shown in Fig. 3b. In our two-stage scheme, the first stage classifies most P2P flows according to the common characteristics of normal P2P and P2P botnet traffic. Most non-P2P flows are filtered, which accelerates the second-stage classification. In the second stage, P2P botnet traffic is detected among the P2P traffic due to the special characteristics of its control messages. The two stages have a low error correlation with each other since the statistics-based classifiers in each stage are trained using different flow feature sets. Thus, a two-stage scheme decreases the final error rate when detecting P2P botnet traffic and also overcomes the class imbalance problem.

3.2 Proposed system

Based on our analysis of related works, we propose an improved hybrid traffic classification system to classify P2P and P2P botnet traffic. Our proposed system consists of a traffic capturing preprocessor and a hybrid traffic classifier. Figure 4 shows the architecture that is implemented using Jpcap (2007) and Weka (2012).

First, the traffic capturing preprocessor captures packets from the network, validates the packets, and filters unnecessary packets.

Then, the hybrid traffic classifier works across two stages. In the first stage, a P2P traffic classifier classifies P2P traffic in two steps. In the second stage, a statistics-based P2P botnet traffic classifier further distinguishes P2P traffic as either normal P2P traffic or P2P botnet traffic.

3.3 P2P traffic classification (in the first stage)

In this subsection, the P2P traffic classifier in the first stage is depicted in greater detail.

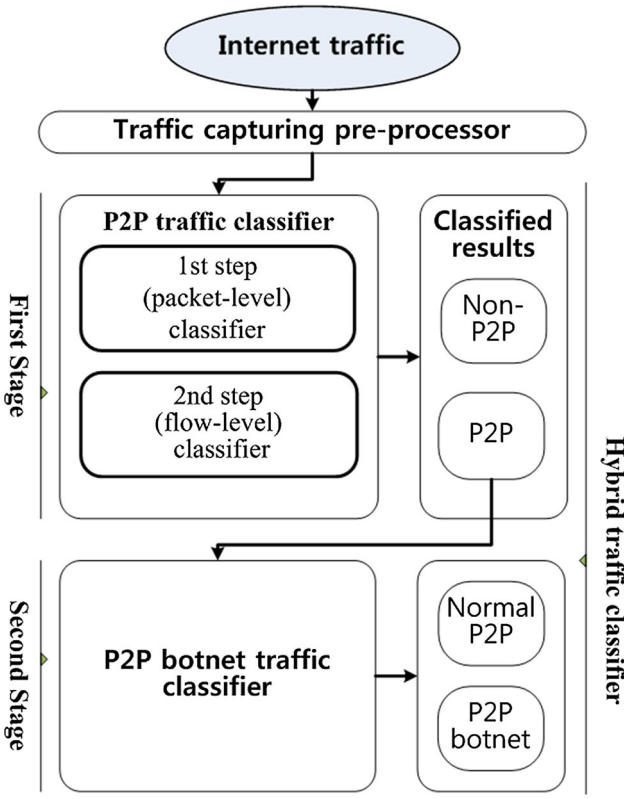


Fig. 4 Architecture of the proposed system

3.3.1 First-step (packet level) classification

In the first step, a signature table stores pre-defined signature strings that are contained in the payload of a P2P packet. The connection heuristics that are proposed in our previous study (Ye and Cho 2014a) are applied in order to reduce the amount of computation required to analyze packets. Figure 5 shows the process of the first-step classification. First, the signature-based classifier checks whether the payload of a packet contains any pre-defined string in the signature table. We classify the entire flow that contains this packet accordingly. If the packet does not contain a signature string, we check whether the packet satisfies the connection heuristics. If a packet is not classified in the first step, it is regarded to be unknown traffic and will be classified in the second step.

3.3.2 Second-step (flow level) classification

Our previous studies (Ye and Cho 2014a, b) indicate that packet size-related features offer better performance than arrival time-related features and volume-related features. The packet size seems to be the best feature for any traffic classifier to use (Este et al. 2009).

We can sort packet size-related features into two categories, as shown in Table 2. Coarse-grained features include the general statistics for all the packets within a flow, and

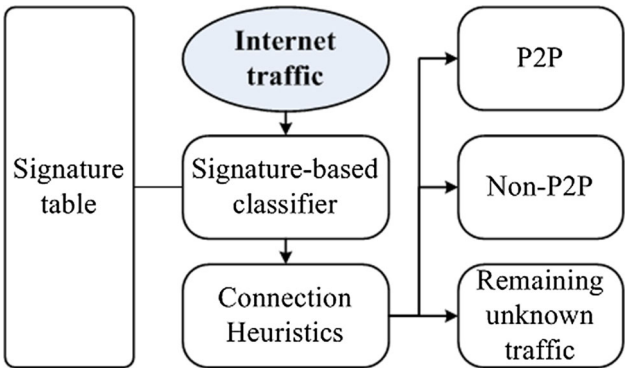


Fig. 5 Process of the first-step (packet level) classification

Table 2 Categories of packet size-related features

Flow feature set	Description
Coarse-grained features	Payload size of packets in a flow (min, max, SD, mean)
Fine-grained features	Payload size of each packets in a flow

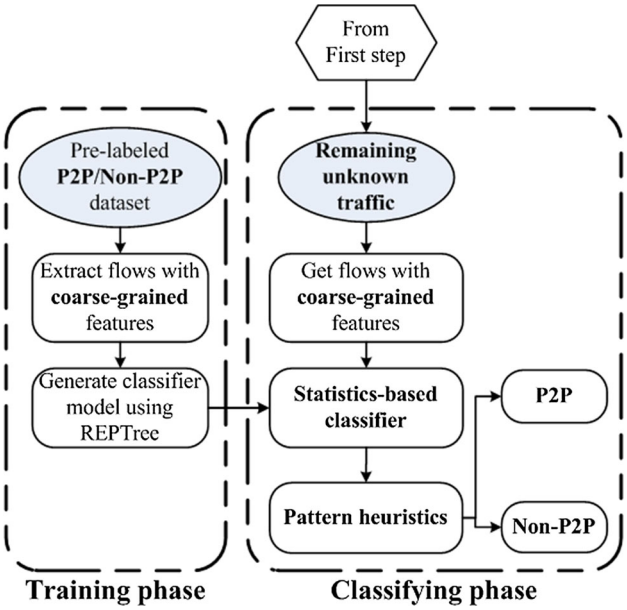


Fig. 6 Process of the second-step (flow level) classification

fine-grained features are obtained directly from information specific to each packet.

Coarse-grained features are used to classify traffic into P2P and non-P2P traffic. Since a P2P botnet utilizes P2P protocols, its traffic exhibits common characteristics of P2P traffic, which are hidden in the coarse-grained features.

On the other hand, fine-grained features are applied to distinguish P2P botnet traffic from normal P2P traffic. For many protocols, the initial packet exchange tends to be unique and follows well-defined behavior when a client joins a network (Zhao et al. 2012; Bernaille et al. 2006). Since P2P bots have special control messages when communicating with each other, the payload size of the first five packets in a flow contains these characteristics.

In our previous study (Ye and Cho 2014a), we show that DT has a high tolerance, high performance, and high comprehensibility for P2P traffic classification. We also show that, among DT algorithms, REPTree is more suitable for classifying P2P traffic than CART and C4.5 (Ye and Cho 2014b). Thus, the REPTree algorithm is used in our proposed scheme to build a statistics-based classifier.

The second step involves a statistics-based classifier at the flow level. This step consists of a training phase and a classifying phase shown in Fig. 6. In the training phase, the statistics-based classifier is built using REPTree with coarse-grained features inferred from a pre-labeled P2P/non-P2P training dataset. During the classifying phase, the statistics-based classifier is used to classify any remaining unknown flows. If the coarse-grained features of a flow satisfy the unique characteristics of P2P traffic, the statistics-based classifier classifies it as a P2P flow. Finally, pattern heuristics proposed in our previous study (Ye and Cho 2014b) are used to rectify faulty results produced by the statistics-based classifier.

3.4 P2P botnet traffic classification (in the second stage)

The second stage involves a P2P botnet traffic classifier that further classifies P2P traffic as normal P2P or P2P botnet traffic, and it also consists of a training phase and a classifying phase shown in Fig. 7. For the training phase, the P2P botnet traffic classifier is built using REPTree with fine-grained features inferred from a pre-labeled P2P botnet/normal P2P training dataset. In the classifying phase, the P2P botnet traffic classifier detects P2P botnet traffic among the classified P2P traffic. If the fine-grained features of a flow satisfy the unique characteristics of P2P botnet traffic, the P2P botnet traffic classifier classifies it as a P2P botnet flow.

3.5 Statistics-based schemes for classifying P2P and P2P botnet traffic

We compare the performance of the single-stage scheme (MOne) and the two-stage scheme (MTwo). The MOne scheme classifies Internet traffic as non-P2P, normal P2P, or P2P botnet traffic directly through the use of a single statistics-based classifier C0, which is trained using REPTree with all coarse-grained and fine-grained features. The MTwo scheme detects P2P botnet traffic through the use of two statistic-based classifiers: classifier C1 but without pattern heuristics shown in Fig. 6 and classifier C2 shown in

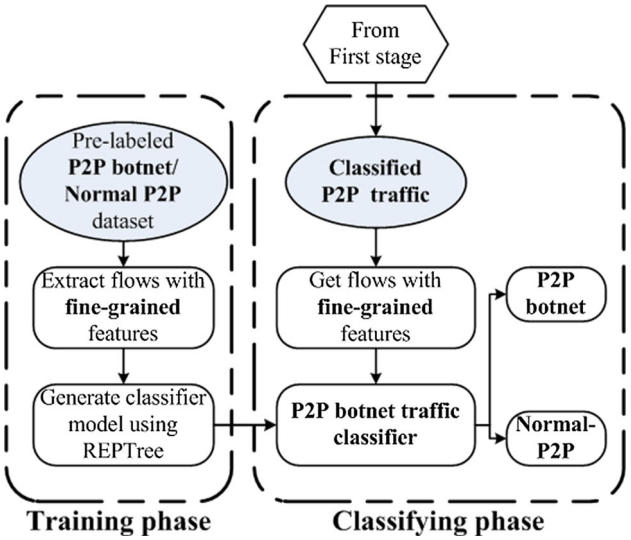
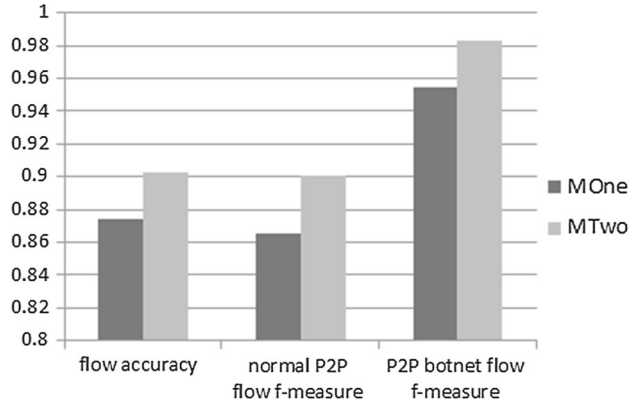
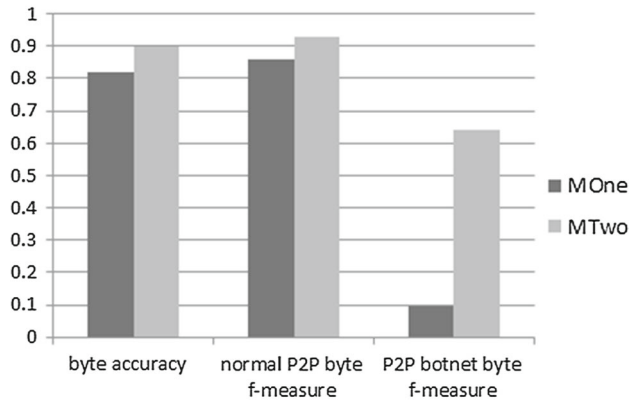


Fig. 7 Second-stage classification process



(a) Flow accuracy, P2P and P2P botnet flow f-measures



(b) Byte accuracy, P2P and P2P botnet byte f-measures

Fig. 8 Comparison of two statistics-based schemes

Fig. 7. At first, classifier C1 classifies unknown traffic as non-P2P or P2P traffic. Then, classifier C2 further classifies the P2P traffic as either normal P2P traffic or P2P botnet traffic.

As shown in Fig. 8, the MOne scheme achieves relatively high flow accuracy, but its byte accuracy is not high and the P2P botnet byte f-measure is low. This is caused by two reasons. First, the part of the P2P botnet traffic that involves characteristics similar to non-P2P traffic is easily misclassified. Second, a small number of large flows, such as for P2P traffic, is easily misclassified as a large number of small flows, such as P2P botnet traffic, due to the class imbalance problem.

In the MTwo scheme, most of the non-P2P traffic is filtered and the P2P botnet traffic is almost classified as P2P traffic by the coarse-grained features in classifier C1. Then, the P2P botnet traffic is well separated from normal P2P traffic with fine-grained features in classifier C2. Since the C1 and C2 classifiers are trained independently by using different flow features, they have a low error correlation with respect to each other. Thus, the MTwo scheme reduces the total error rate to detecting P2P botnet traffic and also reduces the misclassification of large flows. The MTwo scheme achieves a higher flow accuracy as well as a higher byte accuracy.

Table 3 presents a comparison of the above two schemes. Since scheme MOne and MTwo use the same flow features, their computational amounts to collect flow features are the same. In our experiments, the size of the decision tree, the training time ratio and the classifying time ratio between scheme MTwo and MOne are analyzed. Relative to MOne, MTwo requires more memory. However, the training time for MTwo is less than that for MOne since C1 and C2 are trained using different feature sets in parallel. What's more, since the flow features used by C1 or C2 are just a subset of the flow features used by C0, the decision tree sizes for C1 and C2 are smaller than that of C0. Thus, MTwo makes a decision faster.

Table 3 Comparison of schemes for MOne and MTwo

Scheme	MOne	MTwo
Computation for collecting flow features	The same	
Early classification	Yes	
Size of decision tree	Small	Large
Training time	Slow	Fast
Classifying time	Middle	Fast
Flow accuracy	Middle	High
Byte accuracy	Low	High

Since MTwo provides better performance than MOne, we choose a two-stage scheme to classify P2P and P2P botnet traffic. In our proposed scheme, C1 is the statistics-based classifier in Fig. 6, and C2 is the P2P botnet traffic classifier in Fig. 7.

Table 4 Amount of normal flows in the datasets

Traffic type	UNIBS	Ericsson	DKU1	DKU2	DKU3
Non-P2P	53,279	3293	13,190	11,247	11,860
P2P	21,716	5861	14,464	15,734	23,988
Total	74,995	9003	27,654	26,981	35,848

Table 5 Amount of P2P botnet flows in the datasets

P2P botnet traffic datasets	Set1	Set2	Set3	Total
Number of flows	8906	9004	9258	27,168

4 Verification

4.1 Evaluation metrics

In general, the following two standard metrics are used to evaluate traffic classifiers. X is a traffic class in which we are interested (such as normal P2P, P2P botnet) (Powers 2011):

- *Accuracy* The percentage of correctly classified instances among the total number of instances.
- *F-measure* A harmonic mean of the recall and precision of class X. Recall is the percentage of those instances of class X that are correctly classified as belonging to class X. Precision is the percentage of those instances that truly belong to class X among all those classified as class X.

4.2 Datasets

Five benign datasets (UNIBS, Ericsson, DKU1, DKU2, and DKU3) are used for our research, as shown in Table 4. UNIBS is a traffic trace provided by the University of Brescia, and it is collected by the ground truth system (Gringoli et al. 2009). Ericsson is a traffic trace provided by Ericsson Research in Hungary (Szabó et al. 2008). DKU1, DKU2, and DKU3 are collected in Dankook University. These three traces of traffic have payloads that are captured in a controlled environment and are labeled with the actual application types.

Three malicious datasets (Bot1, Bot2, and Bot3) contain P2P botnet traffic generated by P2P botnets. Bot1 includes Storm and Waledac traffic (Saad et al. 2011a). Bot2 consists of Waledac, Conficker, and Storm traffic (Li et al. 2012). And bot3 is composed of C&C traffic of Bredolab, Kelihos-hlux, and Zeus (Guntuku et al. 2013; Singh et al. 2014). We mix these to make three datasets (Set1, Set2, and Set3) to evaluate our scheme, as shown in Table 5.

4.3 Evaluating the implementations of two-stage schemes

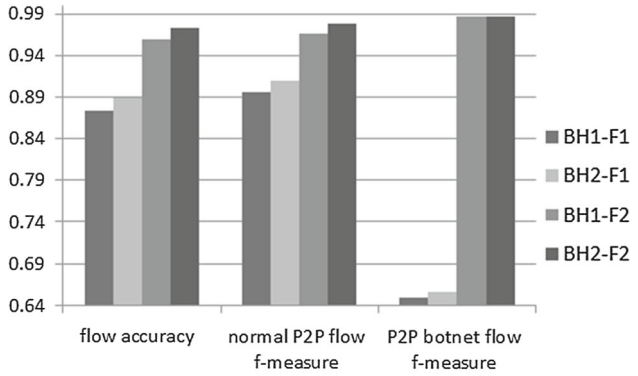
We evaluate several possible implementations of the proposed scheme to show how the heuristics rules and the flow features related to packet size improve the performance of the two-stage hybrid scheme when classifying P2P and P2P botnet traffic. In the first stage, we compare two P2P traffic classifiers, and in the second stage, we test two different P2P botnet traffic classifiers built using two different feature sets (F1 and F2). F1 has arrival time-related features, and F2 has fine-grained features. Table 6 provides four possible two-stage hybrid schemes, where BH2-F2 is our proposed scheme. Figure 9 shows a comparison of the results.

The BH2-F1 and BH2-F2 schemes perform better than BH1-F1 and BH1-F2 since the connection heuristics are added to effectively find more unknown P2P traffic. By applying an ensemble algorithm, the BH1-F1 and BH1-F2 schemes learn too much from the training dataset, especially for P2P botnet traffic, which increases the misclassification of small flows. For example, some HTTP flows that have similar characteristics as P2P botnet flows are misclassified.

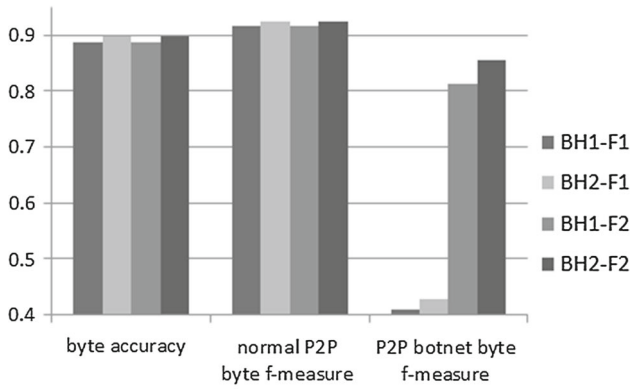
The accuracy of schemes with F1 are the lowest since the features related to arrival time are very sensitive to the network conditions, such as the congestion level, routing policy, buffer management, and monitor's location. The schemes with F2 achieve higher accuracy and f-measures when classifying P2P botnet traffic because the initial exchange of packets tends to be unique for P2P botnet traffic.

Table 6 Hybrid schemes to classify P2P and P2P botnet traffic

Scheme	First stage		Second stage
	P2P traffic classifier		P2P botnet traffic classifier
	First step	Second step	
BH1-F1		Pattern heuristics	REPTree with F1
BH1-F2	Signature	REPTree with coarse-grained features Random subspace ensemble	REPTree with F2
BH2-F1	Signature	Pattern heuristics	REPTree with F1
BH2-F2	Connection heuristics	REPTree with coarse-grained features	REPTree with F2

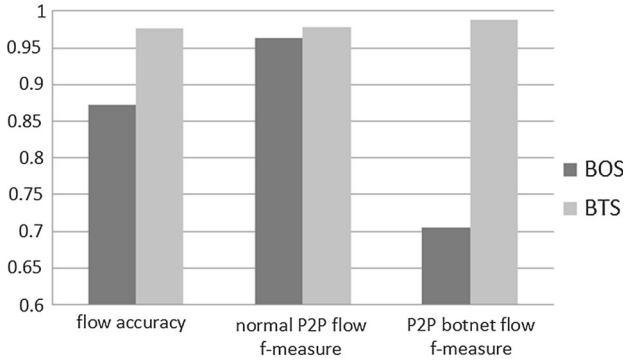


(a) Flow accuracy and f-measures of different schemes with different features

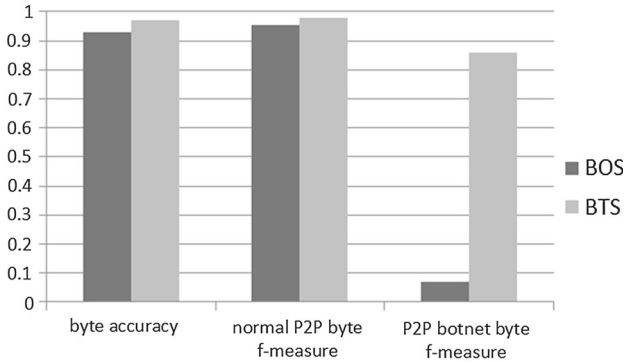


(b) Byte accuracy and f-measures of different schemes with different features

Fig. 9 Comparison of different two-stage schemes



(a) Flow accuracy and f-measures of schemes



(b) Byte accuracy and f-measures of schemes

Fig. 10 Comparison of scheme BOS and BTS

Thus, BH2-F2, our proposed scheme, exhibits the best performance for classifying normal P2P traffic and P2P botnet traffic.

4.4 Validation of proposed two-stage scheme

To validate the superiority of the BH2-F2 scheme (also referred to as BTS), we compare it with a single-stage BOS scheme. BOS directly classifies Internet traffic as non-P2P, normal P2P, or P2P botnet traffic by applying a signature-based classifier, connection heuristics, a statistics-based classifier, and pattern heuristics. Its statistics-based classifier is the C0 classifier described in Sect. 3.5.

As shown in Fig. 10, the proposed BTS scheme provides higher performance than the BOS scheme. In BTS, the first stage filters most of the non-P2P traffic and classifies most of the P2P traffic. Then, the P2P botnet traffic is separated well from normal P2P traffic in the second stage. Due to the low error correlation of two stages, BTS decreases the final error rate for P2P botnet traffic detection and overcomes the class imbalance problem, achieving a higher accuracy.

Table 7 Confusion matrices of classification results in the scheme BTS

(a) Confusion matrix in the first stage

			Predicted class		Total flow
			Non-P2P	P2P	
Actual class	Non-P2P		13906	483	14389
	P2P	Normal P2P	347	21248	21595
		P2P botnet	115	9143	9258

(b) Confusion matrix in the second stage

			Predicted class		Total flow
			Non-P2P botnet	P2P botnet	
Actual class	Non-P2P botnet	Non-P2P	478	5	483
		Normal P2P	21236	12	21248
	P2P botnet		85	9058	9143

Table 7 provides the corresponding confusion matrices of classification results in the scheme BTS. As shown in Table 7(a), 96.64 % of non-P2P flows are filtered and most P2P botnet flows are classified as P2P since P2P botnet traffic shows similar characteristics of coarse-grained features with P2P traffic in the first stage. Then, most P2P botnet flows are detected from classified P2P flows according to the fine-grained features in the second stage, as shown in Table 7(b).

4.5 Comparing with other existing schemes

As shown in Table 8, our proposed scheme classifies P2P botnet traffic with an accuracy of 97.70 % in terms of flow and 97.06 % in terms of bytes, which is higher than that of existing schemes. The SVM classifier used by Barthakur et al. (2012) achieves a high TP rate for the P2P botnet, but the P2P botnet traffic in its testing set is only generated by one P2P botnet malware.

4.6 Evaluating the amounts of computation

In order to evaluate the amount of computation that is necessary, the number of packets that are processed and the number of flows that are classified in each scheme are analyzed, as shown in Fig. 11. In the figure, BSM is a scheme combining BS (signature-based scheme) and BM (statistics-based scheme) directly, and BTS is our proposed two-stage scheme. BS only works at the packet level, and BM only operates at the flow level. With connection heuristics, our BTS scheme greatly reduces the number of processed packets by 36.51 % at the packet level and 59.48 % at the flow level. The number of flows to be classified decreases greatly, by 43.20 % at the first stage and 59.57 % at the second stage.

In our proposed scheme, the total process time (T_{process}) of the hybrid scheme consists of the time to inspect packet payloads ($T_{\text{pkt\_sig}}$) by signature-based classifier, the time to collect flow features ($T_{\text{pkt\_statistics}}$), and the time to classify traffic flows ($T_{\text{flow\_statistics}}$) by using a statistics-based classifier. We denote the total process time for BSM and BTS as follows:

$$T_{\text{BSM process}} = T1_{\text{pkt\_sig}} + T1_{\text{pkt\_statistics}} + T1_{\text{flow\_statistics}} \quad (i)$$
$$T_{\text{BTS process}} = T2_{\text{pkt\_sig}} + T2_{\text{pkt\_statistics}} + T2_{\text{flow\_statistics}} \quad (ii)$$

We assume that the time to inspect each packet payload is the same, the time to collect flow features from each packet is the same, and the time to classify each flow is the same for both two schemes BSM and BTS. The ratio between $T1_{\text{pkt\_sig}}$ and $T2_{\text{pkt\_sig}}$, $T1_{\text{pkt\_statistics}}$ and $T2_{\text{pkt\_statistics}}$, $T1_{\text{flow\_statistics}}$ and $T2_{\text{flow\_statistics}}$ are about 1.57, 1.58, and 1.31, respectively, as calculated using Fig. 11. Thus, we obtain the following equation from (i) and (ii):

$$T_{\text{BSM process}} = T_{\text{BTS process}} + 0.57 \times T2_{\text{pkt\_sig}} + 0.58 * T2_{\text{pkt\_statistics}} + 0.31 * T2_{\text{flow\_statistics}} \quad (iii)$$

$T_{\text{BTS process}}$ is much less than $T_{\text{BSM process}}$ as shown in (iii). This result also suggests that our scheme exhibits lower overhead with heuristics.

5 Conclusion

Accurate P2P traffic classification has become more and more significant for network management, and separating abnormal P2P botnet traffic from normal P2P traffic is also necessary to identify P2P malware and to detect P2P botnets.

Based on an analysis of related works, we propose an improved hybrid traffic classification scheme that is composed of two stages: a P2P traffic classifier consisting of two steps and a P2P botnet traffic classifier. In the first stage, the first step involves a signature-based classifier at the packet level combined with connection heuristics, and the second step involves a statistics-based classifier with pattern heuristics at the flow level. REPTree is selected as an implementation algorithm for the statistics-based classifier, as a result of our analyses. In the second stage, the P2P botnet traffic classifier is also implemented using the same REPTree, but with different flow features to detect P2P botnet traffic from P2P traffic. The first stage filters most of the non-P2P flows and accelerates the classification in the second stage. The two stages have a low error correlation since they use different flow features to train their statistics-based classifiers. Thus, the two-stage scheme can decrease the error rate

Table 8 Comparison of schemes for classifying P2P and P2P botnet traffic

Classified traffic type	Number of stages	Reference	Methods	Flow accuracy	TP rate for P2P botnet
- P2P - Non-P2P	One	Jun et al. (2007)	- Back Propagation ANNs	83.52%	-
		Xusheng (2008)	- SVM	92.38%	-
	Two	Li et al. (2009)	- Coarse-grain classifier (c4.5) - Fine-grain classifier (signature, port)	96.03%	-
		Ye and Cho (2013)	- Signature - Connection heuristics - C4.5	97.46%	-
- P2P botnet - Normal	One	Zhao et al. (2012)	- Bayesian Network Classifier	-	Above 95%
		Barthakur et al. (2012)	- SVM Classifier	97.24%	99.8%
- Normal P2P - P2P botnet - Non-P2P	One	Saad et al. (2011b)	- Artificial Neural Network Classifier	93%	90%
		Garg et al. (2013)	- J48 classifier	-	89.5%
	Two	Our proposed scheme (BTS)	- P2P traffic classifier - P2P botnet traffic classifier	97.70% in flow 97.06% in bytes	97.84%

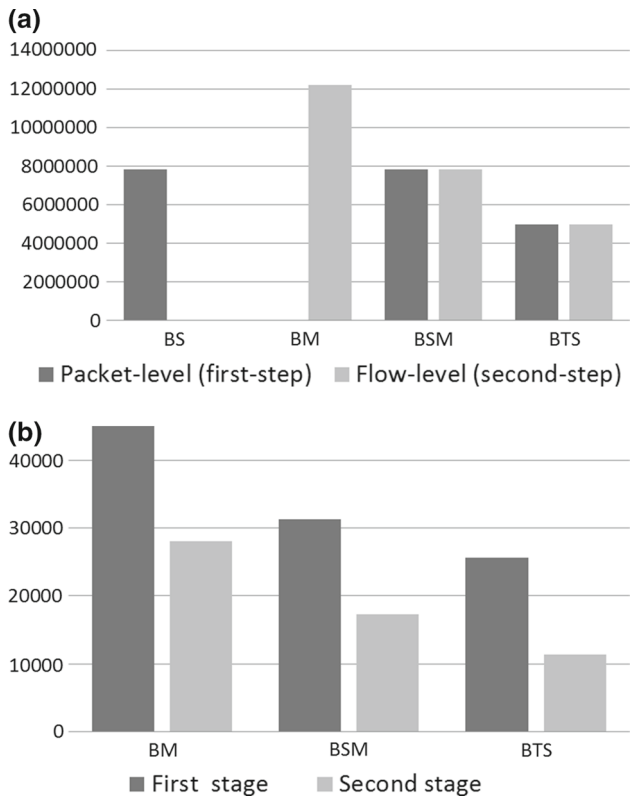


Fig. 11 Amount of computations in each scheme. *BS* signature-based scheme, *BM* statistics-based scheme, *BSM* scheme combining BS and BM directly, *BTS* our proposed two-stage scheme. **a** The number of processed packets in each scheme. **b** The number of classified flows in each scheme

when detecting P2P botnet traffic and can overcome the class imbalance problem.

The results of the analyses and experiments of this study indicate that the proposed scheme exhibits a lower overhead and achieves a higher flow accuracy of 97.70 % and byte accuracy of 97.06 % to classify P2P and P2P botnet traffic, as compared to existing schemes.

In general, existing hybrid classification schemes require a large amount of computation and time-consuming. To overcome these limitations, several methods are tried in our proposed scheme. Connection heuristics are used in the first stage. The connection heuristics reduce the amount of packets processed that need to be processed during analysis and flow feature collection by 36.51 and 59.48 %. The flows needed for classification are greatly reduced by 43.20 % in the first stage and 59.57 % in the second stage. The first stage filters most of the non-P2P flows and accelerates the second-stage classification. Since the flow features in the second stage are collected in the first stage, we do not have to collect the flow features again in the second stage. Early classification can be applied because the flow features used by the statistics-based classifier and the P2P botnet traffic classifier are packet size-related features, which can be obtained before a flow has completed.

There are still some limitations to our proposed scheme. The statistics-based classifier can be expanded to classify particular P2P traffic generated by different P2P applications. In the same way, the P2P botnet traffic classifier can also be expanded to detect specific P2P botnet traffic generated by different malwares in a P2P botnet. In addition, only TCP traffic is considered in this paper, and the traffic dataset can be extended to include UDP traffic of P2P and P2P botnet.

Acknowledgements The present research was conducted by the research fund of Dankook University in 2015.

Compliance with ethical standards

Conflict of interest This research was supported by the research fund of Dankook University in 2015.

References

Barthakur P, Dahal M, Ghose MK (2012) A framework for p2p botnet detection using svm. In: 2012 International conference on cyber-enabled distributed computing and knowledge discovery (CyberC), IEEE, pp 195–200

Bernaille L, Teixeira R, Salamatian K (2006) Early application identification. In: Proceedings of the 2006 ACM CoNEXT conference, ACM, p 6

Castiglione A, De Prisco R, De Santis A, Fiore U, Palmieri F (2014) A botnet-based command and control approach relying on swarm intelligence. J Netw Comput Appl 38:22–33

Chen Z, Yang B, Chen Y, Abraham A, Grosan C, Peng L (2009) Online hybrid traffic classifier for peer-to-peer systems based on network processors. Appl Soft Comput 9(2):685–694

Chiou TW, Tsai SC, Lin YB (2014) Network security management with traffic pattern clustering. Soft Comput 18(9):1757–1770

Dittrich D, Dietrich S (2008) P2p as botnet command and control: a deeper insight. In: 3rd International conference on malicious and unwanted software, 2008. MALWARE 2008. IEEE, pp 41–48

Elhalabi MJ, Manickam S, Melhim LB, Anbar M, Alhalabi H (2013) A review of peer-to-peer botnet detection techniques. J Comput Sci 10(1):169

Erman J, Mahanti A, Arlitt M, Cohen I, Williamson C (2007a) Offline/realtime traffic classification using semi-supervised learning. Perform Eval 64(9):1194–1213

Erman J, Mahanti A, Arlitt M, Williamson C (2007b) Identifying and discriminating between web and peer-to-peer traffic in the network core. In: Proceedings of the 16th international conference on World Wide Web, ACM, pp 883–892

Este A, Gringoli F, Salgarelli L (2009) On the stability of the information carried by traffic flow features at the packet level. ACM SIGCOMM Comput Commun Rev 39(3):13–18

Garg S, Singh AK, Sarje AK, Peddoju SK (2013) Behaviour analysis of machine learning algorithms for detecting p2p botnets. In: 2013 15th International conference on advanced computing technologies (ICACT), IEEE, pp 1–4

Gringoli F, Salgarelli L, Dusi M, Cascarano N, Risso F et al (2009) Gt: picking up the truth from the ground for internet traffic. ACM SIGCOMM Comput Commun Rev 39(5):12–18

Guntuku SC, Narang P, Hota C (2013) Real-time peer-to-peer botnet detection framework based on bayesian regularized neural network. arXiv preprint arXiv:13077464

He H, Che C, Ma F, Luo X, Wang J (2008) Improve flow accuracy and byte accuracy in network traffic classification. In: Advanced intelligent computing theories and applications. With aspects of artificial intelligence, 4th ICIC-2008, vol 5227. Springer, Heidelberg, pp 449–458

He J, Yang Y, Wang X, Zeng Y, Tang C (2014) Peersorter: classifying generic p2p traffic in real-time. In: 2014 IEEE 17th International conference on computational science and engineering (CSE), IEEE, pp 605–613

Jiang H, Shao X (2012) Detecting p2p botnets by discovering flow dependency in C&C traffic. Peer-to-Peer Netw Appl 7(4):320–331

Jpcap (2007) Jpcap introduction. https://github.com/jpcap/jpcap

Jun L, Shunyi Z, Shidong L, Ye X (2007) P2p traffic identification technique. In: 2007 International conference on computational intelligence and security, IEEE, pp 37–41

Karagiannis T, Broido A, Faloutsos M, et al (2004) Transport layer identification of p2p traffic. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ACM, pp 121–134

Keralapura R, Nucci A, Chuah CN (2010) A novel self-learning architecture for p2p traffic classification in high speed networks. Comput Netw 54(7):1055–1068

Kheir N, Wolley C (2013) Botsuer: suing stealthy p2p bots in network traffic through netflow analysis. In: Cryptology and network security, vol 8257, Springer, pp 162–178

Li H, Hu G, Yuan J, Lai H (2012) P2p botnet detection based on irregular phased similarity. In: Proceedings of the 2012 second international conference on instrumentation. Computer, communication and control, IEEE Computer Society, Measurement, pp 79–82

Li J, Zhang S, Lu Y, Yan J (2009) Hybrid internet traffic classification technique. J Electron (China) 26(1):101–112

Lu CN, Huang CY, Lin YD, Lai YC (2012) Session level flow classification by packet size distribution and session grouping. Comput Netw 56(1):260–272

Maly RJ, Mischke J, Kurtansky P, Stiller B (2003) Comparison of centralized (client–server) and decentralized (peer-to-peer) networking. Semester thesis, ETH Zurich, Zurich, Switzerland, pp 1–12

Narudin FA, Feizollah A, Anuar NB, Gani A (2014) Evaluation of machine learning classifiers for mobile malware detection. Soft Comput 1–15. doi:10.1007/s00500-014-1511-6

Palmieri F, Fiore U (2009) A nonlinear, recurrence-based approach to traffic classification. Comput Netw 53(6):761–773

Powers DM (2011) Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. J Mach Learn Technol 2(1):37–63

Saad S, Traore I, Ghorbani A, Sayed B, Zhao D, Lu W, Felix J, Hakimian P (2011a) Detecting p2p botnets through network behavior analysis and machine learning. In: 2011 Ninth annual international conference on privacy, security and trust (PST), IEEE, pp 174–180

Saad S, Traore I, Ghorbani A, Sayed B, Zhao D, Lu W, Felix J, Hakimian P (2011b) Detecting p2p botnets through network behavior analysis and machine learning. In: 2011 Ninth annual international conference on privacy, security and trust (PST), IEEE, pp 174–180

Silva SS, Silva RM, Pinto RC, Salles RM (2013) Botnets:a survey. Comput Netw 57(2):378–403

Singh K, Guntuku SC, Thakur A, Hota C (2014) Big data analytics framework for peer-to-peer botnet detection using random forests. Inf Sci 278:488–497

Soysal M, Schmidt EG (2010) Machine learning algorithms for accurate flow-based network traffic classification: evaluation and comparison. Perform Eval 67(6):451–467

Szabó G, Orincsay D, Malomsoky S, Szabó I (2008) On the validation of traffic classification algorithms. In: Passive and active network measurement, vol 4979, Springer, pp 72–81

Tran H, Hitchens M, Varadharajan V, Watters P (2005) A trust based access control framework for p2p file-sharing systems. In: HICSS'05. Proceedings of the 38th Annual Hawaii international conference on system sciences, 2005. IEEE, 302c pp

Tyagi AK, Aghila G (2011) A wide scale survey on botnet. Int J Comput Appl 34(9):9–22

Valdés L, Montesinos S, Ariza A, Allende SM, Joya G (2015) Peer selection in p2p wireless mesh networks: comparison of different strategies. Soft Comput. doi:10.1007/s00500-014-1572-6

Vania J, Meniya A, Jethva H (2013) A review on botnet and detection technique. Int J Comput Trends Technol 4(1):23–29

Wang B, Li Z, Tu H, Ma J (2009) Measuring peer-to-peer botnets using control flow stability. In: International conference on availability, reliability and security, 2009. ARES'09. IEEE, pp 663–669

Wang R, Tang K (2012) Minimax classifier for uncertain costs. arXiv:1205.0406

Weka (2012) Weka introduction. http://www.cs.waikato.ac.nz/ml/weka/

Xusheng Z (2008) A p2p traffic classification method based on svm. In: International symposium on computer science and computational technology, 2008. ISCSCT'08. IEEE, vol 2, pp 53–57

Ye W (2012) Two step hybrid p2p traffic classification. Master's thesis, Dankook University, Korea

Ye W, Cho K (2013) Two-step p2p traffic classification with connection heuristics. In: 2013 Seventh international conference on innovative mobile and internet services in ubiquitous computing (IMIS), IEEE, pp 135–141

Ye W, Cho K (2014a) Hybrid p2p traffic classification with heuristic rules and machine learning. Soft Comput 18(9):1815–1827

Ye W, Cho K (2014b) P2p traffic classification using advanced heuristic rules and analysis of decision tree algorithms. J Korea Soc Comput Inf 19(3):45–54

Zeng Y, Shin KG (2013) On detection of storm botnets, pp 1–7

Zhang H, Lu G, Qassrawi MT, Zhang Y, Yu X (2012) Feature selection for optimizing traffic classification. Comput Commun 35(12):1457–1471

Zhang J, Perdisci R, Lee W, Luo X, Sarfraz U (2014) Building a scalable system for stealthy p2p-botnet detection. IEEE Trans Inf Forensics Secur 9(1):27–38

Zhao D, Traore I, Ghorbani A, Sayed B, Saad S, Lu W (2012) Peer to peer botnet detection based on flow intervals. In: Information security and privacy research, 28th IFIP TC 11 SEC conference-2012, vol 376. Springer, Crete, pp 87–102