Intelligent Sampling Using an Optimized Neural Network

Zahra Jadidi, Vallipuram Muthukkumarasamy, Elankayer Sithirasenan, and Kalvinder Singh
School of Information and Communication Technology, Gold Coast Campus, Griffith University, QLD 4222, Australia
Email: zahra.jadidi@griffithuni.edu.au, {v.muthu, e.sithirasenan}@griffith.edu.au, kalsingh@au.ibm.com

***Abstract*—Modern Internet has enabled wider usage, resulting in increased network traffic. Due to the high volume of data packets in networking, sampling techniques are widely used in flow-based network management software to manage traffic load. However, sampling processes reduce the likelihood of anomaly detection. Many studies have been carried out at improving the accuracy of anomaly detection. However, only a few studies have considered it with sampled flow traffic. In our study, we investigate the use of an artificial neural network (ANN)-based classifier to improve the accuracy of flow-based anomaly detection in sampled traffic. A feedback from the ANN-based anomaly detector determines the type of the flow sampling method that should be used. Our proposed technique handles malicious flows and benign flows with different sampling methods. To evaluate the proposed sampling technique, a number of flow-based datasets are generated. Our experiments confirm that the proposed technique improves the percentage of the sampled malicious flows by about 7% and it can preserve the majority of traffic information.**

***Index Terms*—Flow-Based Anomaly Detection; Flow Sampling; Artificial Neural Networks; Metaheuristic Algorithms; Monitoring**

I. Introduction

Flow analysis, based on packet header information, is increasingly used for monitoring and traffic management. In the recent past, a number of studies have been carried out using flow traffic to detect anomalies in high-speed networks. Different flow-based network management techniques typically employ sampling techniques to manage the high volume of network traffic. However, sampling processes degrade the accuracy of detecting anomalies. This study deals with difficulties in detecting anomalies in sampled traffic [1].

A flow is a group of unidirectional network packets that pass through a monitoring point during a specific time interval. In each flow, packets are transmitted between a specific source and destination with some common characteristics such as source and destination ports, and protocol [2]. Compared with payload-based methods, flow-based anomaly detection methods are more efficient computationally in terms of memory and time [3]. Also, they decrease the privacy concern because they only analyze packet headers. However, as a flow does not contain any payload, a flow-based method is not useful in detecting attacks related to packet payloads. It has been shown that volume anomalies such as denial of service (DoS) attacks, distributed DoS (DDoS) attacks, worms, port scans and botnets can be detected by a flow-based anomaly detection system [4]. In order to detect anomalies in flow traffic, different machine learning methods have been proposed, for example, self-organizing map (SOM) [2], support vector machine (SVM) [2], hidden Markov model [5], modified random-mutation hill-climbing and C4.5 (MRMHCC4.5) algorithm [6], frequent pattern mining algorithm [7], data mining and visualization [8], statistical techniques [9], chi-square technique [10], semi-supervised methods [1], and artificial neural networks (ANNs) [12, 13], e.g. multilayer perceptron (MLP) [14]. An MLP neural network optimized with a metaheuristic algorithm is used in this study to classify flow traffic. Metaheuristic algorithms are extensively used for the optimization of structure and weights of ANNs. Genetic algorithm (GA) [15], bat algorithm [16], immune algorithm [17], particle swarm optimization (PSO) algorithm [18, 19], and gravitational search algorithm (GSA) [20] are a number of metaheuristic algorithms.

Flow-based anomaly detection methods for high-speed networks mostly use sampled traffic. Two sampling methods, packet sampling and flow sampling, are widely investigated [21, 22]. Packet sampling is performed at routers before flows are generated but flow sampling is applied to flows at NetFlow collectors [23]. NetFlow is a Cisco proprietary which provides network flows and it is enabled on router devices. Implementation of a packet sampling method is easy, but it causes a serious bias in flow statistics. Compared with packet sampling, flow sampling is more efficient in terms of preserving the characteristics of network traffic [24]. However, it is shown that flow sampling methods also negatively affect the accuracy of anomaly detection [1].

Despite the wide use of sampled traffic in networks, there is not sufficient research on the impact of sampling on flow-based anomaly detection. This study aims to fill this gap by making the following contributions:

PSOGSA algorithm - we optimize an MLP neural network to improve its accuracy and reliability in the classification of flow traffic in high-speed networks. In this regard, a metaheuristic algorithm called PSOGSA [25] is used to optimize the interconnection weights of MLP [12]. This optimized classifier is trained with flow-based datasets to distinguish between benign and malicious flow traffic.

doi:10.4304/jnw.11.1.16-27

Flow sampling technique - we propose this technique to reduce the negative impact of sampling on the performance of the flow-based anomaly detector. Our proposed flow sampling technique shown in Fig. 1 determines the required sampling method based on the feedback from the flow-based anomaly detector.

To evaluate the performance of the proposed sampling technique, several packet-based datasets, CAIDA DDoS datasets, CAIDA Traces 2013, and DARPA datasets, are used in this study to generate flow-based datasets [26-28].

The remainder of this paper is organized as follows. Section II critically examines the related works. Section III describes the architectural components of our proposed technique. Section IV describes metaheuristic algorithms used in our flow-based anomaly detector. Section V discusses the proposed sampling technique and then, the datasets used in this paper are explained in Section VI. Section VII provides the experimental results and discusses the strengths and limitations of the proposed solutions. Section VIII concludes the paper.

II. RELATED WORKS

In the last decade, flow-based analysis has been considered by many researchers as a suitable solution for anomaly detection in high-speed networks. The overview of packet-based and flow-based intrusion detection in high-speed networks is given in [13, 29]. An on-line DoS resilient flow-level intrusion detection system for high-speed networks called HiFIND [30] investigates the security of flow-based detection. It uses 2D sketches to detect anomalies. The proposed model can distinguish SYN flooding from different port scans. An important problem in payload-based systems is the spread of encrypted protocols. An improved intrusion detection method [31] is used to detect misuse in encrypted protocols using packet headers.

Because of the flexibility of machine learning methods for learning and detecting new attacks, researchers have considered the application of intelligent methods in high-speed networks [32]. Alshammari and Zincir-Heywood [33] use five learning algorithms such as SVMs and C4.5 to classify SSH- and Skype-based encrypted traffic using flow-based features. The results show that C4.5 gives the best classification performance.

GSA [20] is a swarm based metaheuristic algorithm which is based on Newtonian gravity. GSA is proposed to overcome the slow convergence and local minima problems in traditional training methods in ANNs. An adaptive learning rate, a memory-less algorithm, and fast convergence are important advantages of GSA as compared with similar algorithms such as PSO, and real genetic algorithm [20].

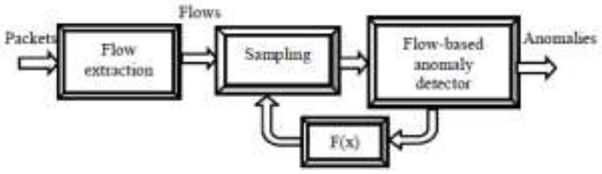


Figure 1. Proposed sampling technique

GSA is used in various flow-based anomaly detection systems [12, 14, 34] to analyse the high volume of flow records. An algorithm based on GSA and PSO (PSOGSA) [25] is used to improve the classification of a flow-based dataset [12].

A dataset called the Winter dataset is introduced in [2] to train a one-class SVM (OC-SVM) to detect malicious flows. The flow-based anomaly detection systems in [12, 14, 34] are also evaluated by means of Winter datasets. Their results confirm that a GSA-based MLP can detect malicious flow traffic in high-speed networks. The flow-based DARPA dataset is another dataset proposed [35]. A prototype is designed [35] with a hybrid software-enabled detection engine based on an improved block-based neural network (BBNN). To provide real-time intrusion detection, this prototype is integrated with a high frequency field programmable gate arrays (FPGA) board. The method is evaluated by the flow-based DARPA dataset. This dataset is also used to evaluate two metaheuristic algorithms in [12].Winter datasets, DARPA datasets, and CAIDA datasets are used in our study.

There are few papers which investigate flow anomaly detection with sampled data. Traffic statistics such as the mean rate and the flow size distribution are affected by sampling methods [36, 37]. Each flow has several features. The size of a flow shows the number of packets in the flow and it is a very important feature for sampling. A significant number of packets are present in large-sized flows, which are important for monitoring. Therefore, the traditional sampling methods are biased toward large-sized flows. However, these methods can corrupt anomaly detection because anomalies often have flows with small sizes [38].

A flow sampling method has good accuracy compared with packet sampling and it can preserve the flow distribution better. However, it needs more memory and CPU power [24, 39]. Various methods have been proposed to decrease memory requirements, for example, smart sampling [40] and sample-and-hold [41]. However, these methods are biased for large flows. Therefore, they cannot capture small flows, which are the source of a large number of attacks [1]. An investigation [1] shows the effect of four sampling methods on the performance of a wavelet-based volume anomaly detection method and two port scan detection algorithms. All four sampling methods degrade the detection rate and the false positive rate of anomaly detection methods.

The impact of sampling on anomaly detection has been investigated in a number of studies [1, 24, 42]. In our study, we have made further enhancements to improve the performance of anomaly detection in sampled traffic.

Intelligent flow sampling (IFS) [42] uses two-stage flow sampling to reduce the negative impact of sampling on traffic analysis. The first stage of IFS is responsible to extract the features required for analytic algorithms, and an adaptive sampling algorithm is proposed for the second stage. The adaptive sampling method focuses more on the flows with rare features to improve anomaly detection in sampled traffic. Another two-stage flow sampling technique is proposed in [43] in which a GSA-based classifier is investigated in detecting anomalies in sampled traffic. Then, a flow sampling method is proposed to improve the detection rate. In this system, the first stage is the feature extraction, which is responsible to extract information of the flow size. A sampling method, which is an optimized selective sampling, is proposed for the second stage of the sampling technique.

In respect of anomaly detection, selective sampling is proposed [44]. Selective sampling and its impact on a sequential non-parametric change-point anomaly detection method is studied in [44]. The results show that even with a small sampling rate, the performance of the anomaly detection method is improved compared with random flow sampling and smart sampling. However, this method only focuses on small flows and it loses significant information included in large flows.

It is shown [38] that in DDoS attacks, worms, and port scans which are small in size, selective sampling is valuable and it can preserve the changes in the entropy of small flows. In contrast, smart sampling is suitable for large flows and it can preserve the majority of flow data [38]. The effectiveness of selective and smart sampling is discussed in several papers [38, 45]. In our study, the combination of these methods enhances efficiency.

III. Architectural Components

The architectural components of our proposed technique are shown in Fig. 1. There are four main modules in this technique: flow extraction, sampling, flow-based anomaly detection, and feedback.

A. Flow Extraction

The flow extraction module is responsible for generating flow traffic which is a prerequisite for detecting malicious flows. This module is a NetFlow simulator. It receives packet-based datasets and generates flows. NetFlow has two components: NetFlow exporter and NetFlow collector. Softflowd [46] and Flowd [47] are open source software which simulate NetFlow exporters and collectors respectively [35]. Softflowd can generate flow records by reading a packet-based captured file. Then these flow records are sent to a NetFlow collector, Flowd. Flowd has a tool called Flowd-reader. This tool reads the following flow fields: source/destination IP, source/destination port, packets, octets, start and end time, flags and protocol. In this study, Softflowd and Flowd are used to generate flow-based datasets.

B. Sampling Module

Researchers have realized that the distribution of traffic features is distorted by sampling procedures [1]. This negatively affects different volume anomaly detection techniques. This study proposes a modified flow sampling technique to improve anomaly detection in sampled traffic. Smart and selective sampling methods are the main components of our proposed technique and they will be discussed in Section V. In our technique, the flow anomaly detector can accurately determine whether there is an attack. The results show which small or large flows are important and then a suitable sampling method is selected by means of feedback from the anomaly detector.

C. Flow-Based Anomaly Detection and Feedback Modules

The flow traffic from the sampling module is fed to the flow-based anomaly detection module, which is based on an ANN classifier. A lightweight MLP, which is a supervised ANN method, is used for this purpose. MLPs have high accuracy in the classification of input traffic. However, the problem of local minima in their traditional training algorithms is addressed by different metaheuristic algorithms. Two important components of metaheuristic algorithms are the selection of the best solutions and randomization. Convergence to optimality is ensured by the first component. Randomization avoids trapping at local optima and enhances the diversity of the solutions. Combining the components appropriately ensures that global optimality is achievable. A metaheuristic algorithm, the hybrid PSOGSA, is used in this study to train MLP to improve the accuracy and reliability [12].

The flow-based anomaly detection module can detect volume anomalies such as DDoS, DoS, worms, botnets, and port scans. When this module detects an anomaly, the feedback module identifies a specific metric which shows which sampling method is required. The sampling module selects the appropriate method using this metric. In this study, the sampling method can be smart or selective.

IV. Algorithms

PSOGSA algorithm is used in the flow-based anomaly detection module to train the MLP-based anomaly detector. Its results are compared with cuckoo algorithm.

A. PSOGSA Algorithm

GSA [20] is a metaheuristic method based on the Newton law of gravity. This algorithm performs an efficient and accurate search for the global optimum. A hybrid of the PSO and GSA algorithms called PSOGSA is proposed [25] in which PSOGSA is faster than GSA in terms of converging speed. This hybrid algorithm combines the ability of social thinking (gbest) in PSO and local search capability of GSA. First, all agents are initialized and then gravitational force, gravitational constant and force among agents [25] are calculated using (1), (2) and (3) respectively. Where, M_a and M_p are active and passive gravitational mass respectively. R_{ij} is the Euclidean distance between agent *i* and *j*. x_i shows the position, *G* is a gravitational constant and G_0 is the initial gravitational constant. Current iteration and maximum number of iterations are shown by *iter* and *maxiter*, and *α* is the descending coefficient. F_i is the total force on agent *i* and *rand* is a random number in interval [0 1].

$$F_{ij}^{d}(t) = G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(t)+\varepsilon}(x_j^{d}(t)-x_i^{d}(t)) \qquad (1)$$

$$G(t) = G_0 \times (-\alpha \times \frac{\text{iter}}{\text{maxiter}}) \quad (2)$$

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \quad (3)$$

The acceleration of each particle is defined as in (4). Where, M_i is the mass of object *i*. Next, the velocities are calculated using (5). Where, $c_j^{'}$ is the acceleration coefficient and *w* is weighting function. The acceleration of the agent *i* is $ac_i(t)$ which is at the iteration *t*. The last variable is *gbest*, which is the best solution. The positions of agents are updated after each iteration, defined as in (6). The process of updating the velocity and the position stops when the final criterion is met [25, 12].

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)'} \quad (4)$$

$$\begin{aligned} v_i(t+1) &= w \times v_i(t) + c_1^{'} \times rand \times ac_i(t) + \\ & c_2^{'} \times rand \times (gbest - x_i(t)) \end{aligned} \quad (5)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

B. Cuckoo Optimization Algorithm

Cuckoo optimization algorithm (COA) [48] is a population-based algorithm which is inspired by the life of the cuckoo which has unique characteristics in terms of egg-laying and breeding. First, there is an initial population in the COA. The survival competition among cuckoos is the basis of the COA algorithm. Mature cuckoos lay their eggs in other birds' nests. If the host bird does not kill the eggs, they will become mature birds. More profit is gained in an area which has more survivals. The COA algorithm continues to find the best position with the maximum profit value. The best position will be the destination of all cuckoo societies. An array related to the value of problem variables in COA is called “habitat”. The dimension of the optimization problem is $\mathrm{N_{var}}$ and the habitat array is $1 \times \mathrm{N_{var}}$ which is defined as in (7). The array shows the current position of cuckoos. Where, $(\mathrm{x_1, x_2, \ldots, x_{N_{var}}})$ are floating point numbers. $\mathrm{f_p}$ is the profit function for calculating the profit of a habitat, defined as in (8). The COA maximizes a profit function. If the COA is used in cost minimization problems, (8) will be changed to (9) [12, 48].

$$\text{habitat}=[\mathrm{x_1, x_2, \ldots, x_{N_{var}}}] \quad (7)$$

$$profit = f_p(habitat) = f_p(x_1, x_2, \ldots, x_{N_{var}}) \quad (8)$$

$$profit = \textit{-cost(habitat)} = -\mathrm{f_c}(\mathrm{x_1, x_2, \ldots, x_{N_{var}}}) \quad (9)$$

Each cuckoo can lay 5 to 20 eggs. These values show the upper and lower limits of eggs for each cuckoo, $\mathrm{var_{hi}}$ and $\mathrm{var_{low}}$. Each cuckoo has an egg laying radius (ELR) as defined in (10). K-means clustering is used in the COA for the grouping of cuckoos. COA ends when more than 95% of cuckoos converge in the same habitat [48]. COA is compared with PSO and genetic algorithm in [48] which shows that COA has merit in terms of convergence speed and global minima achievement.

$$ELR = \alpha \times \frac{\text{Number of current Cuckoo eggs}}{\text{Total number of eggs}} \times (\mathrm{var}_{hi} - \mathrm{var}_{low}) \quad (10)$$

V. Proposed Sampling Technique

This study proposes a sampling technique based on an ANN-based flow anomaly detector. This technique is used for two purposes. It is evaluated to detect anomalies and used to determine the sampling type, as discussed below.

A. ANN-Based Flow Anomaly Detection System

The proposed flow anomaly detection system uses a two-layer MLP. The two-layer MLP has one hidden layer and an output layer. The MLP has three nodes in the hidden layer and two nodes in the output layer. Two output nodes perform the classification of the flow-based traffic into malicious and benign subsets. To avoid local minima and to improve performance, a metaheuristic algorithm, PSOGSA algorithm, is used to optimize the interconnection weights of the MLP neural network. Then, the results are compared with the cuckoo algorithm.

The proposed system is implemented in MATLAB version R2012a (7.14.0.739). To determine the optimum values for the weights, PSOGSA algorithm generates an initial population of masses. The weight coefficients of MLP correspond to masses. The displacement of a mass in space shows updating the weight coefficients to decrease mean square error (MSE). After the calculation of the new velocity in each step, the positions of all masses are updated. These new positions correspond to new weights which can be used to calculate MSE. The mass with the least MSE is the best. Training is finished if it achieves an acceptable error or a maximum number of iterations [12].

In the cuckoo algorithm, the weights of MLP correspond to habitats. This algorithm also starts with an initial population. Then, cuckoos immigrate to areas with more profit in each step until the habitat with maximum profit is obtained. The new habitats are the new weights in the MLP [48]. The parameters of the PSOGSA and cuckoo algorithms are shown in TABLE I. The optimized MLP is able to detect known and unknown attacks after training.

Pre-processing is needed since MLP cannot be trained with the datasets in their original forms. Datasets should be scaled to [-1; +1]. The min-max normalization method is used, defined as in (11). The data are rescaled to the range of (t_{min}, t_{max}). $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of a feature [14, 49].

$$x^{'} = (t_{\max} - t_{\min}) \times \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} + t_{\min} \quad (11)$$

B. Sampling

Depending on the sampling point, there are two categories [50]: an on-line method which samples at NetFlow exporters when they capture packets, and an off-line method which samples flows at NetFlow collectors after receiving flows from NetFlow exporters. As this study works with flows, it proposes an off-line method in which the data communication from the NetFlow collector to the analysing point is reduced.

Flow-based traffic entails heavy-tailed distribution for packets [37]. In this distribution, a large number of flows are small and a small number of them are large. Although there are few large flows, they carry the majority of packets. In this regard, many sampling methods are biased toward large flows because they are important for efficient bandwidth monitoring.

Two well-known flow sampling methods which are used in this study are smart sampling and selective sampling. Our proposed off-line sampling technique provides the sampled traffic based on the network conditions. This method is a hybrid of selective and smart sampling methods.

Smart sampling targets large flows to preserve more traffic information desired for monitoring purposes whereas selective sampling is suitable for anomaly detection. Therefore, our proposed method can sample both small and large flows based on the dictation of the flow-based anomaly detector. In other words, the type of the sampling method is determined using the output of the flow-based anomaly detection system. Initially, it is assumed that the traffic which should be sampled is benign traffic and smart sampling is the default method for producing sampled flow traffic. If an anomaly is detected, the proposed technique changes the sampling method to the selective sampling method. Fig. 2 shows the flowchart of the proposed sampling technique. A fixed sampling rate is chosen in this study. This rate shows the number of sampled flows.

TABLE I. PARAMETERS OF CUCKOO AND PSOGSA ALGORITHMS [12]

PSOGSA	Cuckoo
Number of masses =10 G_0=100 Alfa (α) = 20 $c_1^{'}$ =2 , $c_2^{'}$ =2	Initial number of cuckoos = 20 Minimum number of eggs = 5 Maximum number of eggs = 10 Upper limit for variables = 20 Lower limit for variables = -20
Total number of iterations (T)= 150	

Smart sampling: One type of sampling is probabilistic sampling in which each flow is sampled with a certain probability. An important probabilistic flow sampling is smart sampling which calculates the probability of flows based on their sizes [38]. The probability of each flow is defined according to (12), where x is the flow size in packets and z is a threshold. According to (12), flows that are larger than z are sampled with probability 1. On the other hand, flows smaller than z are sampled with probability proportional to their size. In smart sampling, the sampling rate is controlled by the sampling threshold.

$$p(x) = \begin{cases} x/z & x < z \\ 1 & x \geq z \end{cases} \tag{12}$$

Selective sampling: Selective sampling targets small flows [44]. It has been shown that small flows are usually the source of many network attacks (e.g., DDoS, port scans, worm propagation). Therefore, for better anomaly detection performance, these flows should be preferentially selected. The selective sampling method selects an individual flow with probability *p(x)*, as shown in (13):

$$p(x) = \begin{cases} c & x \leq z \\ z/(n \cdot x) & x > z \end{cases} \tag{13}$$

where x is the flow size in packets, $0 < c \leq 1, n \geq 1$ and z is a threshold. According to (13), flows that are smaller than z are sampled with a constant probability c. On the other hand, flows larger than z are sampled with probability inversely proportional to their size. If the parameter c is defined appropriately, a significant proportion of small flows can be selected without decreasing anomaly detection effectiveness. In addition, increasing the value of parameter n may cause the selection of large flows to be further decreased [38].

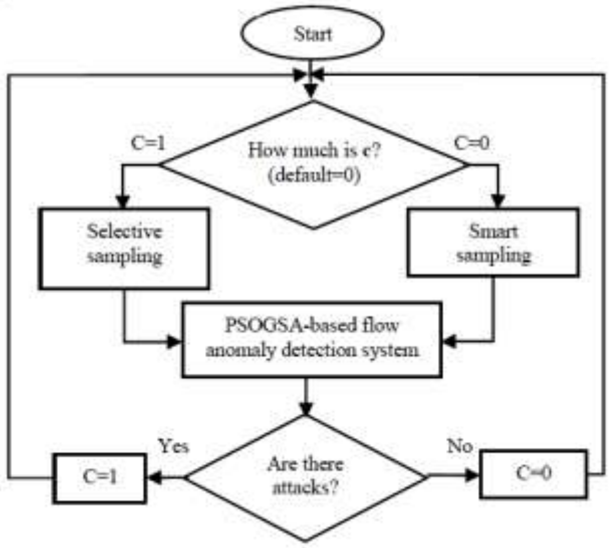


Figure 2. Flowchart of proposed sampling technique

VI. DATASETS

Different flow-based datasets are required to examine our proposed technique. In this study, Winter datasets, DARPA datasets, and CAIDA datasets are used for the evaluation of the proposed technique [26-28]. The first public flow-based dataset, which we call Sperotto dataset [51] was captured by monitoring a honeypot hosted in the University of Twente network. Three services were installed on the honeypot: SSH (OpenSSH) service, FTP and Apache web server. The large number of flows in the Sperotto dataset required a time-consuming training phase. Therefore, it was modified as the Winter datasets [2]. In this case, unlabelled flows, duplicate flows and protocols other than SSH and HTTP were deleted. The modified datasets contain 22,924 malicious flows. All collected flows in the Sperotto dataset were malicious but the benign flows were also required for measuring the performance. To capture the benign traffic in the Winter datasets, a tcpdump was used. The benign dataset contains HTTP, SSH, DNS, ICMP and FTP. The Winter datasets are used in this study. TABLE II presents the distribution of these datasets.

In the Winter datasets, the number of malicious flows is strongly overrepresented compared with benign flow records. This may affect the reliability of our results. Therefore, to provide a more accurate evaluation of our proposed technique, another dataset, which we call the single-host flow DARPA (SHF_DARPA) dataset, is used. This dataset is extracted from packet-based DARPA datasets, which have a variety of attack types [35]. TABLE II shows the detailed information about the SHF_DARPA. It is called single-host because it focuses only on those flows sent to host 172.16.112.50 which receives most attacks. Winter and SHF_DARPA datasets are used to evaluate our flow-based anomaly detector [12].

TABLE II. DISTRIBUTION OF FLOW-BASED DATASETS

		Benign Flows	Malicious flows	Total Traffic
Winter datasets	Training dataset	962	15236	16198
	Testing dataset	942	7688	8630
	Total traffic	1904	22924	24828
SHF_DARPA datasets	Training dataset	59980	5952	65932
	Testing dataset	45053	18586	63639
	Total traffic	105,033	24,538	129,571
Complete flow DARPA datasets	Week1	485877	…..	485877
	Week2	……	406284	406284
	Week3	483139	…….	483139
	Total traffic	969016	406284	1375300
Flow-based CAIDA datasets	CAIDA DDoS	……	32638466	32638466
	CAIDA Internet Traces 2013	4273773	……	4273773
	Total traffic	4273773	32638466	36912239

In respect of sampling, however, the complete version of flow DARPA datasets, which include all hosts, is required. On the other hand, the Winter datasets do not have enough flows to be used for sampling purposes. Therefore, we need to generate our flow-based datasets. Fig. 3 shows the procedure of generating datasets. Softflowd and Flowd software perform NetFlow simulation [35]. NetFlow simulators generate two different flow-based datasets in this study: complete flow-based DARPA datasets which have the traffic of all hosts, and flow-based CAIDA datasets. TABLE II provides the distribution of the generated datasets. In this study, each flow record has seven features: a) Packets; b) Octets; c) Duration; d) Source port; e) Destination port; f) TCP flags; and g) IP protocol.

A. Characteristics of Complete Flow-Based DARPA Datasets

This study examines DARPA datasets [28] to identify the characteristics of the real traffic. In DARPA, first-week and third-week datasets are attack-free datasets whereas the second-week dataset contains labelled attacks. In this study, all existing flows in these datasets are extracted for the evaluation of the proposed technique. Fig. 4 shows the distribution of normal and malicious flows, based on their sizes, in the complete flow-based DARPA datasets. The cumulative number of packets shows how many packets are included in the flows so far.

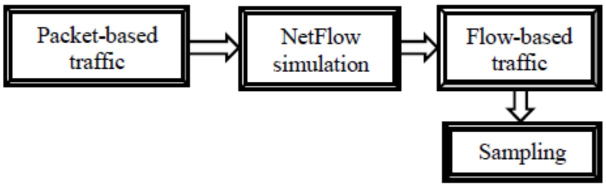
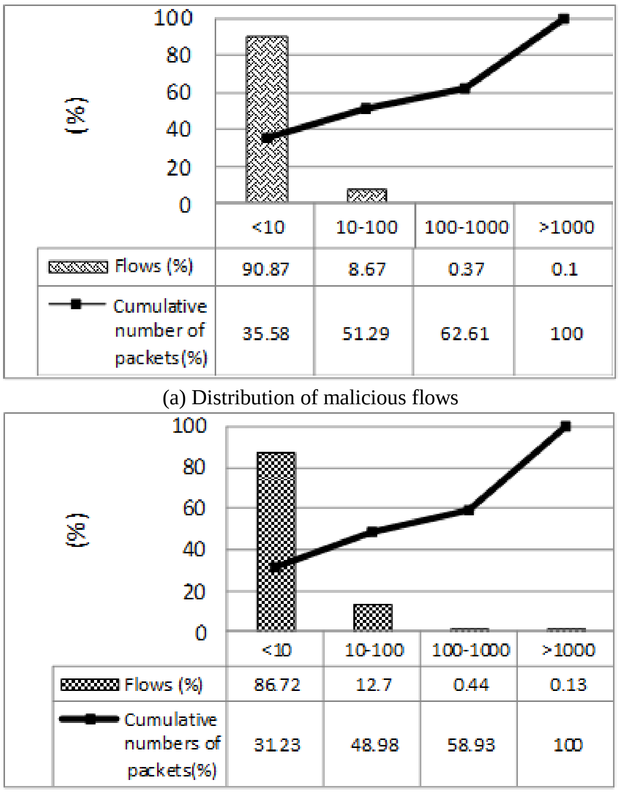


Figure 3. NetFlow simulation



(a) Distribution of malicious flows

(b) Distribution of benign flows

Figure 4. Distribution of flows in the complete flow-based DARPA datasets

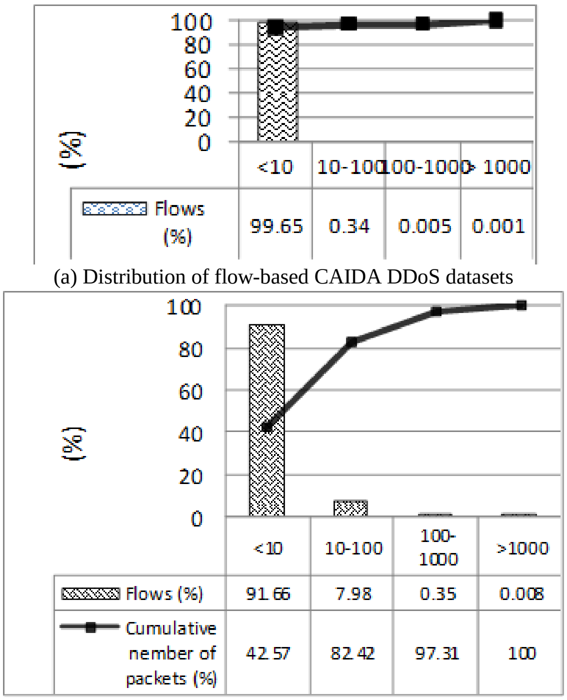
According to this figure, the concentration of the traffic is in flows of small size but large flows containing the majority of packets are rare. In respect of anomaly detection, sampling small flows helps to preserve more attack information.

B. Flow-based CAIDA Datasets

The second dataset used in this study is called CAIDA, which includes CAIDA DDoS datasets and CAIDA Internet Traces 2013 [26, 27]. CAIDA DDoS datasets contain one hour of DDoS attack traffic. As the datasets are very large, four CAIDA DDoS datasets are randomly selected in this study and converted to flow-based datasets. Fig. 5 (a) shows the distribution of the generated flow-based CAIDA DDoS datasets. As DDoS attacks generate small flows, almost all DDoS flow sizes number fewer than 10. On the other hand, CAIDA Internet Traces are normal traces, which are assumed not to have DDoS attacks. Fig. 5 (b) provides information about flows generated from CAIDA Internet Traces 2013. This again confirms the reverse proportion of the frequency and the flow size.

VII. Results and Discussion

To measure the performance of our proposed flow-based anomaly detector, four metrics are used: accuracy, error rate (ER), miss rate (MR) and false alarm rate (FAR). These metrics are defined as in (14), (15), (16), and (17).



(a) Distribution of flow-based CAIDA DDoS datasets

(b) Distribution of flow-based CAIDA Internet Traces 2013

Figure 5. Distribution of flow-based CAIDA datasets

True positive (tp) and true negative (tn) correspond to correct detection of malicious and benign traffic respectively. False positive (fp) shows the error in the detection of benign traffic and false negative (fn) is the wrong detection of malicious traffic [12, 52].

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{14}$$

$$ER = \frac{f_n + f_p}{t_p + t_n + f_p + f_n} \tag{15}$$

$$MR = \frac{f_n}{t_p + f_n} \tag{16}$$

$$FAR = \frac{f_p}{t_n + f_p} \tag{17}$$

High MR means there are a lot of undetected intrusions. On the other hand, FAR causes an anomaly detector to generate false alarms. A rise in the number of alarms that needs to be analysed causes important alarms to be ignored. Therefore, limiting FAR in anomaly detection is a priority [2]. The selected parameters in TABLE I give the lowest FAR [12]. To provide a comprehensive evaluation, it is initially assumed that there is no sampling. Then, Winter and SHF_DARPA datasets are pre-processed and used to train and test the optimized MLP-based anomaly detectors.

Our study requires an accurate classifier for the flow-based anomaly detector because the results of the anomaly detector determine the type of sampling method. In this regard, we use a PSOGSA-based MLP classifier. PSOGSA, which is the combination of GSA and PSO, has better convergence speed compared to GSA and PSO individually [25].

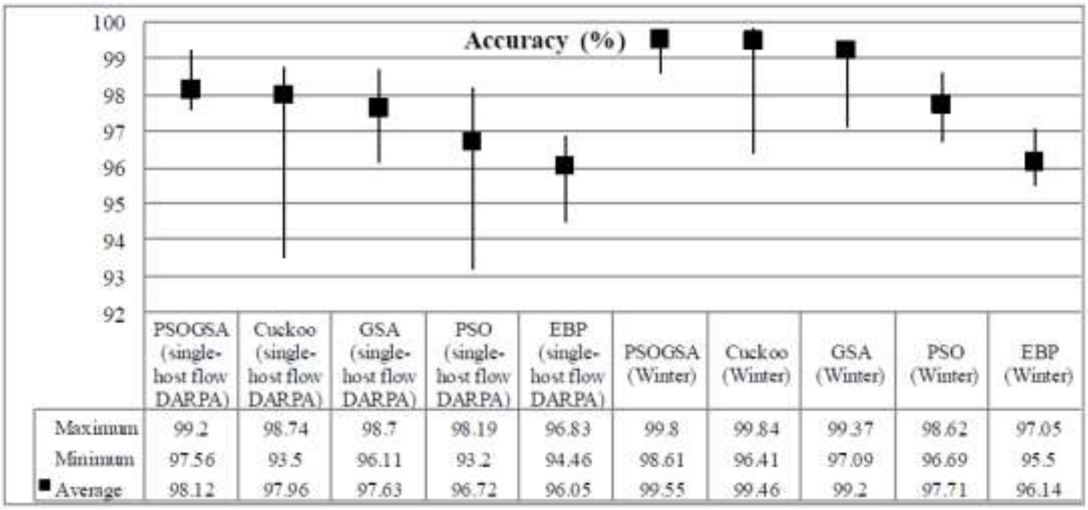
TABLE III and TABLE IV compare the performance of our PSOGSA with several metaheuristic algorithms. Our PSOGSA-based MLP is also compared with other studies which use the same datasets as our study (see TABLE III and TABLE IV). All of the optimized MLPs are capable of detecting unknown attacks. As it is shown in TABLE IV, both PSOGSA and cuckoo-based systems have high accuracy in terms of classifying benign and malicious traffic [12]. However, PSOGSA-based MLP, chosen in this study, provides better accuracy and lower FAR.

TABLE III. PERFORMANCE METRICS OF DIFFERENT ALGORITHMS [12]

Dataset	Detector	tp	tn	fp	Fn	Training Time (s)
SHF_DARPA	PSOGSA based MLP	17927	44515	538	659	4.13e+03
	Cuckoo based MLP	17883	44456	597	703	7.91e+01
	GSA based MLP	17775	44356	697	811	3.17e+03
	PSO based MLP	17414	44138	915	1172	2.96e+03
	EBP based MLP	17177	43949	1104	1409	2.10e+01
Winter	PSOGSA-based MLP	7651	940	2	37	2.63e+03
	Cuckoo based MLP	7650	933	9	38	4.56e+01
	GSA based MLP	7636	925	17	52	1.56e+03
	PSO based MLP[34]	7493	939	3	195	…
	EBP based MLP[34]	7367	930	12	321	…

TABLE IV. PERFORMANCE COMPARISON OF DIFFERENT FLOW-BASED ANOMALY DETECTORS [12]

Dataset	Detector	Accuracy (%)	ER (%)	MR (%)	FAR (%)
SHF_DARPA	PSOGSA-based MLP	98.12	1.88	3.55	1.19
	Cuckoo based MLP	97.96	2.04	3.78	1.33
	GSA based MLP	97.63	2.37	4.36	1.55
	PSO based MLP	96.72	3.28	6.31	2.03
	EBP based MLP	96.05	3.95	7.58	2.45
	BBNN [35]	99.92	3.18	...	5.14
	SVM (RBF) [35]	92.07	6.56	…	5.20
	SVM (sigmoid) [35]	99.73	3.54	…	5.59
	Na ïve Bayes [35]	46.83	23.02		2.49
Winter	PSOGSA based MLP	99.55	0.45	0.48	0.21
	Cuckoo based MLP	99.46	0.55	0.49	0.96
	GSA based MLP	99.20	0.80	0.68	1.81
	PSO based MLP [34]	97.71	2.29	2.54	0.32
	EBP based MLP [34]	96.14	3.86	4.18	1.27
	OC-SVM [2]	98.29	…	4.71	0
	KNN [53]	91.31	…	…	0
	Liblinear [53]	91.37	…	…	5.9
Complete flow DARPA	PSOGSA based MLP	97.63	…	3.52	0.95
Flow-based CAIDA	PSOGSA based MLP	97.16	…	1.27	1.42



	PSOGSA (single-host flow DARPA)	Cuckoo (single-host flow DARPA)	GSA (single-host flow DARPA)	PSO (single-host flow DARPA)	EBP (single-host flow DARPA)	PSOGSA (Winter)	Cuckoo (Winter)	GSA (Winter)	PSO (Winter)	EBP (Winter)
Maximum	99.2	98.74	98.7	98.19	96.83	99.8	99.84	99.37	98.62	97.05
Minimum	97.56	93.5	96.11	93.2	94.46	98.61	96.41	97.09	96.69	95.5
■ Average	98.12	97.96	97.63	96.72	96.05	99.55	99.46	99.2	97.71	96.14

Figure 6. Accuracy with ten trials

TABLE V. COMPARISON OF DIFFERENT SAMPLING METHODS IN PRESERVING MALICIOUS FLOWS

	Complete flow DARPA datasets		Flow-based CAIDA datasets		Flow-based CAIDA datasets	
			Train CAIDA_DDoS	Train CAIDA_traces	Test CAIDA_DDoS	Test CAIDA_traces
	Malicious	Benign	DDoS_143436	Sanjose-2013221 Chicago-2013815	DDoS_144436 DDoS_145436	Sanjose-2013718 Chicago-2013718
Total flows	406284	969016	2659763	346022	6792168	1467105Z
Randomly selected	59082	140918	59082	140918	59082	140918
Sampling rate	0.1 (20000 flows)				0.1 (20000 flows)	
Selective sampling	12401 malicious flows (22.99 %)		…		14361 DDoS flows (24.31%)	
Smart sampling	8088 malicious flows (13.69 %)		…		7090 DDoS flows (12 %)	
Proposed technique	12125 malicious flows (20.52 %)		…		14170 DDoS flows (23.98%)	
Random sampling [43]	15 %				20 %	
Smart sampling [43]	10 %				7 %	

PSOGSA-based MLP has the highest accuracy and a low FAR in TABLE IV, whereas the EBP has the lowest training time, TABLE III. The only methods which give lower FAR compared with PSOGSA are OC-SVM [2] and KNN [53] but they provide lower accuracy. The OC-SVM in this table is trained with only malicious flows of the Winter datasets. This affects the FAR of OC-SVM which achieves zero FAR.

We compare the accuracies of PSOGSA with other metaheuristic algorithms in ten similar trials. The distribution of the accuracies is shown in Fig. 6. Results indicate that PSOGSA has the least variation, showing that it has the most reliable performance compared to other methods. Accordingly, we use PSOGSA-based MLP for our ANN anomaly detection system.

In TABLE IV, the PSOGSA-based MLP is separately trained and tested with the complete flow-based DARPA and flow-based CAIDA datasets. Then, it is used to decide about the required sampling method. All results are averaged over ten experiments.

A. Impact of Sampling Methods

The proposed sampling technique is evaluated with the complete flow-based DARPA datasets and the flow-based CAIDA datasets which are generated in this study. The results are compared with other sampling methods and with another study. TABLE V compares different sampling methods in preserving malicious flows.

Because of the huge amount of complete flow-based DARPA and flow-based CAIDA data, initially, 200,000 flows are randomly selected from each dataset, TABLE V. The proportion of malicious to benign flows is the same for both selected datasets, and it is equal to that of complete flow DARPA dataset. Similar to NetFlow, a fixed sampling rate, which is 0.1, is selected in this study. In complete flow-based DARPA dataset, training and testing datasets are different but they are the same size.

Selective sampling is the method proposed to improve anomaly detection in sampled flow traffic. According to TABLE V, the results from the proposed sampling and selective sampling were very close. In flow-based CAIDA datasets, they sample twice as many malicious flows as smart sampling. Therefore, the proposed technique is almost as effective as selective sampling for anomaly detection. However, the advantage of the proposed technique is that it samples large flows in normal situations of a network, and hence it can preserve more packets than selective sampling. Therefore, it can be used in both traffic monitoring and anomaly detection.

Selective sampling always has bias toward small flows [38, 44]. Sampling small flows in all time slots causes the loss of most large flows; hence, selective sampling loses the majority of packets carried by large flows [38]. On the other hand, smart sampling mostly samples large flows and it does not effectively sample small flows which are the source of a large number of anomalies [24]. Although there is poor detection of anomalies when using smart sampling, this method can preserve the majority of packets and it is desired for monitoring tools [38]. Our technique proposes switching between selective sampling and smart sampling methods based on specific network situations. The traffic of each time slot is considered and the suitable sampling method is defined for flows in the following time slot.

The performances of sampling methods in preserving malicious flows are compared in TABLE V. The performance of the proposed method is also compared with another study in TABLE V. For more evaluation, Fig. 7 compares these methods based on the number of preserved packets.

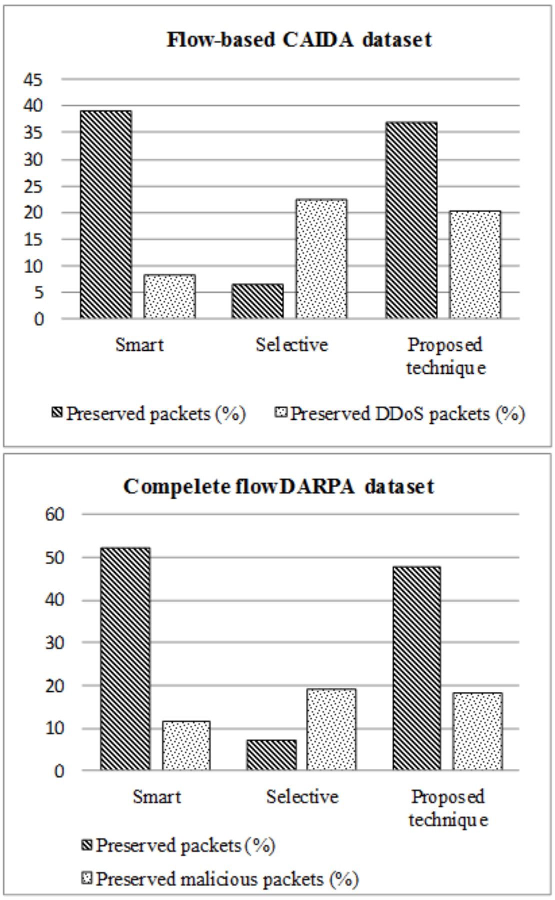


Figure 7. Comparison between preserved packets of different sampling methods (sampling rate 0.1)

This figure shows that the efficiency of our technique in preserving packets is almost equal to that of smart sampling. In addition, selective sampling helps our technique to sample malicious packets carried by malicious flows [38]. Therefore, for both flow-based CAIDA and complete flow DARPA datasets, the percentage of malicious packets preserved by our proposed technique is close to that of selective sampling (see Fig. 7). The results confirm that the proposed technique can be used in traffic monitoring and anomaly detection.

The important issue is that a number of attacks are lost in the sampling methods. An ideal sampling should decrease the loss of attacks. In this study, IP addresses, which are the destination of malicious flows, are called attacked IPs. On this assumption, TABLE VI compares the number of sampled attacked IPs. Smart sampling has the lowest sampled attacked IPs and it loses many attacks. On the other hand, the proposed technique and selective sampling can preserve most attacked IP.

In addition, the performance of our sampling technique in preserving traffic information is evaluated for different sampling rates and it is compared with other sampling methods (see Fig. 8). Our technique can preserve packets similar to smart sampling in all sampling rates, however, its sampled malicious flows are significantly better and it is almost equal to that of selective sampling. This figure shows an increase in the sampling rate helps the proposed technique to save more malicious data; therefore, the accuracy of the flow-based anomaly detector will be increased.

In total, the proposed technique can sample large flows, which are mostly ignored in selective sampling. Monitoring tools usually sample large flows; therefore, this technique is a good option for monitoring purposes. In addition, it can sample small malicious flows. Thus, it captures flows required for flow-based anomaly detection.

TABLE VI. COMPARISON OF DIFFERENT SAMPLING METHODS IN PRESERVING ATTACKED IPS

	Sampling method	Sampled attacked IP	Un-sampled attacked IP
Flow-based CAIDA	Original traffic	208	0
	Selective sampling	151	57
	Smart sampling	45	163
	Proposed technique	144	64
Complete flow DARPA	Original traffic	186	0
	Selective sampling	175	11
	Smart sampling	116	70
	Proposed technique	172	14

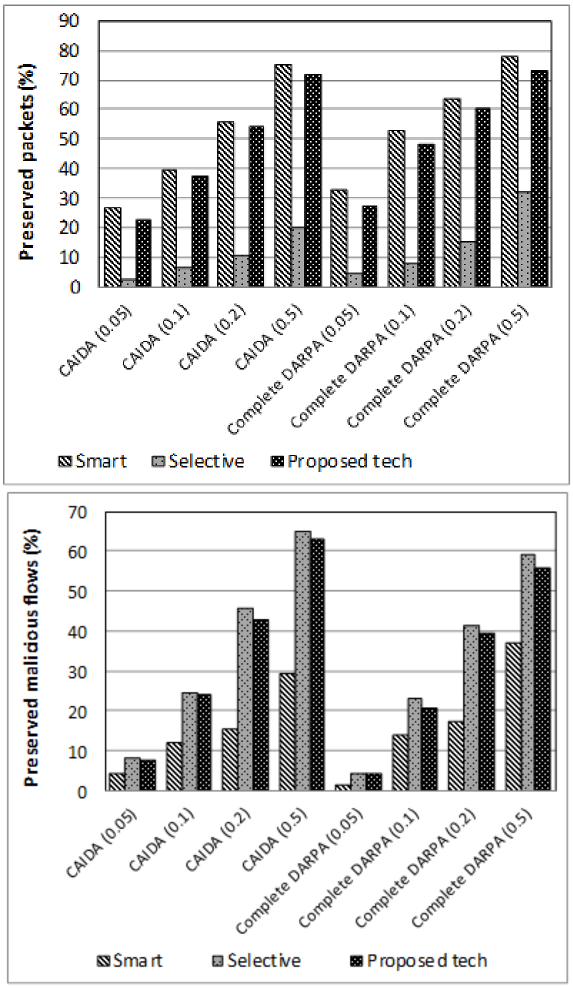


Figure 8. Comparison of proposed technique and other sampling methods with different sampling rates

VIII. CONCLUSION

In this study, we improved anomaly detection in flow data. An MLP neural network was employed as a flow-based anomaly detector in which a metaheuristic algorithm called PSOGSA was deployed to optimize the interconnection weights of the MLP. The results confirmed that this supervised method could provide high accuracy in classifying benign and malicious flows and it had a low false alarm rate in flow traffic. Sampled flow traffic is a monitoring solution which is widely used in computer networks. The negative impact of sampling on anomaly detection is a challenging issue. In this regard, we proposed a technique to improve the performance of the anomaly detector in sampled flow traffic. In this technique, we selected the sampling type based on the output of the flow-based anomaly detector. The proposed technique could efficiently detect anomalies in sampled traffic and its percentage of sampled malicious flows was improved by about 7%. In addition, it could decrease the loss of information. Therefore, this technique can be used for both anomaly detection and monitoring. One limitation in the evaluation of the flow-based anomaly detection methods is the lack of public datasets. In this study, we generated two flow-based datasets, complete flow-based DARPA and flow-based CAIDA datasets. In future work, in addition to the flow size, source/destination IP/port will be taken into consideration as important metrics in the sampling method because the distribution of these metrics changes during different attacks.

ACKNOWLEDGMENT

The authors would like to thank UNSW@ADFA for providing the single-host flow-based DARPA datasets used in this research. They are also grateful to CAIDA for two datasets provided for generating flow-based CAIDA datasets.

REFERENCES

[1] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, “Is sampled data sufficient for anomaly detection?,” *in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement,* 2006, pp. 165-176.

[2] P. Winter, E. Hermann, and M. Zeilinger, “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*,2011, pp.1-5.

[3] A. Sperotto and A. Pras, “Flow-based intrusion detection,” *in Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, 2011, pp. 958-963.

[4] Li, Bingdong, Jeff Springer, George Bebis, and Mehmet Hadi Gunes. “A survey of network flow applications.” *Journal of Network and Computer Applications* 36, no. 2 (2013): 567-581.

[5] S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows using hidden Markov model," *computers & security*, vol. 22, pp. 45-55, 2003.

[6] D. Lei, C. You, and Y. Xiaochun, "Optimizing IP flow classification using feature selection," *in Parallel and Distributed Computing, Applications and Technologies,*

2007. PDCAT'07. Eighth International Conference on, 2007, pp. 39-45.

[7] X. Li and Z.-H. Deng, "Mining frequent patterns from network flows for monitoring network," *Expert Systems with Applications*, vol. 37, pp. 8850-8860, 2010.

[8] A. Shahrestani, M. Feily, R. Ahmad, and S. Ramadass, "Architecture for applying data mining and visualization on network flow for botnet traffic detection," *in Computer Technology and Development, 2009. ICCTD'09. International Conference on,* 2009, pp. 33-37.

[9] M. J. Chapple, T. E. Wright, and R. M. Winding, "Flow anomaly detection in firewalled networks," *in Securecomm and Workshops,* 2006, 2006, pp. 1-6.

[10] N. Muraleedharan, A. Parmar, and M. Kumar, "A flow based anomaly detection system using chi-square technique," *in Advance Computing Conference (IACC), 2010 IEEE 2nd International,* 2010, pp. 285-289.

[11] Z. Jadidi, V. Muthukkumarasamy, and E. Sithirasenan, K. Singh, "Flow-Based Anomaly Detection Using Semi-Supervised Learning", *in International Conference on Signal Processing and Communication Systems (ICSPCS)* 2015, in press.

[12] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, “Metaheuristic Algorithms Based Flow Anomaly Detector”, *In Communications (APCC), 2013 19th Asia-Pacific Conference on, IEEE,* 2013, pp. 723-728.

[13] Z. Jadidi, V. Muthukkumarasamy, and E. Sithirasenan, K. Singh, "Based Intrusion Detection Techniques," *The State of the Art in Intrusion Prevention and Detection,* p. 285, 2014.

[14] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan, Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm. in proc. *IEEE ICDCS Workshops on the 2nd International Workshop on Network Forensics, Security and Privacy(NFSP),* 2013, pp.76-81.

[15] M. Castellani and H. Rowlands, "Evolutionary artificial neural network design and training for wood veneer classification," *Engineering Applications of Artificial Intelligence,* vol. 22, pp. 732-741, 2009.

[16] X.-S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations,* vol. 29, pp. 464-483, 2012.

[17] R. Pasti and L. N. de Castro, "The Influence of Diversity in an Immune–Based Algorithm to Train MLP Networks," *in Artificial Immune Systems, ed: Springer,* 2007, pp. 71-82.

[18] P. Zhaoyu, L. Shengzhu, Z. Hong, and Z. Nan, "The application of the PSO based BP network in short-term load forecasting," *Physics Procedia,* vol. 24, pp. 626-632, 2012.

[19] M. A. Cavuslu, C. Karakuzu, and F. Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning," *Applied Soft Computing,* vol. 12, pp. 2707-2718, 2012.

[20] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” *Information Sciences,* vol. 179, pp. 2232-2248, 2009.

[21] S. T. Zargar, J. Joshi, and D. Tipper, "DiCoTraM: A distributed and coordinated DDoS flooding attack tailored traffic monitoring," *in Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on,* 2014, pp. 120-129.

[22] R. Lin, O. Li, Q. Li, and K. Dai, "Exploiting Adaptive Packet-Sampling Measurements for Multimedia Traffic Classification," *Journal of Communications,* vol. 9, 2014.

[23] J. M. Khalife, A. Hajjar, and J. D áz-Verdejo, "Performance of OpenDPI in identifying sampled network traffic," *Journal of Networks*, vol. 8, pp. 71-81, 2013.

[24] K. Bartos and M. Rehak, “Towards efficient flow sampling technique for anomaly detection,” *in Traffic Monitoring and Analysis, ed: Springer*, 2012, pp. 93-106.

[25] S. Mirjalili, S. Z. Mohd Hashim, and H. Moradian Sardroudi, “Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm,” *Applied Mathematics and Computation,* vol. 218, pp. 11125-11137, 2012.

[26] The CAIDA UCSD “DDoS Attack 2007” *Dataset http://www.caida.org/data/passive/ddos-20070804dataset:xml*

[27] The CAIDA UCSD Anonymized Internet Traces 2013 http://www.caida.org/data/passive/passive2013dataset:xml

[28] DARPA Dataset http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

[29] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, “An overview of IP flow-based intrusion detection,” *Communications Surveys Tutorials, IEEE,* vol. 12, pp. 343-356, 2010.

[30] Z. Li, Y. Gao, and Y. Chen, “HiFIND: A high-speed flow-level intrusion detection approach with DoS resiliency,” *Computer Networks*, vol. 54, pp. 1282-1299, 2010.

[31] Z. M. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemoto, and N. Kato, “Combating against attacks on encrypted protocols,” *in Communications, 2007. ICC’07. IEEE International Conference* on, 2007, pp. 1211-1216.

[32] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys (CSUR),* vol. 41, p.15, 2009.

[33] R. Alshammari and A. N. Zincir-Heywood, “Machine learning based encrypted traffic classification: identifying SSH and skype,” *in Computational Intelligence for Security and Defense Applications, 2009. CISDA2009. IEEE Symposium on,* 2009, pp.1-8.

[34] M. Sheikhan and Z. Jadidi, “Flow-based anomaly detection in high-speed links using modified GSA-optimized neural network,” *Neural Computing and Applications*, vol. 24, pp. 599-611, 2014.

[35] Q. A. Tran, F. Jiang, and J. Hu, “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,* 2012, pp. 201-208.

[36] J. Mai, A. Sridharan, C.-N. Chuah, H. Zang, and T. Ye, “Impact of packet sampling on portscan detection,” *Selected Areas in Communications, IEEE Journal on,* vol. 24, pp. 2285-2298, 2006.

[37] V. Carela-Espaol, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Sol-Pareta, “Analysis of the impact of sampling on NetFlow traffic classification,” *Computer Networks,* vol. 55, pp. 1083-1099, 2011.

[38] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, “Network anomaly detection and classification via opportunistic sampling,” *Network, IEEE,* vol. 23, pp. 6-12, 2009.

[39] N. Hohn and D. Veitch, “Inverting sampled traffic,” IEEE/ACM Transactions on Networking (TON), vol. 14, pp. 68-80, 2006.

[40] N. Duffield, C. Lund, and M. Thorup, “Properties and prediction of flow statistics from sampled packet streams,”

in Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, 2002, pp. 159-171.

[41] C. Estan and G. Varghese, “New directions in traffic measurement and accounting,” vol. 32: ACM, 2002.

[42] K. Bartos and M. Rehak, "IFS: Intelligent flow sampling for network security–an adaptive approach," *International Journal of Network Management,* 2015.

[43] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and K. Singh, "Performance of Flow-based Anomaly Detection in Sampled Traffic," *Journal of Networks,* 2015, in press.

[44] G. Androulidakis and S. Papavassiliou, “Improving network anomaly detection via Selective flow-based sampling,” *Communications, IET,* vol. 2, pp. 399-409, 2008.

[45] F. Raspall, “Efficient packet sampling for accurate traffic measurements,” *Computer Networks,* vol. 56, pp. 1667-1684, 2012.

[46] http://www.mindrot.org/projects/softflowd/

[47] http://www.mindrot.org/projects/flowd/

[48] R. Rajabioun, “Cuckoo optimization algorithm,” Applied Soft Computing, vol. 11, pp. 5508-5518, 2011.

[49] J. Han, M. Kamber, and J. Pei, “Data mining: concepts and techniques” *San Francisco, CA, itd: Morgan Kaufmann* 5 (2001).

[50] S. Fernandes, C. Kamienski, J. Kelner, D. Mariz, and D. Sadok, “A stratified traffic sampling methodology for seeing the big picture,” *Computer Networks,* vol. 52, pp. 2677-2689, 2008.

[51] A. Sperotto, R. Sadre, F. van Vliet, and A. Pras, “A labeled data set for flow-based intrusion detection,” *in IP Operations and Management, ed: Springer,* 2009, pp. 39-50.

[52] M. A. Maloof, “Machine learning and data mining for computer security,” *Springer New York,* 2006.

[53] C. Guo, Y.-J. Zhou, Y. Ping, S.-S. Luo, Y.-P. Lai, and Z.-K. Zhang, “Efficient intrusion detection using representative instances,” *computers security,* vol. 39, pp. 255-267, 2013.