Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Flow-based intrusion detection: Techniques and challenges

Muhammad Fahad Umer [a,*], *Muhammad Sher* [a], *Yaxin Bi* [b]

[a] *Department of Computer Science & Software Engineering, Faculty of Basic and Applied Sciences, International Islamic University, Islamabad, Pakistan*

[b] *School of Computing and Mathematics, Faculty of Computing and Engineering, Ulster University, UK*

ARTICLE INFO

Article history:
Received 19 October 2016
Received in revised form 15 March 2017
Accepted 22 May 2017
Available online 1 June 2017

Keywords:
Network flows
Flow-based intrusion detection
Flow-based datasets
Attack detection
Machine learning

ABSTRACT

Flow-based intrusion detection is an innovative way of detecting intrusions in high-speed networks. Flow-based intrusion detection only inspects the packet header and does not analyze the packet payload. This paper provides a comprehensive survey of current state of the art in flow-based intrusion detection. It also describes the available flow-based datasets used for evaluation of flow-based intrusion detection systems. The paper proposes a taxonomy for flow-based intrusion detection systems on the basis of the technique used for detection of maliciousness in flow records. We review the architecture and evaluation results of available flow-based intrusion detection systems. We also identify important research challenges for future research in the area of flow-based intrusion detection.

1. Introduction

The need of always-up high-speed network services for businesses and governments cannot be over emphasized. Information technology companies and service providers make continuous efforts to increase the capacity of network links and hardware. This tremendous increase in data transfer rate, computation power, and expansion of computer networks has resulted in complex information security challenges which require alternate solutions.

Intrusion detection systems (IDS) are an important tool for the protection of IP networks. Intrusion detection systems analyze the network traffic and system logs to detect an attack. If an attack is detected, the intrusion detection system raises an alert (Garcia-Teodoro et al., 2009). Traditionally, intrusion detection systems use deep packet inspection (AbuHmed et al., 2008) or stateful protocol analysis to detect attacks in the network traffic. Deep packet inspection is difficult to implement when network traffic is encrypted (Koch, 2011). Also, inspecting the complete payload is computationally costly and can become a performance bottleneck in high-speed IP networks (Sperotto and Pras, 2011). The stateful protocol analysis checks complete semantics of protocols against the specification and any out of range value is considered an intrusion. Stateful analysis techniques are protocol specific and can also be computationally costly (Liao et al., 2013).

Due to limitations of the packet and protocol-based intrusion detection systems, researchers are focusing on alternative approaches to protect IP networks. An innovative solution for securing IP networks from unauthorized access is the use of flow-based intrusion detection (Copeland, 2007). Flow-based

[*] *Corresponding author.*
E-mail address: fahad.phdcs62@iiu.edu.pk (M.F. Umer).

http://dx.doi.org/10.1016/j.cose.2017.05.009
0167-4048/

intrusion detection systems use network flow records as input and try to find out if the network traffic is normal or malicious (Sperotto and Pras, 2011). Since only the flow records are inspected, the intrusion detection system is relieved from the complex and time-consuming processing of packet content inspection. The flow packets analyzed on a network are on the average equal to 0.1% of the traffic. Whereas the network load is measured in bytes, the overhead due to a flow collection and export protocol (Netflow) is on average 0.2% (Sperotto and Pras, 2011). Therefore, flow-based intrusion detection process is fast as compared to packed-based inspection. Another advantage of flow-based intrusion detection is its independence from encapsulated payload.

The flow-based intrusion detection is a relatively new field, and research in this area is gaining momentum. Many techniques have been proposed in recent years that use network flow data for intrusion detection. In this paper, we review state of the art in flow-based intrusion detection systems and discuss open issues and future research challenges.

We organize the paper as follows. We discuss related survey and review articles in Section 2. We also describe how our study is different from the existing review articles. Section 3 gives an introduction to flow-based intrusion detection and discusses its pros and cons. Section 4 briefly outlines different flow-based datasets used for performance evaluation of the flow-based system. In Section 5, we propose a taxonomy for intrusion detection systems based on the technique used for attack detection in network flows. Then we review the available flow-based intrusion detection systems in each class of method step by step. We also give a list of commercially available flow-based intrusion detection systems in Section 6. In Section 7, we present our observations on the existing techniques. We identify important challenges and open issues in the field of flow-based intrusion detection and describe them in Section 8. In Section 9, we discuss the future of flow-based intrusion detection systems. Finally, we summarize our work in Section 10.

2. Related work

Intrusion detection is a prominent research area. A number of survey and review articles covering state of the art in intrusion detection have been published. In this section, we briefly describe the important intrusion detection review articles and discuss the significance of our work.

We classify the available intrusion detection review articles into scenario-based, technique-based, attack-based and general-purpose categories. Scenario-based review articles analyze all types of intrusion detection systems designed for a specific network scenario or architecture. Anantvalee and Wu (2007) discuss intrusion detection techniques used for Mobile Ad-hoc Networks (MANETs). The authors reviewed and compared the existing systems and provided directions for future research. Another survey for intrusion detection and prevention system in MANETs is presented in Nadeem and Howarth (2013). Patel et al. (2013) give a survey of intrusion detection and prevention systems (IDPS) in cloud computing. The authors describe the characteristic of cloud computing and discuss the challenges faced in the development of IDPS for the cloud. The article also identifies the requirements for a cloud-based IDPS. A survey of intrusion detection system in Wireless Sensor Networks (WSNs) is given in (Butun et al., 2014). The article includes a brief survey of IDSs and discusses their applicability to WSNs. The authors give a detailed review on IDSs devised for WSNs with their advantages and disadvantages. The survey also provides a general model for an IDS applicable to WSNs.

Technique-based review articles analyze a particular type of intrusion detection systems on the basis of detection algorithm. In technique-based reviews, a taxonomy of detection algorithms is constructed and available intrusion detection systems are reviewed for every category. Technique-based review articles are helpful in performance comparison of different detection algorithms. Garcia-Teodoro et al. (2009) present a review of anomaly-based network intrusion detection systems (A-NIDS). The authors classify the A-NIDS into statistical, knowledge based and machine learning based categories. The article reviews the A-NIDS techniques for every category and discusses their pros and cons. A list of commercially available A-NIDS is also given. In the end, the article discusses the open issues and challenges posed by the anomaly detection systems. Another survey of anomaly detection methods in computer networks is presented by Zhang et al. (2009). The survey organized anomaly detection techniques in four categories: Statistics, Classification, Machine Learning, and Finite State Machines. The article describes available techniques in each of these categories. Tsai et al. (2009) give a survey of intrusion detection systems using machine learning techniques. The authors compare the intrusion detection systems on the basis of classifier design, datasets used, and other experimental setups. The article also provides limitations in developing intrusion detection and also discusses future research directions. A survey of intrusion detection systems using computational intelligence techniques is given in Wu and Banzhaf (2010). The survey reviewed the application of artificial neural networks, fuzzy systems, evolutionary computation, artificial immune systems, swarm intelligence, and soft computing algorithms for intrusion detection. A survey of data mining and machine learning methods for intrusion detection is given by Buczak and Guven (2015). The article categorizes the available systems under twelve different methods of data mining and machine learning. The Computational complexity of different methods has also been described. In the end, the article gives recommendations for use of data mining and machine learning techniques in intrusion detection. A survey of flow-based intrusion detection techniques is presented by Drašar et al. (2014). The survey reviews the flow-based techniques using similarity matching for attack detection. The techniques are grouped on the basis of order of similarity functions. Vasilomanolakis et al. (2015) present a survey of collaborative intrusion detection systems (CIDS). The authors define requirements for the successful deployment of CIDS in large IT systems and critical infrastructures. The available CIDSs are classified into centralized, decentralized, and distributed classes; and a detailed survey of techniques in every class is presented.

Attack-based review articles establish a taxonomy of network attacks. For every type of attacks, the available intrusion detection systems are reviewed. Sperotto et al. (2010) give detailed survey of flow-based intrusion detection systems. The article gives an introduction of flow-based intrusion system and also describes the motivation behind the use of flow-based intrusion detection. A taxonomy of network attacks is created and flow-based techniques addressing the attack types are described. In the conclusion, the authors critically discuss the flow-based intrusion detection and identify future research directions.

General-purpose review articles provide state of the art in intrusion detection in different dimensions. A survey of intrusion detection and prevention systems is presented in Patel et al. (2010). In the survey, deficiencies in the existing systems are analyzed and use of intelligent techniques such as machine learning and autonomic computing is proposed for detection of known and unknown threats. A comprehensive review of intrusion detection systems is presented in Liao et al. (2013). The article proposes a taxonomy of intrusion detection systems based on the system deployment, data source, timeliness and detection strategy. Some future challenges for intrusion detection systems have also been presented. Bhuyan et al. (2014) provide an extensive review of network anomaly detection methods, systems and tools. The article identifies six different categories of network anomaly detection methods. The authors have described the advantages and disadvantages for each class of methods and discussed the related systems. The article also gives details of evaluation measures and datasets used for benchmarking of intrusion detection systems. In the end, an extensive discussion on open issues and challenges in network anomaly detection is also given.

In this paper, we provide an up-to-date technique-based review of the flow-based intrusion detection systems. Our work differs from the existing survey and review papers in following aspects:

1. We give a comprehensive coverage of flow-based detection techniques. An earlier survey of flow-based techniques, conducted by Sperotto et al. (2010), is now seven years old. A recent survey of flow-based detection technique was conducted by Drašar et al. (2014). However, it focuses on similarity matching methods. A number of techniques like that of Winter et al. (2011b), Zhang et al. (2012), and François et al. (2012) have not been discussed.
2. The focus of our work is flow-based intrusion detection. Other surveys like Bhuyan et al. (2014) and Buczak and Guven (2015) discuss some flow-based techniques but with limited details.
3. We give a brief introduction of flow-based intrusion detection. We describe a general flow-based intrusion detection model and discuss the pros and cons of flow-based intrusion detection.
4. We present a summary of publicly available flow-based intrusion datasets. We also describe the process used for generation of dataset and give details of important flow attributes.
5. We create a technique-based taxonomy of flow-based intrusion detection systems. This is different from the earlier survey papers which were organized on the basis of attack types. We review the available flow-based intrusion detection techniques in every class of methods and discuss their advantages and disadvantages.
6. We provide a list of commercially available intrusion detection systems which use flow data for network attack detection.
7. In the end, we identify important open issues and research challenges in flow-based intrusion detection.

3. Flow-based intrusion detection

3.1. *Network flows*

Flow-based intrusion detection systems use network flow data for intrusion detection. The network flow data have a number of applications e.g. billing, traffic analysis, network visibility, congestion control, and intrusion detection (Li et al., 2013). A network flow or a flow is defined as a set of packets or frames passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties (Trammell and Claise, 2013). The observation points in the network can be flow probes or flow enabled network devices. The information of network flow is stored in a flow record. The processing of flow records in a network is managed through a flow export and collection protocol. The importance of flow data has led all major vendors to offer built-in flow collection and export support in their network hardware. Different vendors have their own flow protocols but Cisco Netflow has been the most popular.

The Internet Engineering Task Force (IETF) has adopted Netflow version 9 for the development of a standard flow export and collection protocol, named Internet Packet Flow Information Exchange (IPFIX) (Trammell and Claise, 2013). IPFIX is a flexible protocol with around 280 attributes. IPFIX allows export of flow records in a custom format defined by the export template. Unlike Netflow, IPFIX contains specific fields which can be used by vendors to store proprietary information. Fig. 1 shows the IPFIX flow export and collection architecture. The IPFIX architecture consists of following three processes:

- **Observation points with a metering process.** Observation points collect the packets passing through specific interfaces. These packets are forwarded to a metering process. The metering process timestamps the packets. These timestamped packets can be sampled or filtered because the total number of packets can be very large in a high speed network. These packets are cached for specific intervals such that all packets required for a specific flow are received.
- **Exporting process.** The rules for generating IPFIX flow records are defined in an exporting process. The process generates the IPFIX records in the format defined by an IPFIX template and forwards them to the collecting process using the underlying transport protocol.
- **Collecting process.** The collecting process collects IPFIX records from exporting process and stores them in a flow database. The database is accessible to the flow-analysis applications for required purpose.

3.2. *Architecture of flow-based intrusion detection system*

Fig. 2 shows the general architecture of a flow-based intrusion detection system. The system takes IPFIX/Netflow records as input. The flows records can have many attributes. However, not all of these attributes will be required in the attack detection

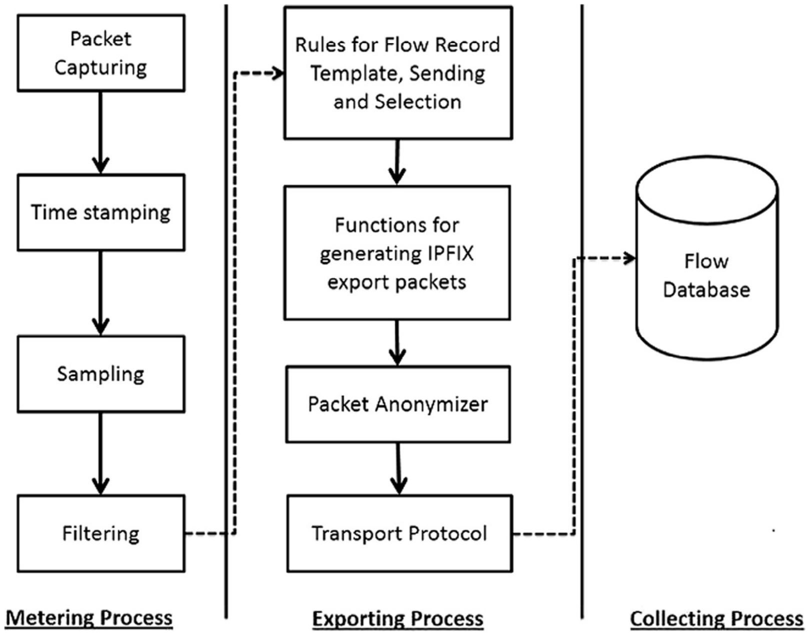

Fig. 1 – IPFIX process architecture.

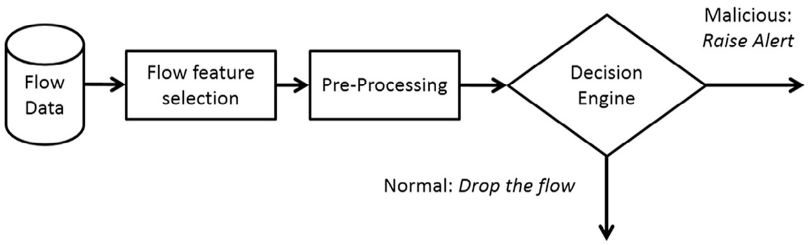

Fig. 2 – A general flow-based intrusion detection model.

decision. The feature selection phase only selects relevant attributes required for decision making. A pre-processing phase converts the flow records in a specific format which is acceptable to the detection algorithm. In the detection phase, the algorithm marks the flow records as malicious or normal. If the flow is normal, it is considered safe and dropped with no subsequent action while malicious flow can raise an alert and become the subject of further inspection.

3.3. *Pros and cons of flow-based intrusion detection*

Flow-based intrusion detection has a number of advantages over traditional intrusion detection systems. The flow-based IDS only analyzes network flow records. The flow records contain aggregated information of packet headers. The network traffic information is summarized in the form of IP flows and the amount of data processed by the IDS is reduced. Flow-based intrusion detection is, therefore, best suited for protection of backbone links where processing of complete network traffic is computationally difficult (Sperotto and Pras, 2011).

A number of modern network applications use end-to-end encryption. It is not possible to inspect the encrypted data at an intermediate location by a packet-based intrusion detection system. In this case, flow-based intrusion detection is an appropriate choice because no packet data scanning is required. Flow-based inspection has fewer privacy concerns than packet-based inspection because user information is protected from any intermediate scan.

The collection of flow data from the network can easily be distributed among multiple flow collection points. Most of the latest hardware offers built-in flow-collection support. Thus flow data can be collected from multiple locations across the network with any additional cost (Golling et al, 2014). The collection of flow records also has additional usages such as billing, congestion control, and network behavioral analysis (Hofstede et al., 2014a). At best, flow-based intrusion detection systems have a near real-time response, low deployment cost, and the ability to operate on high-speed backbone links (Golling et al, 2014).

With some excellent benefits, flow-based intrusion detection also has some drawbacks. The IP flow records used for intrusion detection contain generalized network information. It is therefore difficult for the flow-based IDS to distinctly detect an attack using the generalized information. Flow-based techniques do not scan the packet payload. Therefore flow-based techniques cannot detect the network attacks hidden in the packet payload and are not as accurate as packet-based detection (Sperotto and Pras, 2011).

4. Flow-based intrusion datasets

The intrusion datasets are used for benchmarking the performance of intrusion detection systems (IDS). The datasets contain both normal and malicious network traffic. The IDS detects the malicious traffic present in the dataset. The performance of an IDS is evaluated by the actual number of attacks and the number of attacks detected by the IDS. Publicly available datasets make it easier to obtain symmetrical results for comparing different IDS. The intrusion datasets are generated in following two ways (Marek Malowidzki and Mazur, 2015):

- A laboratory environment is set up to simulate the different network scenarios. The attacks are artificially launched using scripts and traffic samples are collected from the network. This type of dataset is easy to develop, and all attack types can be manually injected. However, such datasets do not represent the real-world network traffic scenario. The intrusion detection systems evaluated on such datasets are not guaranteed to give similar results in a real-world deployment.
- Another way to create intrusion datasets is to collect traffic samples from real-world networks. These datasets represent the actual nature of network traffic. However, these datasets may not contain all required types of attacks. The realistic datasets are difficult to build. Companies and enterprises do not permit collection of traffic samples from their network due to confidentiality and privacy issues. Also, legal laws do not allow publishing of actual data in public domain. Usually, user related information is removed from the dataset to address the privacy issues.

A taxonomy of intrusion datasets is given in Bhuyan et al. (2014). Most of the intrusion datasets included in the taxonomy are packet based, except TUIDS, which is packet and flow-based (Gogoi et al., 2012). Packet-based intrusion dataset can also be used for evaluation of flow-based IDS. Due to the importance of flow-based intrusion detection, researchers are now developing native flow-based intrusion datasets. The flow-based datasets are available in the form of network flow records. In the following section, we give brief summaries of the available flow-based datasets.

4.1. *Sperotto intrusion dataset*

The Sperotto dataset is the first publicly available flow-based dataset (Sperotto et al., 2009). It consists of 14.2 M flow records collected through a "Honeypot" deployment in University of Twente campus network. Four standard services SSH, HTTP, FTP and AUTH/IDENT were run over the honeypot for six days. During the flow collection, one hacker installed an IRC proxy over the honeypot which has also generated some traffic. Both traffic dump and the services log file were downloaded and passed through a correlation process for the alert generation. The correlation process succeeded in labeling more than 98.5% of the flows and 99.99% of the alerts. The dataset is available in the form of Netflow version 5 records. Table 1 shows the number of flows in the dataset.

Table 1 – Details of flow records – Sperotto intrusion dataset.

Traffic Type	Number of Flows	Category
SSH	13,942,629	Malicious
FTP	13	Malicious
HTTP	9,798	Malicious
AUTH-INDET	19,1339	Side-effect
IRC	7,383	Side-effect
OTHERS	18,970	Side-effect

4.2. *ISOT intrusion dataset*

The ISOT dataset consists of several existing publicly available malicious and non-malicious datasets (Szabó et al., 2008). The malicious portion contains malicious traffic of Storm and Waledac botnets. The normal traffic is the combination of two existing datasets obtained from the Traffic Lab at Ericsson Research in Hungary (Szabó et al., 2008) and the Lawrence Berkeley National Lab (LBNL). The Ericsson Lab dataset has a variety of normal traffic including web browsing, gaming and torrent traffic. The LBNL trace data consists of network trace recorded over three months and contains network traffic for web, email and streaming media applications.

All datasets are merged into each other using a special process. The resulting dataset contains 22 subnets of normal traffic from LBNL and one subnet of both malicious and normal traffic from honeypot and Ericsson Lab. The records in the dataset contain eleven attributes including seven flow-based and four host-based attributes. Table 2 shows the total number of malicious and normal flows in the dataset.

Table 2 – Details of flow records – ISOT intrusion dataset.

Type	No of Unique flows
Malicious	55,904 (3.33%)
Non-malicious	1,619,520 (96.66%)
Total	1,675,424 (100%)

4.3. *TU intrusion dataset*

TUIDS (Tezpur University Intrusion Dataset) is a packet and flow-based dataset generated in a laboratory environment in Tezpur University (Gogoi et al., 2012). The experimental setup used for generation of dataset consists of one router, one layer-three switch, two layer-two switches, one server, two workstations and forty nodes. The attacks are generated against different nodes. Another LAN of 350 nodes was also connected with the experimental setup. The attacks are launched from the LAN and also within the setup.

The dataset contains both packet and flow-based data. The flow-based dataset is in the form of Netflow version 5 records. The flow records have 16 basic attributes, four time-windows attributes and four connection based attributes. The details of

Table 3 – Details of flow records – TU intrusion dataset.

Category	Training dataset	Testing dataset
Normal flows	23,120	16,770
Attack flows	29,723	23,955
Total	52,843	40,725

normal and malicious records in the flow-based dataset is given in Table 3.

4.4. *CTU-13 dataset*

The CTU-13 dataset was created in CTU University, Czech Republic (Garcia et al., 2014). The dataset consists of botnet traffic mixed with normal and background communication traffic. The traffic capture process consists of 13 different scenarios where a particular malware traffic was captured in each scenario. The environment for traffic capture consists of virtual machines running the Microsoft Windows XP SP2 operating system on top of a Linux Debian host. These virtual machines were bridged into the University network. The traffic was captured both on the Linux host and on the university network router connected to the Linux host. During the labeling process, all traffic was initially given the background label. The normal label was given to the traffic that was originated from switches, proxies, and legitimate computers. All traffic that came from the known infected machines was labeled botnet. The CTU dataset contains bidirectional Netflow records. Table 4 gives details of the malware and traffic flow records in each scenario.

4.5. *SSHCure intrusion dataset*

SSHCure intrusion dataset consists of SSH compromise attacks on a campus network (Hofstede et al., 2014b). The dataset was exported from the four Cisco 6500 series routers in a Netflow v5 records. The dataset has two segments, D1 and D2. Both segments were collected over a period of one month on the campus network of the UT. The two segments reflect two different scenarios. The D1 segment consists of SSH traffic targeting honeypots. The D2 segment has the SSH data from normal servers. The D1 and D2 segments have 632 and 10,716 attacks respectively. The ground truth for the dataset is obtained from the corresponding log files of servers and honeypots.

5. Techniques in flow-based intrusion detection

Flow-based intrusion detection systems have been designed using different techniques. We have created a taxonomy of flow-based intrusion detection systems on the basis of detection method. Fig. 3 shows the taxonomy hierarchy. We classify the flow-based intrusion detection systems into *statistical*, *machine learning*, and *other* techniques. In the following sections, we review the architecture and performance results of available flow-based intrusion detection systems in each category.

5.1. *Statistical techniques*

Statistical methods build a profile of the normal network traffic using a statistical function of network traffic parameters. This profile of the normal traffic is used to check the unseen incoming traffic. The similarity of the network traffic and profile of normal network traffic is calculated using statistical measures. If the similarity measure is above the pre-defined threshold, the flow is marked malicious or normal otherwise (Liao et al., 2013; Qayyum et al., 2005). We further categorize the statistical techniques into univariate, multivariate and time-series methods.

5.1.1. *Univariate statistical techniques*

Univariate statistical techniques analyze a single variable at a time e.g., mean and standard deviation. These techniques assume an underlying known distribution of data. A flow based system to detect TCP port scans is presented in Muraleedharan and Parmar (2010). TCP port scanning is the first step in launching an attack and attackers use TCP scans to determine the port numbers of critical user services. The authors construct long-term and short-term profiles of TCP scans. The long-term and short-term profiles have different parameters of IP flows with their statistical mean and standard deviation. These parameter values are used as a threshold to detect a TCP scan. The authors compared the proposed system with

Table 4 – Detail of flow records – CTU-13 intrusion dataset.

Id.	Bot	Characteristic	Total Flows	Botnet Flows	Normal Flows	Background Flows
1	Neris	IRC, SPAM, Click Fraud	2,824,636	39,933	30,387	2,753,290
2	Neris	IRC, SPAM, Click Fraud, FTP	1,808,122	18,839	9,120	1,778,061
3	RBot	IRC, Port Scan, US	4,710,638	26,759	116,887	4,566,929
4	RBot	IRC, DDOS, US	1,121,076	1,719	25,268	1,094,040
5	Virut	SPAM, Port Scan, HTTP	129,832	695	4,679	124,252
6	Mentri	Port Scan	585,919	4,431	7,494	546,795
7	Sogou	HTTP	144,077	37	1,677	112,337
8	Merli	Port Scan	2,954,230	5,052	72,822	2,875,282
9	Neris	IRC, SPAM, Port Scan, Click Fraud	2,753,884	17,880	43,340	2,525,565
10	Rbot	IRC, DDOS, US	1,309,791	106,315	15,847	1,187,592
11	RBot	IRC, DDOS, US	107,251	8,161	2,718	96,369
12	NSIS.ay	PP	325,471	2,143	7,628	315,675
13	Virut	SPAM, PS, HTTP	1,925,149	38,791	31,939	1,853,217

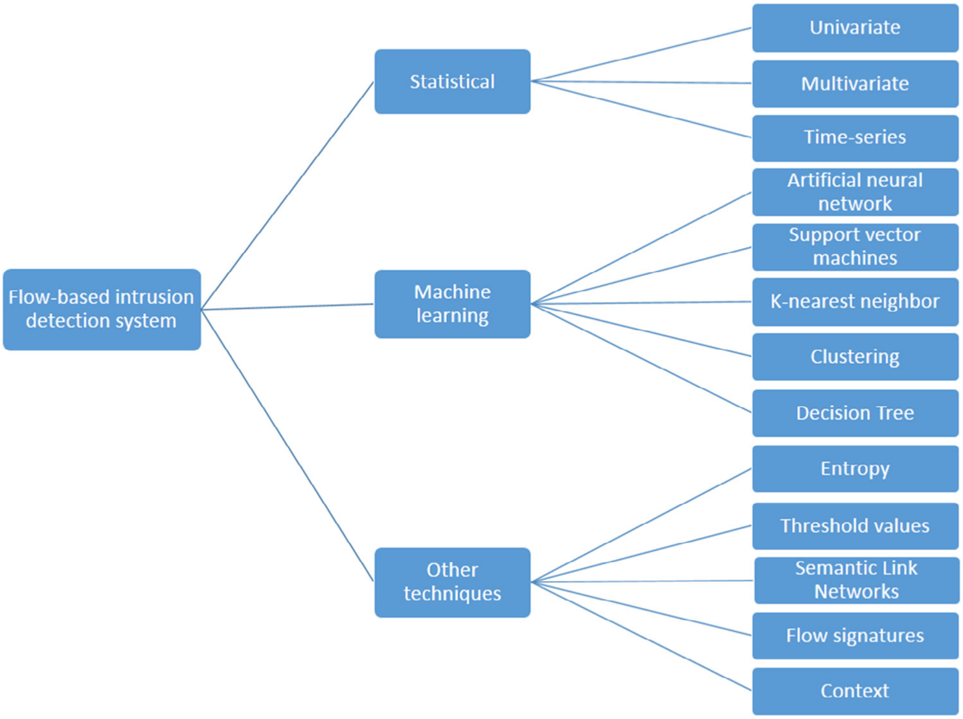

Fig. 3 – Taxonomy of flow-based intrusion detection systems.

a well known IDS Snort for detection of TCP scans in the incoming flows. The flow data for the evaluation are generated from a live network connected over the Internet. The results show that the proposed system detects all 13 types of scans as compared to Snort, which can only identify 8 types of scans. The proposed technique is suitable for preliminary passive protection of a network. It cannot detect TCP scans which stealth themselves as legitimate network traffic and keep the mean and standard deviation values under the threshold.

A technique to detect flooding attacks in backbone network traffic is proposed in Salem et al. (2011). The traffic traces are aggregated into flow records using sketch data structure. Least Mean Square (LMS) filter and Pearson Chi-square deviation are used to detect changes in the flow records. The authors used the MAWI dataset for evaluation (Fontugne et al., 2010). The results show that the proposed approach outperforms other divergence techniques and achieve a detection rate of 100% with false alarm rate of 3.8%. However, these results are obtained by converting a packet-based dataset to custom flows and standard Netflow/IPFIX flow records are not used as input.

A novel flow-based metric, Congestion Participation Rate (CPR), to detect low-rate DDoS (LDoS) attacks is proposed in Zhang et al. (2012). The CPR of a flow F is defined as the ratio of incoming packets in congestion to the total incoming packets of that flow. A high CPR value means there is a greater probability that flow is malicious. All flows with CPR greater than the predefined threshold are classified as malicious and dropped. The authors have conducted validation experiments using NS2 simulations, test-bed experiments, and LBNL/ICSI enterprise traces. In all experiments, the calculated CPR is under range for normal flows and becomes high in case of LDoS attack. The CPR has the advantage of detecting small scale slow-ramped attacks and can be used in integration with other intrusion detection techniques.

A solution for detection of DNS tunnels using flow records is proposed in Ellens et al. (2013). In DNS tunneling, another protocol or payload is tunneled through DNS packets. DNS tunneling can pose a significant risk to the network. In this paper, eight flow-based variables are derived from the flow record which acts as indicative of DNS tunneling. Three different anomaly detection techniques, threshold method, Brodsky-Darkhovsky method and distribution-based method, have been used to evaluate the traffic characteristic using the flow-based variables. The approach is validated using various datasets and the results show that the method can detect different tunnel usage scenarios with a high detection rate.

The IPFIX/Netflow export process exports flow records after certain time interval. During the time interval, short-lived attacks can persist and are not detected until the flow records are exported and processed by the intrusion detection system. A solution for real-time intrusion detection of DDoS attacks using NetFlow and IPFIX is proposed in Hofstede et al. (2013). The authors extend the IPFIX/Netflow export process and directly connect it with a lightweight intrusion detection module. The intrusion detection module uses a time-series forecasting based on Exponentially weighted moving average (EWMA) mean calculation. Specific metrics related to DDoS attacks are measured and compared with the forecasted value. The traffic sample is considered malicious if the measured value does not lie within the range of the forecasted value. The characteristics of malicious flows are added to a blacklist which is then used to filter the malicious traffic. The technique is validated on a dataset captured from a service provider backbone network. The detection algorithm achieves 92% detection rate and false positive rate of 0.01% for a span of 900 s. The proposed approach can be useful in detecting DDoS attack. However, performance issues can arise if multiple attacks occurred. In this case, the IDS would not be able to inspect the flow records at the speed at which they are being collected. The algorithm suggested in Hofstede et al. (2013) has also been applied for detection of DDoS attack in Cisco IOS (van der Steeg et al., 2015).

5.1.2. *Multivariate statistical techniques*

Multivariate techniques analyze the relationships between two or more variables. Multivariate techniques include Principal Component Analysis (PCA), linear discriminate analysis, and discriminate analysis.

PCA for anomaly detection in flow data is used in Kanda et al. (2013). PCA is an unsupervised learning dimensionality reduction technique. The authors used sketch structure with hashing of network traces. The hashed network traces are converted into entropy time-series and given as input to a PCA classifier. The technique uses a three-step sketch structure which helps in attaining higher detection rate and lower false positive. The technique is evaluated over nine-year traces of MAWI dataset with different parameter tuning. The variants of proposed technique show an improvement when compared to other PCA based anomaly detectors over the same dataset. The maximum accuracy obtained in the form of F1-measure by the authors is 0.90 which is lower compared to the other techniques discussed earlier. The reason for low F1-measure can be attributed to the real-time nature, variance and complexity of the dataset used.

A profile-based anomaly detection system using PCA and flow analysis is proposed in Fernandes et al. (2015). This approach creates profiles for all types of normal traffic flow which are called Digital Signature of the Network Segments (DSNSFs). The process of anomaly detection is divided into two steps: traffic characterization and anomaly detection. The traffic characterization step extracts quantitative attributes from the flow records and creates the corresponding DSNSFS using principal component analysis. The anomaly detection step creates confidence bands using DSNSFs. These bands are matched with the normal traffic signatures, and any abnormality is notified to the system administrator. The proposed system has been evaluated on a real network using sFlow flow export and collection protocol. The evaluation of anomaly detection phase achieves 94% true positive rate. However, this technique is difficult to implement in modern networks as it is not possible to generate the signatures of all network traffic in advance.

5.1.3. *Time-series statistical techniques*

Time-series based statistical techniques use previously observed values to forecast new values. Sperotto et al. (2008) used time-series analysis anomaly characterization in flow traffic. Nguyen et al. (2008) apply Holt–Winters forecasting method to detect an anomaly in the flow traffic. They use four metrics to construct a flow: total bytes, total packets, the number of flows with similar volume to the same destination socket and the number of flows that has a similar volume and same source and destination address, but to different ports. These four metrics are used to detect three types of anomalies: flooding, TCP SYN, and port scan. The Holt–Winters method keeps track of the normal metrics values and raises an anomaly flag if any value goes out of range. The technique is limited to only three anomalies and can be bypassed if the attacker keeps the flow metrics values within range.

A high-speed flow level intrusion detection system (HiFIND) is presented in Li et al. (2010). The use of flow information for high speed and DoS resilient intrusion detection was initially proposed in Li et al. (2005) and Gao et al. (2006). HiFIND uses a small set of packet header fields including Source/Destination IP and Source/Destination ports. It focuses on three types of attacks: SYN Flooding, Horizontal Scan and Vertical Scan. The authors use Holt–Winters double exponential smoothing and EWMA with season indices method for change detection in network traffic. HiFIND is applied in three phases and false-positives are reduced by separating the intrusion and network anomalies caused by misconfiguration. The performance evolution of HiFIND is carried out using both simulation and onsite deployment. A custom dataset of one-day traffic traces consisted of 900M flow records is used. The authors compared the HiFIND with other statistical detection techniques for flow-based detection, and results show that HiFIND has similar accuracy but is memory efficient in worst case scenarios. HiFIND is one of the few models to implement the intrusion detection systems security (Sadre et al., 2012). The use of Holt–Winters double exponential smoothing and EWMA with season indices can have the drawback of statistical seasoning effect. The authors only used a 4-feature NetFlow record without including the protocol field. Therefore the system may not be able to detect the attacks sent on UDP packets. Another limitation of the HiFIND system is the inability to detect small and slow-ramped attacks.

5.2. *Advantages and disadvantages of statistical techniques*

Table 5 provides summary of statistical flow-based techniques. The advantages of statistical techniques for flow-based intrusion detection are as follows:

- Statistical techniques do not require prior knowledge of network attacks (Bhuyan et al., 2014).
- These can accurately detect the attacks which cause abrupt and highly deviated changes in the network traffic e.g. DoS attacks.

Disadvantages of statistical technique are as follows:

- High dimensionality and variation in network traffic can affect the performance of statistical intrusion detection systems (Gyanchandani et al., 2012).
- It is difficult to compute the statistics of normal network traffic.
- Small and slow-ramped attacks can bypass the statistical techniques by keeping the impact of attack under statistical thresholds.

Table 5 – Summary of flow-based intrusion detection system using statistical techniques.

Article	Technique	Dataset	Performance Measure	Result
Muraleedharan and Parmar (2010)	Mean and Standard deviation	Custom	No of TCP scans detected	100%
Salem et al. (2011)	Chi Square Deviation	MAWI	Detection rate	100%
Zhang et al. (2012)	Change Detection	LBNL/ICSI	Comparison	Maximum
Ellens et al. (2013)	Change Detection	Custom	Detection rate	100%
Hofstede et al. (2013)	(EWMA) Forecasting	Detection rate	Custom	92%
Kanda et al. (2013)	PCA	MAWI	F1-Measure	0.0984
Fernandes et al. (2015)	PCA	Real flow data	True Positive	94%
Nguyen et al. (2008)	Holt–Winters double exponential smoothing	MAWI	ROC Curve	Maximum
Li et al. (2010)	Holt–Winters double exponential smoothing	LBNL/ICSI	Detection accuracy (False positive reduction)	99.99%

5.3. *Machine learning*

Machine learning techniques have been extensively used in intrusion detection systems (Gyanchandani et al., 2012; Liao et al., 2013; Tsai et al., 2009) and also remain in focus in flow-based intrusion detection. Machine learning techniques include artificial neural network, support vector machines, k-nearest neighbor, decision trees and clustering. In the following section, we discuss the flow-based intrusion detection systems which are built on machine learning techniques.

5.3.1. *Artificial neural networks*

Artificial neural networks model human brain and use small interconnected input units called neurons. Every neuron in neural network takes part in decision making, and the result is combined. Artificial Neural networks provide a solution to the anomaly detection problem by modeling the users behavior. Beghdad (2008) discussed different neural networks used for anomaly based intrusion detection systems.

A flow-based anomaly detection system using statistical feature vectors and back-propagation neural network classifier is proposed in Song et al. (2006). The system uses 22 features of network flow records which are relevant to DoS attacks. Three scenarios of DoS attacks, resource depletion, bandwidth attack and a combination of resource depletion and bandwidth attack have been considered. The technique is evaluated on DARPA and a custom intrusion dataset. For the DARPA dataset, the detection rate is 88% and false alarm is 0.2% and there are no spoofing flooding attacks. For custom dataset, the performance is better. The detection rate is 94% and false alarm rate is 0.2%.

A flow-based intrusion detection technique using the block based neural network is proposed in Tran et al. (2012). The authors use a hardware-based detection engine for real-time processing of high volume of data. A field-programmable gate array (FPGA) is used to construct a block based neural network (BBNN). The BBNN optimization is carried out using a genetic algorithm with focus on increasing detection rate and decreasing false alarm rate. For evaluation, the authors have compared the Sperotto dataset with DARPA. The authors use DARPA because the Sperotto dataset contains fewer normal traffic samples than DARPA. The DARPA dataset is initially available in *tcpdump* format and is converted into the NetFlow format using Softflowd and Flowd tools. The authors manually label these NetFlow records by reading the original DARPA dataset. The technique has been evaluated using SVM, Radial basis function, and Naive Bayes methods. The detection rate of the BBNN is the same as of SVM, but the running time is quite good. The hardware-based BBNN took 0.005 s as compared to 8.531 s for SVM. Therefore, use of FPGA is promising for designing IDS for high-speed networks. More realistic results can be obtained by evaluating the technique on a flow-based dataset.

A two-stage neural network for intrusion detection using flow data is proposed in Abuadlla et al. (2014). The first stage detects significant changes in the traffic that could be an attack. If an attack is detected in the first stage, the flow data is forwarded to a second stage that determines the type of attack. Authors have used a multi-layer feed-forward neural network (MLFF) for attack detection in first stage and radial basis function network (RBFN) for attack classification in second stage. The first stage uses six flow features while second stage II uses 11 flow features. All first and second stage features are computed from Netflow version 5 records. The Netflow records are generated from the DARPA dataset using the softflowd tool. Three different training algorithms, Resilient back-propagation, Levenberg–Marquardt and Radial Basis Function networks, have been used for training of both neural network stages. The first stage neural network gives 94.2% detection rate and 3.4% false positive rate with Levenberg–Marquardt network. For the second stage, the best detection rate of 99.42% is obtained with Levenberg–Marquardt network while Radial basis function gives the lowest false positive rate of 2.6%. Use of multiple stages helps in achieving higher efficiency since most of the input records are discarded in first stage.

A multi-layer perceptron (MLP) with a heuristic optimization algorithm to detect anomalies in network traffic using flow data is suggested in Jadidi et al. (2013). The MLP interconnection weights are optimized using two heuristic techniques: Cuckoo and Particle Swan Optimization with Gravitational Search Algorithm (PSOGSA). Two datasets, DARPA and a subset of Sperotto dataset (Winter et al., 2011a), have been used. The comparison of results shows that multi-layer perceptron with PSOGSA optimization gives the highest accuracy of 99.55% and 0.21% false alarm rate. However, the authors concluded that the proposed approach uses centralized processing and cannot detect distributed attacks such as DDoS.

5.3.2. *Support vector machines*

Support Vector Machines (SVM) is a classification technique which maps the dataset in an n-dimensional space. SVM uses vectors in the space as classes. If the data is not linearly separable, kernel functions are used to construct high-dimensional space. SVM gives accurate results and lower false positive rate in intrusion detection (Liao et al., 2013). The SVM has also been used in flow-based intrusion detection.

A one class-SVM based model for intrusion detection using flow data is proposed in Winter et al. (2011a). The one-class SVM learns the behavior of a single class type. The authors used the malicious dataset for the training of the one-class SVM. The learning on malicious records is fast since the ratio of malicious flows is low as compared to the normal flows. The dataset used for evaluation of the one class-SVM is extracted from the Sperotto dataset (Sperotto et al., 2009). The dataset consists of 200 flow records and attributes. The accuracy results were obtained with 98% accuracy and 0% false alarm rate. However, the accuracy results can drop down to 72% if a port attribute is missing. The technique has several weaknesses as explained in the author's public errata.[1]

A technique for anomaly detection in large volumes of Netflow records using support vector machines is presented in Wagner et al. (2011). The technique takes both the contextual and the quantitative information of Netflow records into account. The approach applies kernel function on the Netflow records and forwards the computed values to a one-class SVM. The technique is evaluated on Netlow data volumes provided by an internet service provider. The authors use Flame tool to inject eight different attacks in the dataset. The results of experiment using the one-class SVM are promising and an average accuracy of 92% is obtained over all attack classes.

5.3.3. *K-nearest neighbor(k-NN)*

K-NN uses the knowledge of neighboring points to classify the input example. K-NN has been extensively applied for packet-based intrusion detection systems (Li and Guo, 2007; Lin et al., 2015; Su, 2011) and also used for flow-based intrusion detection.

A flow-based intrusion detection system using k-NN technique with fuzzy logic is proposed in Shubair et al. (2014). The work uses k-NN for selecting the best matching class. Least Mean Square technique is used for error reduction. The flow-based intrusion detection systems require additional computational intelligence as compared to packet-based systems as only the header information is available for decision. Fuzzy logic, therefore, seems to be a good choice for selecting the flow class label. Although the technique gives good results, the authors tested with only 200 training examples whereas the actual dataset contains around 14.2 M records (Sperotto et al., 2009).

A network intrusion detection technique using Optimum-path forest clustering (OPFC) is given in Costa et al. (2015). The OPFC is a k-NN graph which uses probability density function for weighting of nodes. The authors use improved nature inspired technique for optimization of OPFC. Three techniques, Bat Algorithm, Gravitational Search, Harmony search and Particle Swarm Optimization techniques are used to determine the best value of *k*. All three techniques are applied to eight packet and flow-based datasets. The results of the evaluation are obtained in the form of purity measure. The authors have compared the performance of OPFC with K-mean clustering and self-organizing maps (SOM). The evaluation on the Netflow dataset gives purity measure of 0.9577, 0.75945 and 0.2145 for OPFC, K-mean and SOM respectively. Therefore OPFC outperforms the other two clustering techniques in flow-based detection.

5.3.4. *Clustering techniques*

Clustering techniques identify novel and useful patterns in the data. These patterns can be used to group the similar instances together in different clusters. Lakhina et al. (2005) has already used clustering techniques for mining of anomalies in network flows. Recently, Casas et al. (2011) propose a network anomaly detection system using multiple unsupervised clustering techniques. The system captures packets from the network and aggregates into flows at random time slot. A change detection algorithm based on time series analysis is used to separate the malicious flows. The technique uses subspace and density-based clustering to create partitions of data in each sub-space. The algorithm also ranks the clusters in order of abnormality. All clusters above the detection threshold are considered anomalies. The detection threshold is unique for every type of attacks. The technique is evaluated on MAWI dataset and results are obtained in the form of ROC curve. The proposed technique has a larger area under ROC curve when compared with other unsupervised learning methods. This technique has the advantage that it requires no signature or training and can be instantly used to monitor the network traffic.

A distributed intrusion detection system based on Artificial Immune System and unsupervised clustering is presented in Hosseinpour et al. (2014). The authors have used DBSCAN clustering algorithm. The clustering engine labels the network traffic as malicious and non-malicious. The output of clustering engine is used to provide online and real-time training data for training of primary immune response detectors. The immune response detectors are placed around networks hosts. The technique is evaluated on KDD99 dataset and achieves F1-measure of 0.738.

A ward clustering approach to detect the dictionary attacks against SSH is presented in Satoh et al. (2015). SSH is a common way to access the remote servers over the Internet and remain a favorite attack target. The authors used two key innovations to detect an attack in SSH protocol. First, two criteria have been employed that are not available in original SSH protocol i.e. checking the existence of connection protocols and inter-arrival time of an auth-packet and the next. Second, the authors have identified transit point of each sub-protocol in SSH. The two criteria and flow of traffic during the sub-protocol transit point are inspected. The technique uses Ward's clustering method based on Euclidean distance for evaluation of flow data. The proposed technique is evaluated on a dataset generated through two observation points connected with a server to the Internet. The best results include 99.90% detection rate for unsuccessful SSH attack attempts and 92.80% detection of successful SSH attempts. The technique is promising for detection of stealthy dictionary attacks in SSH and should be used at host-level detection.

[1] https://www.cs.princeton.edu/pwinter/ntms11-errata.html

5.3.5. *Decision tree*

Decision Trees (DTs) create a tree model by creating rules based on the attribute value for every tree node. Thaseen and Kumar (2013) have discussed the application of decision trees to intrusion detection.

Zhao et al. (2013) present a flow-based solution to detect botnets. Botnets are a collection of compromised hosts controlled by a malicious user for various types of attacks and cyber crimes (Silva et al., 2013). The authors argue that flow-based approaches are better than payload inspection since most of the botnets use encrypted communication channels. The proposed method uses a decision tree algorithm with Reduced Error Pruning algorithm to construct the botnet classifier. The flow records consist of 12 attributes. The classifier is evaluated on a dataset containing traces of two botnets. The technique gives a detection rate of 98.3% and 99.9% for malicious and non-malicious categories. The technique also successfully detects novel botnets. The proposed technique is simple and efficient, but it can be evaded by inflicting small changes in the attribute values. Also a flow analysis window of 300 s is too long, and the algorithm can miss small scale malicious flows (Casas et al., 2014).

Haddadi et al. (2014) propose another solution for botnet behavior detection using genetic programming and decision trees. The proposed techniques are evaluated on a custom generated dataset of three bots. In addition publicly available dataset from Snort and NETSREr have also been used. The authors extracted two types of flow attributes from the datasets. The first set consists of common flow attributes e.g. sender/receiver IP addresses, ports. The second set uses TCF flags attributes. Evaluation results show sets of flow attributes. The first set uses the attributes similar to Netflow version 5. The second set of flow records uses TCP flags.

An efficient flow-based botnet detection using an array of supervised machine learning is presented in Stevanovic and Pedersen (2014). The technique uses 39-feature set flow records. The algorithm is evaluated on ISOP dataset, which is a combination of four publicly available malicious and non-malicious datasets. The results show that Random Forest algorithm gives overall best performance (Zhang et al., 2008).

5.4. *Advantages and disadvantages of machine learning techniques*

Table 6 gives a summary of flow-based intrusion detection system using machine learning techniques. The advantages of using machine learning technique for flow-based intrusion detection include the following:

- Intrusion detection models using machine learning techniques can adapt themselves in response to the traffic passing through.
- These techniques have high detection rate.
- Machine learning technique such as Artificial Neural Network are able to generalize the model from limited information.

The disadvantages of IDS using machine learning technique are as follows:

- It is a very difficult task to construct representative training datasets for supervised machine learning methods.
- Training process for machine learning techniques is computationally costly.
- These techniques have high false-positive alarm rate.
- Unsupervised learning techniques require background knowledge to determine the number of clusters.

5.5. *Other techniques*

Entropy is an important data mining technique. Entropy captures the important characteristic of features in traffic distribution. These features are used to detect the abnormal and malicious behavior in the network traffic. Network flow records can have a large number of attributes. Entropy technique can be used to select the attributes which play an

Table 6 – Summary of flow-based intrusion detection systems using Machine Learning techniques.

Article	Technique	Dataset	Performance Measure	Result
Song et al. (2006)	ANN	DARPA, Custom	Detection rate	DARPA – 88%, Custom – 94%
Tran et al. (2012)	ANN	DARPA	True Positive rate	99.92%
Jadidi et al. (2013)	ANN	Sperotto, ADFA	Detection accuracy	99.55% for Sperotto
Abuadlla et al. (2014)	ANN	DARPA	Detection rate	1st stage – 94.2%, 2nd stage 99.42%
Winter et al. (2011a)	SVM	Sperotto	Class prediction	98% for Malicious traffic
Wagner et al. (2011)	SVM	Custom	Detection accuracy	92%
Shubair et al. (2014)	kNN, Fuzzy Logic	Sperotto, custom	Correctness rate	99.34% for Sperotto
Costa et al. (2015)	k-mean clustering	Sperotto	Purity measure	0.9577 for Sperotto
Casas et al. (2011)	DBSCAN clustering	MAWI	Area under ROC curve	Maximum
Hosseinpour et al. (2014)	DBSCAN, AIS	KDD99	F1-score	0.738
Satoh et al. (2015)	Clustering	Custom	Detection rate	99.99% and 92.4% for successful and unsuccessful attempts
Zhao et al. (2013)	Decision Tree	Custom	Detection rate	98.3% and 99.9% for malicious and non-malicious
Haddadi et al. (2014)	Decision Tree (C4.5)	Custom	Detection rate	97% and 99% for two datasets
Stevanovic and Pedersen (2014)	Random Forest (ensemble of decision trees)	ISOT	F1-measure	0.9596

important role in intrusion detection decision. Wagner and Plattner (2005) used entropy for detection of worms and anomaly in IP networks using flow records. An algorithm for detection of abrupt changes in network entropy time series is proposed in Winter et al. (2011b). The technique is based on the idea that any network attack will inflict significant changes in flow attribute values which can be detected in entropy time series. The authors have proposed an abrupt change detection algorithm which uses a dynamic anomaly score to detect the attack. The algorithm is evaluated on a dataset obtained from an ISP's server. The dataset contains five days' traffic of unidirectional network flows. Two synthetic anomalies HTTP DoS attack and horizontal network scan are manually injected at a specific time. The technique successfully detects the change in the traffic at the given time. As also indicated by authors, the technique can be evaded with small scaled DDoS attacks. However, the technique does not need training data and can be straightforwardly used to monitor the network activity.

A technique to detect large-scale anomalies in the network traffic is proposed in François et al. (2012). The technique stores the profiles of normal traffic. All incoming flow records are first aggregated and subsequently compared with profiles of normal traffic. The deviation of incoming flow records with normal traffic profiles is measured using the Shannon Entropy formula. For evaluation, a custom dataset is obtained from a commercial service provider and attacks are manually inserted using the Flame tool. A drawback of the technique is the requirement of normal network profiles which are quite difficult to generate in multi-service real-world networks.

An entropy-based Internet traffic anomaly detection system is discussed in Bereziński et al. (2014). The authors have used Shannon, Renyi and variants of Tsallis entropies combined with a set of feature distributions. The entropy techniques are employed in a flow-based framework. A variant of the Sperotto dataset is used for evaluation of the proposed methodology. In training mode, a profile is created for normal traffic using time-specific entropy values. Any value exceeding the upper limit of entropy limit is considered abnormal. The results show that Tsallis and Renyi's entropies performed best while Shannon entropy and counter-based methods performed poorly. This technique can generate false positives if there is a benign change in network traffic such as congestion.

In Qin et al. (2015), authors describe an entropy-based approach for DDoS attack detection. The technique calculates the entropy values of the selected flow features. The features are used by the clustering algorithm to construct a normal flow profile. This normal flow profile is used to detect DOS attack in the incoming traffic. The technique is evaluated on the DARPA dataset and results are obtained in the form of DF rate. DF rate is defined as the ratio of detection rate and false positive rate. The technique obtains a best DF rate of 7.

A number of techniques using flow metric threshold values, flow signatures and semantic link networks (SLN) have also been used for flow-based intrusion detection. Dubendorfer et al. (2005) used threshold values to detect an intrusion in the network flows. SSHCure, a flow-based system for detection of SSH attacks, also uses flow metrics threshold (Hellemons et al. 2012). The SSHCure employs an efficient algorithm for the real-time detection of ongoing attacks and allows identification of compromised attack targets. Three phases of the SSH attack scanning phase, brute force phase, and die-off phase are identified. During scanning phase, the attacker scans an IP to find the SSH daemon. In brute-force phase, the attacker tries to log in the SSH server. If the login is successful, The die-off phase shows the attack traffic between an attacker and target host. SSHCure uses two flow metrics to detect an attack in all three phases. The first metric is an upper bound of two packets per flow. The second metric defined a minimum number of flow records for every attack. Every attack phase uses a different threshold value for the two flow metrics. The detection performance of the system is validated with empirical traffic data. The real-time implementation of SSHcure shows that algorithm has various shortcomings that ultimately cause compromises to remain undetected or false alarms to be raised (Hofstede et al., 2014b).

Vizváry and Vykopal (2013) present a flow-based technique for remote desktop protocol (RDP) brute force attack detection using flow signatures. Use of flow patterns for attack detection was proposed by Kim et al. (2004). In Vizváry and Vykopal (2013), the authors analyze the flow traffic signature of RDP clients, brute force tools and successful authentication events. The flow signatures are evaluated on campus network of Masaryk University for two months. The authors used RdpMonitor, a publicly available NfSen plug-in with derived NetFlow signature, to automate the detection. The plug-in successfully detects the malicious traffic and reports that approximately 40% of all RDP related traffic in the campus network is malicious.

A novel approach for detection of cyber attacks using Semantic Link Networks (SLNs) is proposed in AlEroud and Karabatis (2014). Semantic Link Networks mine time, location, and other contextual information from the flow data. The contextual information is used by the semantic links among the alerts for suspicious flows on probabilistic semantic networks (SLNs). These semantic links help in retrieving relevant suspicious activities that can be a part of multi-step attacks. The technique is evaluated on a mix of Sperotto and ICSX dataset and achieved F1 score of 0.97. The semantic link technique is also compared with other flow-based intrusion detection systems, and results show that it outperforms other methods.

Hofstede et al. (2014b) enhance the SSHCure and propose two phase detection algorithm for SSH compromise detection. The first phase is brute-force phase. The brute-phase is detected by checking for the same number of packets per flow. Two unsuccessful connection attempts from the attacker will have the same number of packets per flow. The second phase is compromise phase. The technique transforms the compromise phase in six attack scenarios. If the flow of SSH traffic matches a particular attack scenario, an attack is detected, and the connection is closed down. The technique is validated on the SSHCure dataset and obtains an accuracy of 83% and 99% for the two segments in the SSHCure dataset.

A behavioral botnet detection method using Markov Chains is presented in García et al. (2014). The technique analyzes the Command & Control (C&C) channel of botnet communication using flow features and a 4-tuple structure consisting of Source IP, Destination IP, Destination Port and Protocol. The authors used a Markov Chain to model the different states in

Table 7 – Flow-based intrusion detection systems using Entropy, Threshold values, Flow Signatures, Semantic Link Networks (SLN) and Markov Chains.

Article	Technique	Dataset	Performance Measure	Result
François et al. (2012)	Entropy	Custom	Graph Generation	Anomaly detected
Qin et al. (2015)	Entropy	DARPA	DF Rate	7
Bereziński et al. (2014)	Entropy	Sperotto	Anomaly type detection	100%
Hellemons et al. (2012)	Threshold values	Custom	True Positive rate	99%
Vizváry and Vykopal (2013)	Flow signature	Custom	Brute force Detection	Yes
AlEroud and Karabatis (2014)	Semantic Links	Sperotto	F1-score	0.97
Hofstede et al. (2014b)	Threshold value	SSHCure	Detection accuracy	83% and 99%
García et al. (2014)	Markov Chain	CTU-13	F1-measure	92%
Gogoi et al. (2014)	CatSub+, K-point, GBBK	TUIDS	Recall	0.99, 0.98 and 0.98 for DoS, Probe and Normal
Wijesinghe et al. (2015)	Flow signature	Custom	Botnet detection	Detects all botnets

the C&C channel. The proposed method is trained and evaluated using the CTU-13 dataset. The technique gives an F1-measure of 92% and false positive rate of 0.05%. The approach is capable of detecting variety of botnets.

In Gogoi et al. (2014), the authors propose a multi-level hybrid intrusion detection method which combines supervised, unsupervised and outlier-based methods for intrusion detection. Three algorithms, CatSub+, K-point and GBBK have been used for supervised, unsupervised and outlier detection respectively. The hybrid framework is evaluated on a number of datasets including a flow-based dataset, TUIDS. Although results for flow-based evaluation are good, the technique has a weakness. The selection of supervised, unsupervised or outlier-based classifier at a particular level for a given dataset is based on the classification accuracy of the individual classifier for a given dataset. This adjustment is difficult to implement in real-time scenarios and similar performance results might not be achieved.

A botnet detection method using flow templates is presented in Wijesinghe et al. (2015). The technique creates flow-based profiles of different botnets and compares them with network traffic for botnet detection. The experimental environment consists of virtual machines infected with botnets. Results show that techniques detect all family of botnets. However, this approach requires prior knowledge of botnet operation and may not be able to detect unknown botnets. Table 7 gives summary of these techniques.

6. Commercial applications of flow-based intrusion detection

The flow-based intrusion detection has also gained attention of commercial vendors. The Lancope StealthWatch system,[2] available through Cisco, is an enterprise network monitoring and security intelligence solution completely based on NetFlow. The StealthWatch system performs collection, aggregation, and analysis of NetFlow and other contextual data. The system is able to detect malicious behaviors linked to APTs, insider threats, DDoS and malware. Cisco's Next-generation Intrusion Prevention System[3] with FireSIGHT Management Center[4] also have Netflow capabilities.

The Scrutinizer System[5] developed by Plixer Inc. is a flow-based incident response and behavior analysis product. Scrutinizer performs the collection, threat detection, and reporting of network activities using a number of flow technologies. It also offers real-time situational awareness and historical behaviors of the network. The system uses fixed algorithms to detect DoS attacks, network scans and other abnormal network behavior.

Flowmon Anomaly Detection System (ADS)[6] is another flow-based intrusion detection system supporting NetFlow, IPFIX and NetStream protocols. Flowmon ADS uses intelligent behavioral analysis algorithms to identify threats, attacks, incidents and configuration issues. It also provides protection against DoS attacks.

The IBM QRadar Security Intelligence Platform[7] is a security information and event management (SIEM) system with anomaly detection, incident forensics and vulnerability management. The QRadar also includes IP flow-based analysis support.

Juniper Networks JSA Series Secure Analytics[8] give a complete set of network monitoring and surveillance tools for enterprise level. The JSA series products have support for NetFlow, J-Flow, sFlow, and IPFIX. The products also include Network Behavior Anomaly Detection (NBAD) to detect rough servers and APTs.

In open-source world, the Bro[9] is a comprehensive platform for general network traffic analysis. It also has the capability of intrusion detection. In addition to other network

[2] https://www.lancope.com/products-services-lancope

[3] http://www.cisco.com/c/en/us/products/security/ngips/index.html

[4] http://www.cisco.com/c/en/us/products/security/firesight-management-center/index.html

[5] https://www.plixer.com/products/scrutinizer/

[6] https://www.flowmon.com/en/products/flowmon/anomaly-detection-system

[7] http://www-03.ibm.com/software/products/en/qradar

[8] http://www.juniper.net/us/en/products-services/security/secure-analytics/

[9] https://www.bro.org/

tapping tools, Bro also uses Netflow for network analysis and threat detection.

7. Observations

In the previous section, we have discussed various classes of methods being used for designing flow-based intrusion detection system. Our observations about existing flow-based intrusion detection techniques are as follows:

- A number of existing studies on flow-based detection use statistical methods. There is a need to exploit the true potential of machine learning techniques for flow-based detection. Techniques such as Bayesian Network, Ensemble Learning, Evolutionary computing, and Sequential Pattern Mining can be considered.
- A number of techniques examined in this paper only address specific attack types. Muraleedharan and Parmar's (2010) work only detects TCP scan. Hellemons et al. (2012) and Satoh et al. (2015) give solution for protection against SSH attacks. Solutions for detection of DOS attacks using IP flow records are given in Hofstede et al. (2013) and Zhang et al. (2012). Such techniques give better results for the particular attack type or scenario but have not been evaluated against other attacks. It is difficult to integrate such techniques into a comprehensive flow-based intrusion detection framework for overall protection of an enterprise network.
- Some flow-based intrusion detection use packet-based datasets to generate the flow records for evaluation of flow-based techniques (Abuadlla et al., 2014; Qin et al., 2015; Salem et al., 2011; Song et al., 2006; Tran et al., 2012). However, evaluation on such datasets is not guaranteed to give similar results in real-world flow-based detection. Some techniques use custom datasets which make it difficult to compare the results with other techniques (Hellemons et al., 2012; Wagner et al., 2011).
- Several studies do not use a representative dataset to obtain validation results. Shubair et al. (2014) and Winter et al. (2011a) use a dataset of 200 flow records for validation whereas original dataset has nearly 1.4 million records. The techniques evaluated on such dataset will not perform better in real-world and will give many false-positives. Guo et al. (2013) present a solution for obtaining representative instances from large datasets. The approach is useful when input size is sufficiently large and can affect the space complexity of the algorithm.
- There are a number of evaluation measures used in existing work to obtain the experimental results. These include Precision, Recall, F1-Score, True positive rate, DF rate, Correctness rate, Purity measure, ROC curve and Area under ROC curve, etc. They are basically comparable to some extent with standard evaluation measures, which can be derived from each other. However, a number of techniques report the results as a normal or malicious flag which does not provide a quantitative result for intrusion detection.

8. Open issues and challenges

The comparison of the packet and flow-based techniques shows that flow-based techniques are a better choice for protection of high-speed networks (Hellemons et al., 2012; Golling et al, 2014). However, flow-based intrusion detection is not matured enough to replace the traditional packet-based techniques (Sperotto et al., 2010). Sperotto et al. (2010) and Golling et al. (2014) propose that flow-based detection should be applied in combination with packet-based detection. Flow-based intrusion detection should be implemented at an initial layer of intrusion detection while detailed in-depth intrusion detection should be performed at a second layer using packet-based inspection. However, such techniques will still have the drawbacks of packet-based detection. Another problem in this hybrid arrangement is storage of packet payload till the time flow-based detection is completed. We identify following challenges for further research in flow-based intrusion detection:

- Intrusion detection datasets are a valuable tool for evaluation of proposed techniques. As discussed in Section 4, very few public flow-based intrusion datasets exist. There is an urgent need for the development of public flow-based datasets having a variety of attack traffic for evaluation and comparison of different flow-based intrusion detection techniques.
- IPFIX/Netflow specify around 280 attributes for network flow record. Researchers use a different selection of flow attributes to evaluate the network traffic. Tran et al. (2012) and Jadidi et al. (2013) use a 4 and 7-tuple flow record respectively. Zhao et al. (2013) employ a 12-tuple flow record. However, no literature exists which establish the relationship between flow attributes and attack types. Increasing the number of flow attributes will also increase the computation cost whereas using few flow attributes can miss important network information. It is therefore important to explore the relation of flow attributes with intrusion detection performance.
- Flow-based techniques do not have access to the packet payload. Therefore flow-based intrusion detection systems cannot detect attacks which are embedded in packet payload and do not alter the flow of traffic such as SQL injection and cross-side scripting (Vykopal et al., 2013).
- Some techniques use flow records consisting of four or five flow attributes (Jadidi et al., 2013; Winter et al., 2011a) which are not sufficient to analyze the flow of traffic. A solution to this problem is computation of additional flow metrics using basic flow attributes (Zhang et al., 2012).
- An important issue in flow-based intrusion detection is the use of flow sampling techniques. The Packet Sampling (PSAMP) Protocol (Claise, 2009) specifies different sampling techniques for flow export process. Sampling techniques which are a true representative of the original population can help in archiving better results. Therefore it is important to investigate the effect of sampling techniques with the accuracy and efficiency of flow-based intrusion detection.
- The value of flow export interval critically affects the performance of flow-based intrusion detection system (Vykopal

et al., 2013). If flow export interval is kept longer, a short-timed attack can occur, and intrusion detection system may not be able to detect it. On the other hand, a shorter flow-interval can overload the system and will affect the overall efficiency. This relation of flow export interval with detection performance should be analyzed in detail (Hofstede et al., 2013).

9. Future of flow-based intrusion detention

The increased use of end-to-end encryption in network applications, e.g. websites, mobile apps, and emails have left limited space for the application of payload-based intrusion detection systems. The payload-based intrusion detection is also not feasible for large-scale backbone and service provider networks. Sperotto et al. (2010) envisaged the use of flow-based detection for spam detection, botnet detection, and distributed attack detection. Our review also shows that the research in flow-based detection is gaining attention as an alternative to traditional packet-based intrusion detection.

The existing research in flow-based intrusion detection is focused on monitoring changes in network traffic using statistical or machine learning techniques. The standalone flow-based inspection of network traffic is now evolving into Network Behavior Analysis (NBA) (Shackleford, 2016). Network Behavior Analysis (NBA) approaches collect network flow records from a variety of devices including, routers, switches, servers from across the network. The NBA does not rely on individual traffic flow metrics and also makes use of network performance and surveillance data. NBA uses intelligent machine learning techniques to build a normal profile of the network. The normal profile acts as a baseline against detection of intrusions. Use of NBA with machine learning techniques has a number of advantages over other intrusion detection techniques (Liao et al., 2013). The integrated use of flow-based inspection, network performance and security metrics provide comprehensive protection for enterprise level networks against intrusions. Latest commercial intrusion detection applications like Stealthwatch from Cisco and Scrutinizer from Plixer are also built on Network Behavior Analysis.

10. Summary

In this paper, we have performed an up-to-date review of available flow-based intrusion detection systems. We proposed a taxonomy of flow-based intrusion detection systems on the basis of the technique used for detection of malicious flows. We discussed the architecture, algorithms, and datasets of flow-based techniques in every class of method. Our discussion reveals that an important aspect in the evaluation of flow-based intrusion detection systems is the use of native flow-based dataset. We also provide a brief summary of available flow-based datasets. Other significant contributions are the discussion on commercial flow-based IDS products and future of flow-based intrusion detection. In the end, we gave observations about the existing systems and also future research direction.

REFERENCES

Abuadlla Y, Kvascev G, Gajin S, Jovanovic Z. Flow-based anomaly intrusion detection system using two neural network stages. Comput Sci Inf Syst 2014;11(2):601–22.

AbuHmed T, Mohaisen A, Nyang D. 2008. A survey on deep packet inspection for intrusion detection systems. arXiv preprint arXiv:0803.0037.

AlEroud A, Karabatis G. 2014. Context infusion in semantic link networks to detect cyber-attacks: A flow-based detection approach. In: Semantic Computing (ICSC), 2014 IEEE International Conference on. IEEE, pp. 175–182.

Anantvalee T, Wu J. A survey on intrusion detection in mobile ad hoc networks. In: Wireless network security. Springer; 2007. p. 159–80.

Beghdad R. Critical study of neural networks in detecting intrusions. Comput Secur 2008;27(5):168–75.

Bereziński P, Pawelec J, Małowidzki M, Piotrowski R. 2014. Entropy-based internet traffic anomaly detection: A case study. In: Proceedings of the Ninth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX. June 30–July 4, 2014, Brunów, Poland. Springer, pp. 47–58.

Bhuyan MH, Bhattacharyya DK, Kalita JK. Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 2014;16(1):303–36.

Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 2015;18(2):1153–76.

Butun I, Morgera SD, Sankar R. A survey of intrusion detection systems in wireless sensor networks. IEEE Commun Surv Tutor 2014;16(1):266–82.

Casas P, Mazel J, Owezarski P. UNADA: unsupervised network anomaly detection using sub-space outliers ranking. In: Networking 2011. Springer; 2011. p. 40–51.

Casas P, Mazel J, Owezarski P. 2014. Coping with 0-day attacks through unsupervised network intrusion detection. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International. IEEE, pp. 24–29.

Claise B. 2009. Packet sampling (psamp) protocol specifications. IETF.

Copeland JA III, 2007. Flow-based detection of network intrusions. US Patent 7,185,368.

Costa KA, Pereira LA, Nakamura RY, Pereira CR, Papa JP, Falcão AX. A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks. Inf Sci (Ny) 2015;294:95–108.

Drašar M, Vizváry M, Vykopal J. Similarity as a central approach to flow-based anomaly detection. Int J Netw Manag 2014;24(4):318–36.

Dubendorfer T, Wagner A, Plattner B. 2005. A framework for real-time worm attack detection and backbone monitoring. In: First IEEE International Workshop on Critical Infrastructure Protection (IWCIP'05). IEEE, pp. 10–pp.

Ellens W, Żuraniewski P, Sperotto A, Schotanus H, Mandjes M, Meeuwissen E. Flow-based detection of DNS tunnels. In: Flow-based detection of DNS tunnels. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 124–35.

Fernandes G, Rodrigues JJ, Proença ML. Autonomous profile-based anomaly detection system using principal component analysis and flow analysis. Appl Soft Comput 2015;34:513–25. doi:10.1016/j.asoc.2015.05.019.

Fontugne R, Borgnat P, Abry P, Fukuda K. 2010. MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In: Proceedings of the 6th International COnference. ACM, p. 8.

François J, Wagner C, State R, Engel T. 2012. SAFEM: Scalable analysis of flows with entropic measures and SVM. In:

Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, pp. 510–513.

Gao Y, Li Z, Chen Y. 2006. A dos resilient flow-level intrusion detection approach for high-speed networks. In: 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06). IEEE, pp. 39–39.

García S, Uhlíř V, Rehak M. 2014. Identifying and modeling botnet C&C behaviors. In: Proceedings of the 1st International Workshop on Agents and CyberSecurity. ACM, p. 1.

Garcia S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. Comput Secur 2014;45:100–23.

Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G, Vázquez E. Anomaly-based network intrusion detection: techniques, systems and challenges. Comput Secur 2009;28(1):18–28.

Gogoi P, Bhuyan MH, Bhattacharyya D, Kalita JK. Packet and flow based network intrusion dataset. In: Contemporary computing. Springer; 2012. p. 322–34.

Gogoi P, Bhattacharyya D, Borah B, Kalita JK. MLH-IDS: a multi-level hybrid intrusion detection method. Comput J 2014;57(4):602–23.

Golling M, Hofstede R, Koch, R., 2014. Towards multi-layered intrusion detection in high-speed networks. In: Cyber Conflict (CyCon 2014), 2014 6th International Conference On. IEEE, pp. 191–206.

Guo C, Zhou Y-J, Ping Y, Luo S-S, Lai Y-P, Zhang Z-K. Efficient intrusion detection using representative instances. Comput Secur 2013;39:255–67.

Gyanchandani M, Rana J, Yadav R. Taxonomy of anomaly based intrusion detection system: a review. Neural Netw 2012; 2(43):1–14.

Haddadi F, Runkel D, Zincir-Heywood AN, Heywood MI. 2014. On botnet behaviour analysis using GP and C4.5. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 1253–1260.

Hellemons L, Hendriks L, Hofstede R, Sperotto A, Sadre R, Pras A. SSHCure: a flow-based SSH intrusion detection system. In: Dependable networks and services. Springer; 2012. p. 86–97.

Hofstede R, Bartos V, Sperotto A, Pras A. 2013. Towards real-time intrusion detection for NetFlow and IPFIX. In: Network and Service Management (CNSM), 2013 9th International Conference on. IEEE, pp. 227–234.

Hofstede R, Celeda P, Trammell B, Drago I, Sadre R, Sperotto A, et al. Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. IEEE Commun Surv Tutor 2014a;16(4):2037–64.

Hofstede R, Hendriks L, Sperotto A, Pras A. SSH compromise detection using NetFlow/IPFIX. ACM SIGCOMM Comput Commun Rev 2014b;44(5):20–6.

Hosseinpour F, Amoli PV, Farahnakian F, Plosila J, Hämäläinen T. Artificial immune system based intrusion detection: innate immunity using an unsupervised learning approach. Int J Digit Content Tech Appl 2014;8(5):1.

Jadidi Z, Muthukkumarasamy V, Sithirasenan E. 2013. Metaheuristic algorithms based flow anomaly detector. In: Communications (APCC), 2013 19th Asia-Pacific Conference on. IEEE, pp. 717–722.

Kanda Y, Fontugne R, Fukuda K, Sugawara T. Admire: anomaly detection method using entropy-based PCA with three-step sketches. Comput Commun 2013;36(5):575–88.

Kim M-S, Kong H-J, Hong S-C, Chung S-H, Hong JW. 2004. A flow-based method for abnormal network traffic detection. In: Network operations and management symposium, 2004. NOMS 2004. IEEE/IFIP. Vol. 1. IEEE, pp. 599–612.

Koch R. 2011. Towards next-generation intrusion detection. In: Cyber Conflict (ICCC), 2011 3rd International Conference on. IEEE, pp. 1–18.

Lakhina A, Crovella M, Diot C. Mining anomalies using traffic feature distributions. SIGCOMM Comput Commun Rev 2005;35(4):217–28. http://doi.acm.org/10.1145/1090191.1080118.

Li B, Springer J, Bebis G, Gunes MH. A survey of network flow applications. J Netw Comput Appl 2013;36(2):567–81.

Li Y, Guo L. An active learning based TCM-KNN algorithm for supervised network intrusion detection. Comput Secur 2007;26(7):459–67.

Li Z, Gao Y, Chen Y. Towards a high-speed router-based anomaly/intrusion detection system. Poster Presentation; 2005. Available from: http://conferences.sigcomm.org/sigcomm/2005/poster-121.pdf.

Li Z, Gao Y, Chen Y. HiFIND: a high-speed flow-level intrusion detection approach with DoS resiliency. Comput Netw 2010;54(8):1282–99. [Accessed 1 October 2016].

Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y. Intrusion detection system: a comprehensive review. J Netw Comput Appl 2013;36(1):16–24.

Lin W-C, Ke S-W, Tsai C-F. CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl Based Syst 2015;78:13–21.

Marek Malowidzki PB, Mazur M. 2015. Network intrusion detection: Half a kingdom for a good dataset. In: Proceedings of NATO STO SAS-139 Workshop.

Muraleedharan N, Parmar A. ADRISYA: a flow based anomaly detection system for slow and fast scan. Int J Netw Secur Appl 2010;234–45.

Nadeem A, Howarth MP. A survey of MANET intrusion detection & prevention approaches for network layer attacks. IEEE Commun Surv Tutor 2013;15(4):2027–45.

Nguyen HA, Nguyen TV, Kim DI, Choi D. 2008. Network traffic anomalies detection and identification with flow monitoring. In: Wireless and Optical Communications Networks, 2008. WOCN'08. 5th IFIP International Conference on. IEEE, pp. 1–5.

Patel A, Qassim Q, Wills C. A survey of intrusion detection and prevention systems. Inf Manag Comput Secur 2010;18(4):277–90.

Patel A, Taghavi M, Bakhtiyari K, JúNior JC. An intrusion detection and prevention system in cloud computing: a systematic review. J Netw Comput Appl 2013;36(1):25–41.

Qayyum A, Islam M, Jamil M. 2005. Taxonomy of statistical based anomaly detection techniques for intrusion detection. In: Emerging Technologies, 2005. Proceedings of the IEEE Symposium on. IEEE, pp. 270–276.

Qin X, Xu T, Wang C. 2015. DDoS attack detection using flow entropy and clustering technique. In: 2015 11th International Conference on Computational Intelligence and Security (CIS). IEEE, pp. 412–415.

Sadre R, Sperotto A, Pras A. 2012. The effects of DDoS attacks on flow monitoring applications. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, pp. 269–277.

Salem O, Makke A, Tajer J, Mehaoua A. 2011. Flooding attacks detection in traffic of backbone networks. In: Local Computer Networks (LCN), 2011 IEEE 36th Conference on. IEEE, pp. 441–449.

Satoh A, Nakamura Y, Ikenaga T. A flow-based detection method for stealthy dictionary attacks against secure shell. J Inf Secur Appl 2015;21:31–41.

Shackleford D. Real-time adaptive security; 2016. Available from: https://www.sans.org/reading-room/whitepapers/analyst/real-time-adaptive-security-34740.

Shubair A, Ramadass S, Altyeb AA. kENFIS: kNN-based evolving neuro-fuzzy inference system for computer worms detection. J Intell Fuzzy Syst 2014;26(4):1893–908.

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

Song S, Ling L, Manikopoulo C. 2006. Flow-based statistical aggregation schemes for network anomaly detection. In: 2006 IEEE International Conference on Networking, Sensing and Control. IEEE, pp. 786–791.

Sperotto A, Pras A. 2011. Flow-based intrusion detection. In: Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on. IEEE, pp. 958–963.

Sperotto A, Sadre R, Pras A. Anomaly characterization in flow-based traffic time series. In: International workshop on IP operations and management. Springer; 2008. p. 15–27.

Sperotto A, Sadre R, Van Vliet F, Pras A. A labeled data set for flow-based intrusion detection. In: IP operations and management. Springer; 2009. p. 39–50.

Sperotto A, Schaffrath G, Sadre R, Morariu C, Pras A, Stiller B. An overview of IP flow-based intrusion detection. IEEE Commun Surv Tutor 2010;12(3):343–56.

Stevanovic M, Pedersen JM. 2014. An efficient flow-based botnet detection using supervised machine learning. In: Computing, Networking and Communications (ICNC), 2014 International Conference on. IEEE, pp. 797–801.

Su M-Y. Real-time anomaly detection systems for denial-of-service attacks by weighted k-nearest-neighbor classifiers. Expert Syst Appl 2011;38(4):3492–8.

Szabó G, Orincsay D, Malomsoky S, Szabó I. On the validation of traffic classification algorithms. In: International conference on passive and active network measurement. Springer; 2008. p. 72–81.

Thaseen S, Kumar CA. 2013. An analysis of supervised tree based classifiers for intrusion detection system. In: Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on. IEEE, pp. 294–299.

Trammell B, Claise B. 2013. Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. Tech. rep.

Tran QA, Jiang F, Hu J. 2012. A real-time NetFlow-based intrusion detection system with improved BBNN and high-frequency field programmable gate arrays. In: Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on. IEEE, pp. 201–208.

Tsai C-F, Hsu Y-F, Lin C-Y, Lin W-Y. Intrusion detection by machine learning: a review. Expert Syst Appl 2009;36(10):11994–2000.

van der Steeg D, Hofstede R, Sperotto A, Pras A. 2015. Real-time DDoS attack detection for Cisco IOS using NetFlow. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, pp. 972–977.

Vasilomanolakis E, Karuppayah S, Mühlhäuser M, Fischer M. Taxonomy and survey of collaborative intrusion detection. ACM Comput Surv 2015;47(4):55.

Vizváry M, Vykopal J, 2013. Flow-based detection of RDP brute-force attacks. In: Proceedings of 7th International Conference on Security and Protection of Information (SPI 2013).

Vykopal J, Drašar M, Winter P. 2013. Flow-based brute-force attack detection. Fraunhofer Research Institution AISEC, Garching near Muenchen, 41–51.

Wagner A, Plattner B. 2005. Entropy based worm and anomaly detection in fast IP networks. In: 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05). IEEE, pp. 172–177.

Wagner C, François J, Engel T. Machine learning approach for IP-flow record anomaly detection. In: International Conference on Research in Networking 2011. Berlin Heidelberg: Springer; 2011. p. 28–39.

Wijesinghe U, Tupakula U, Varadharajan V. 2015. An enhanced model for network flow based botnet detection. In: Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015). Vol. 27. p. 30.

Winter P, Hermann E, Zeilinger M. 2011a. Inductive intrusion detection in flow-based network data using one-class support vector machines. In: New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on. IEEE, pp. 1–5.

Winter P, Lampesberger H, Zeilinger M, Hermann E. On detecting abrupt changes in network entropy time series. In: Communications and multimedia security. Springer; 2011b. p. 194–205.

Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: a review. Appl Soft Comput 2010;10(1):1–35.

Zhang C, Cai Z, Chen W, Luo X, Yin J. Flow level detection and filtering of low-rate DDoS. Comput Netw 2012;56(15):3417–31.

Zhang J, Zulkernine M, Haque A. Random-forests-based network intrusion detection systems. IEEE Trans Syst Man Cybern Part C 2008;38(5):649–59.

Zhang W, Yang Q, Geng Y. 2009. A survey of anomaly detection methods in networks. In: Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on. IEEE, pp. 1–3.

Zhao D, Traore I, Sayed B, Lu W, Saad S, Ghorbani A, et al. Botnet detection based on traffic behavior analysis and flow intervals. Comput Secur 2013;39:2–16.

Muhammad Fahad Umer is a PhD researcher at Department of Computer Science & Software Engineering, International, Islamic University, Islamabad. Since 2004 he has been working in the field of Network & Cyber security. His research focuses on using machine learning techniques for building intelligent intrusion detection systems.

Dr. Muhammad Sher is a Professor at Department of Computer Science & Software Engineering, International Islamic University. He has more than 120 scientific publications. He received his PhD in Computer Science from TU Berlin, Germany. His research interests include Next Generation Networks, IP Multimedia Sub-systems and Network Security.

Yaxin Bi is a Reader in School of Computing and Mathematics. His principle research interests include multiple supervised and unsupervised machine learning-based classification systems and ensemble methods in conjunction with the Dempter-Shafer (DS) theory of evidence; data analytics and decision making with uncertainty methods for satellite data exploitation with an emphasis on anomaly/change detection; sentiment analytics for opinion mining and cyberbullying detection; and sensor fusion for activity/event recognition in smart environments. Yaxin has more than 130 peer-reviewed publications in international journals and conferences. He has served for a number of international conferences as general chair, program co-chair and program committee member, he also is an associate editor for International Journal of Artificial Intelligence Review as well as a steering committee member for the series of International Conference on Knowledge Science, Engineering and management (KSEM).