Performance of Flow-based Anomaly Detection in Sampled Traffic

Zahra Jadidi, VallipuramMuthukkumarasamy, ElankayerSithirasenan, 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*—In recent years, flow-based anomaly detection has attracted considerable attention from many researchers and some methods have been proposed to improve its accuracy. However, only a few studies have considered anomaly detection with sampled flow traffic, which is widely used for the management of high-speed networks. This gap is addressed in this study. First, we optimize an artificial neural network (ANN)-based classifier to detect anomalies in flow traffic. The results show that although it has a high degree of accuracy, the classifier loses significant information in the process of sampling. In this regard, we propose a sampling method to improve the performance of flow-based anomaly detection in sampled traffic. While existing sampling methods for anomaly detection preserve only small malicious flows, the proposed algorithm samples both small and large malicious flows. Therefore, the detection rate of the flow-based anomaly detector is improved by about 5% using our algorithm. To evaluate the proposed sampling method, three flow-based datasets are generated in this study.**

***Index Terms*—Flow-Based Anomaly Detection; Artificial Neural Networks; Gravitational Search Algorithm; Sampling**

I. INTRODUCTION

Rapidly growing networks require scalable methods to analyse the high volume of traffic. Flow-based analysis based on packet headers has been introduced to manage traffic in high-speed networks. In the recent past, a number of studies have been carried out using flow traffic to detect anomalies in high-speed networks. However, the increasing number of flows in modern networks is a problem for network administrators. Various flow-based network management techniques use sampling to manage the high volume of flow traffic. Sampling methods negatively affect the accuracy of anomaly detectors. This study investigates the impact of sampling on flow-based anomaly detection.

A flow is defined as a group of packets having a number of common features such as source IP address, source port, destination IP address, destination port, and protocol [1, 2]. Flow-based anomaly detection is an efficient method for volume attacks generating a large number of flows in a short period. Denial of service (DoS) attacks, distributed DoS (DDoS) attacks, worms, scans and botnets are examples of volume anomalies [3]. In addition, flow-based anomaly detection decreases privacy concerns in comparison with packet-based methods due to the absence of payload [1, 4].

Different methods have been used in developing flow-based anomaly detection systems such as support vector machine (SVM) [3], hidden Markov model [5], self-organising map (SOM) neural network [6], modified random-mutation hill-climbing and C4.5 (MRMHC-C4.5) algorithm [7], frequent pattern mining algorithm [8], data mining and visualization [9], statistical techniques [10], chi-square technique [11], semi-supervised methods [12], and artificial neural networks (ANNs), e.g. multilayer perceptron (MLP) [13].

Heuristic algorithms are extensively used for the optimization of structure and weights of ANNs. Genetic algorithm (GA) [14], simulated annealing [15], immune algorithm [16], particle swarm optimization (PSO) algorithm [17, 18], and gravitational search algorithm (GSA) [19] are a number of heuristic algorithms. GSA [19] is a swarm based heuristic 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 [19]. GSA is used in various flow-based anomaly detection systems [13, 20, 21] to analyse the high volume of flow records. An algorithm based on GSA and PSO (PSOGSA) [22] is used to improve the classification of a flow-based dataset [20]. In addition, a prototype classifier based on GSA is developed in [23] to classify instances in multi-class datasets. The paper compares the result with other algorithms like PSO and artificial bee colony (ABC). The results show the effectiveness of a GSA-based classifier in resolving classification problems. GSA is used in this study to improve the accuracy of an MLP classifier.

An overview of flow-based intrusion detection [1] shows the limitations of this method. The large number of flows in the current networks leads network administrators to widely use sampling methods to adapt link traffic to the memory budget and decrease CPU usage. Flow generators such as NetFlow, which is proprietary to Cisco, use sampling methods to reduce the required resources [1, 24].

Manuscript received July 5, 2015; revised September 20, 2015.

doi:10.4304/jnw.10.9.512-520

Traditional sampling methods are mostly designed for monitoring purposes [25]. As these methods change the traffic characteristics, they can decrease the accuracy of anomaly detection algorithms [26, 27]. The impact of sampling on anomaly detection has been investigated in a number of studies [25, 26, 28]. In our study we have made further enhancements to improve the performance of anomaly detection in sampled traffic.

There are two main groups of sampling methods: packet sampling and flow sampling [28, 29]. The packet sampling method, applied to packets before generating flows, is improved in [30], which proposes an adaptive packet sampling method to provide accurate measurements of network traffic. The use of packet sampling methods distorts the distribution of flow features.

In terms of preserving the characteristics of flow traffic, it has been shown that flow sampling, applied to flows, is more efficient than packet-based sampling [29, 31]. However, the required memory and CPU power is greater [26, 28]. In this regard, some flow sampling methods such as smart sampling [32] and sample-and-hold [33] have been introduced to reduce the required memory.

Flow size is the number of packets in a flow. In flow traffic, a large number of flows (including malicious and benign flows) have a small size; however, a small number are large-sized. Small flows carry relatively few packets, but large flows carry the majority of packets.

Traditional sampling methods are mainly designed to handle large flows; however malicious flows are mostly small in size. Therefore traditional methods fail to identify the small malicious flows [34]. The negative impact of sampling methods on machine learning classifiers is shown in [35] which proposes a solution to improve the results. The impact of four sampling methods, random packet sampling, random flow sampling, smart sampling and sample-and-hold sampling, on the performance of a wavelet-based, volume anomaly detection method and two port scan detection algorithms are investigated in [25]. The results clearly show the destructive impact of all of these sampling methods on the detection rate of the anomaly detection method.

The comparison of a number of sampling methods [26] shows that sampling methods often aim to provide quality traffic monitoring and do not preserve the traffic features required for anomaly detection. This is addressed in [36] which proposes selective sampling. Selective sampling [36] targets small sizes to improve anomaly detection. However, it is suitable only for small malicious flows and hence, large anomalies are lost. Smart sampling [32], on the other hand, is a probabilistic method biased for large flows and it gives low probability to small flows which are mostly the source of attacks [25, 34].

The impact of selective sampling and smart sampling on anomaly detection is studied in [34] which considers the entropy changes during a number of attacks in sampled traffic. According to the results, selective sampling is a suitable choice for small-sized anomalies, but large-sized anomalies are better sampled by smart sampling.

Intelligent flow sampling [28] addresses the negative impact of sampling on traffic analysis with two-stage flow sampling. The first stage extracts 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.

Our study investigates the impact of flow sampling in ANN-based anomaly detection and proposes a sampling technique that is capable of capturing both small and large malicious flows.Our proposed flow sampling technique (FST) consists of two stages (see Fig. 1). 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 (OSS), is proposed for the second stage of FST to sample malicious flows. The components of our proposed technique are shown in Fig. 1.

The detection rate in ANN-based anomaly detection of sampled traffic is improved in our study. Our contribution is as follows:

Benchmark data - flow-based datasets are essential for evaluating the flow-based anomaly detection systems. CAIDA DDoS datasets [37], CAIDA Traces 2013 [38], and DARPA datasets are packet-based datasets used in our study to generate flow-based datasets.

GSA algorithm - it is used to optimize the interconnection weights of a two-layer MLP neural network.This optimized classifier is trained with our generated flow-based datasets to distinguish between benign and malicious flow traffic (Fig. 1).

FST - flow-based anomaly detection using sampled traffic is an open issue which is improved in this study by proposing FST (Fig. 1). The first step of FST extracts the distribution of flow sizes, and then the results are passed to the proposed sampling method, OSS.

The proposed OSS method is used to sample both small and large malicious flows. In this method, a threshold is defined based on the total number of packets received by each size. Flow sizes, small or large, which receive packets more than the threshold, are sampled with greater priority. OSS improves the performance of our anomaly detector in sampled traffic.

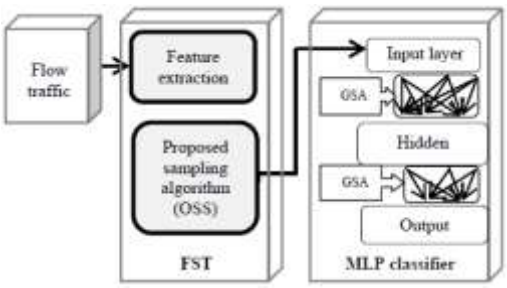

Figure 1. Components of the study

The remainder of this paper is organized as follows. Section II explains the methodology. Section III provides a brief description of the datasets used. Section IV provides the experimental results of the ANN-based flow

anomaly detection system. Section V describes the method used in this study to deal with sampled traffic and section VI concludes the paper.

II. METHOLOGY

This study investigates the impact of flow sampling on ANN-based anomaly detection. An accurate ANN-based classifier is required to be investigated with different sampling algorithms. We use an MLP neural network to detect anomalies in flow traffic. To improve the performance, the interconnection weights of this classifier are optimized using the GSA algorithm (see Fig. 1) [13]. The performance of the optimized MLP is evaluated in flow sampling methods, smart sampling and selective sampling.

A. Gravitational Search Algorithm

According to Newton's law, each particle in the universe attracts other particles. This is the basis of the GSA algorithm [19]. In GSA, agents are considered as objects and their masses are used for measuring their performance. All objects move towards the objects with heavier masses. Each mass (agent) in GSA also has a position corresponding to the solution of a problem.Heavy masses move more slowly and are known as good solutions. There are three kinds of masses [19]. M_a is an active gravitational mass which shows the strength of the gravitational field because of a particular object. M_p is a passive gravitational mass which is related to the strength of an object's interaction with the gravitational field. Inertial mass or M_i, is related to the resistance of an object to change its state of motion when a force is applied. The agents with bigger inertial masses have slower motion in the search space and a more accurate search while the bigger gravitational mass has faster convergence due to having a higher attraction. A fitness function is used to determine the gravitational and inertial masses [19]. In (1), the position of i^{th} agent in the d^{th} dimension is x_i^d .

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n) \qquad for \qquad i = 1, 2, ..., N \quad (1)$$

The force on mass *i* from mass *j* at time *t*is $F_{ij}^d(t)$ and is defined as in (2). M_{aj} is the active gravitational mass of agent *j* and M_{pi} is the passive gravitational mass of agent *i*. Gravitational constant at time *t* is G(t). ε is a small constant and $R_{ij}(t)$ shows the Euclidian distance between agents *i* and *j*. $R_{ij}(t)$ is defined as in (3).

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \quad (2)$$

$$R_{ij}(t) = \| X_i(t), X_j(t) \|_2 \quad (3)$$

The total force on agent *i*is as in (4), where $rand_j$ shows a random number in the interval [0, 1]. The acceleration of the agent *i* in direction $d^{th}$is defined as in (5). The next velocity and position of an agent are defined as in (6) and (7).

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \quad (4)$$

$$a_i^d(t) = \frac{F_i^a(t)}{M_{ii}(t)} \quad (5)$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (6)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (7)$$

G, the gravitational constant, is G_0 at the beginning and will be decreased with time. In GSA, *G* is defined as in (8), where α is a constant and *T* is the total number of iterations and *t* is the current iteration.

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \quad (8)$$

The inertial mass is defined as in (9) while $fit_i(t)$ is the fitness value of agent *i*. The *best(t)* and *worst(t)* for minimization problems are defined by (10) and (11) and for maximization they are defined by (12) and (13):

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (9)$$

$$best(t) = \min_{j\in\{1,...,N\}} fit_j(t) \quad (10)$$

$$worst(t) = \max_{j\in\{1,...,N\}} fit_j(t) \quad (11)$$

$$best(t) = \max_{j\in\{1,...,N\}} fit_j(t) \quad (12)$$

$$worst(t) = \min_{j\in\{1,...,N\}} fit_j(t) \quad (13)$$

In GSA, only *Kbest* agents attract other agents. *Kbest* has the initial value of K_0 but it is reduced with time. At first, all agents apply the force and finally only one agent remains applying force to the others so we will have (14) instead of (4).

$$F_i^d(t) = \sum_{j\in kbest j\neq i} rand_j F_{ij}^d(t) \quad (14)$$

We can summarize the function of GSA as follows [19]: 1) Initialize population. 2) Fitness evaluation for each agent. 3) Update *G(t)*, best(t) and worst(t). 4) Calculate M and acceleration for agents. 5) Calculate the new velocity and position. 6) If it meets end of criterion, it is the best solution, else go to step 2.

The comparison between GSA and PSO shows that GSA has merit in the field of optimization [19].

B. GSA-based Flow Anomaly Detection System

The classification of nonlinearly separable patterns can be performed using an MLP with two layers. An MLP is commonly trained with back-propagation (BP) algorithms. In BP algorithms, sometimes the convergence is to the points that are the best solutions locally (local minima) not globally (global minimum). To avoid local minima, heuristic optimization algorithms have been proposed.

GSA is a heuristic optimization algorithm which normally works well in searching for the global minimum. In our proposed anomaly detection system, we first employ a GSA algorithm to optimize the interconnection weights of a two-layer MLP (see Fig. 1). Then, the optimized MLP is deployed to detect anomalies in flow-based traffic. 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.

The GSA algorithm generates an initial population of masses corresponding to the weight coefficients of the MLP. The movement of masses shows an update in the weight coefficients to decrease mean square error (MSE). In each step, the positions of all masses are updated based on the calculated velocity in that step. These new positions correspond to the new weights. These weights are used to calculate MSE. Training is finished whenever it achieves the maximum number of iterations or an acceptable error.

The parameters of the GSA algorithm are shown in Table 1. We manually tried different numbers to findthese optimum parameters which cause the highest accuracy. The generated flow-based datasets are used to train our GSA-based MLP. We implement our system in MATLAB version R2012a (7.14.0.739). The optimized MLP cannot be trained with the dataset in its original form, therefore preprocessing is required. The dataset should be scaled to the range [-1; +1] to achieve optimal classification results. We use the Min-Max normalization method performing as given in (15) [13]. x_{max} andx_{min} are the maximum and minimum values of each feature. The data is rescaled to the range of values (t_{max} , t_{min}).

$$x' = (t_{\max} - t_{\min}) \times \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} + t_{\min} \qquad (15)$$

C. Flow Sampling Methods

The size of a flow is the number of packets carried by the flow. The distribution of flows is heavy-tailed for flow size. The majority of flows are small-sized while few of them are large [35].

Although there are few large flows, they carry a large number of packets. Therefore, traditional sampling methods are biased toward large flows, as they are important for efficient bandwidth monitoring. These methods cannot sample small flows which are the source of most attacks. A number of sampling methods could improve anomaly detection by focusing on small flows, but they lose the small population of large malicious flows carrying a great number of malicious packets [34].

TABLE I. GSA PARAMETERS

Description	Parameters	Value
Number of masses	M	5
Initial value of gravitational constant	G_0	100
Alpha	α	20
Total number of iteration	T	600

This section describes two important probabilistic flow sampling methods, smart sampling and selective sampling.

Smart sampling: It allocates a probability to each flow, based on the flow size [32, 34]. The probability is defined as in (16), where *x* shows the flow size in packet number and *t* is a threshold. In the smart sampling method, the probability of flow sizes larger than the threshold is 1. However, the probability under the threshold is proportional to flow sizes.

$$p(x) = \begin{cases} x/t & x < t \\ 1 & x \geq t \end{cases} \qquad (16)$$

Smart sampling cannot sample small malicious flows effectively due to its focus on large sizes.

Selective sampling: It aims to sample small flows with greater priority [36]. Each flow is sampled with probability *p(x)* as shown in (17), where *x* is the flow size, *t* is a threshold, $0 < c \leq 1$ and $n \geq 1$. The probability of flow sizes smaller than the threshold is a constant number, *c*. However, the probability for sizes larger than the threshold is inversely proportional to the size [34]. Selective sampling adjusts the number of samples using parameter *c*.

$$p(x) = \begin{cases} c & x \leq t \\ t/(n \cdot x) & x > t \end{cases} \qquad (17)$$

III. FLOW-BASED DATASETS

While bench mark datasets are essentialfor the evaluation of flow-based anomaly detection systems, there are not sufficient public datasets which can be used. The first labelled flow-based dataset was captured in Twente University network, by monitoring a honeypot [39]. A 10-Gbps optical Internet connection was monitored for this dataset. However, this dataset contained only malicious flows and it was very large. Therefore, it was modified [3] and a new dataset called Winter's dataset was generated. In spite of real networks, the number of benign flows in Winter's dataset was very small compared with malicious flows. As this dataset over-represented the number of malicious flows, this may have affected the evaluation of anomaly detection systems. This problem was addressed in another flow-based dataset extracted from packet-based DARPA dataset [40]. This dataset, however, contained only flow records with a specific destination IP address. Therefore, in this study we generate a number of flow-based datasets to provide a more comprehensive evaluation of flow-based anomaly detection.

NetFlow is responsible for generating flow records in Cisco routers. A NetFlow consists of two components: NetFlow exporter and NetFlow collector (see Fig. 2). The exporter creates flow records using the headers of incoming packets. The collector stores these flow records and sends them to NetFlowanalysers for further analysis.

In this study, NetFlow is simulated to generate the required flow-based datasets. NetFlow components, NetFlow exporter and NetFlow collector, are simulated

using Softflowd and Flowd [40]. Softflowd [41] receives packets and generates flow records. These flows are then sent to the NetFlow collector, Flowd [42].

Packet-based datasets, CAIDA DDoS datasets, CAIDA Traces 2013 and DARPA datasets [37, 38], are used in this study to generate a number of flow-based datasets.The CAIDA DDoS dataset contains one hour of DDoS attacks which occurred in 2007. This dataset contains special types of DDoS attacks which use all the bandwidth of the network from the Internet to a server. These DDoS attacks stop access to that server.

The CAIDA Traces 2013 dataset is captured by monitoring high-speed Internet backbone links. The traces are in two groups, CAIDA's Equinox-Chicago and Equinox-Sanjose. These traces are publicly available for research on Internet traffic and security systems.

The existing flow-based DARPA dataset [40] focuses only on flows sent to a specific destination address. Therefore, a flow-based DARPA dataset including all destination hosts is generated in this study. Table II shows detailed information about the number of generated flows in each dataset. Each flow record has eight features in the generated datasets: a) source IP address, b) source port, c) destination IP address, d) destination port, e) packets, b) octets, f) TCP flags, and g) IP protocol.

IV. Experimental Results and Analysis

For measuring the performance of the anomaly detector, four metrics are used [13]. Recall, Error Rate (ER), Miss Rate (MR) and False Alarm Rate (FAR) are defined as in (18) to (21) [13]. True Positive (tp) and True Negative (tn) show correct detection of malicious and benign traffic respectively. False Positive (fp) corresponds to the incorrect detection of benign traffic and False Negative (fn) is the error in the detection of malicious traffic [3].

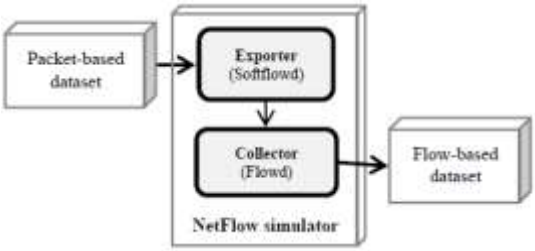

Figure 2. Simulation of NetFlow

TABLE II. GENERATED FLOW-BASED DATASETS

Dataset	Total flows	Malicious flows
Flow CAIDA DDoS	32638466	32638466
Flow CAIDA Traces 2013	4273773	……
Flow DARPA datasets	1375300	406284

$$Recall = \frac{t_n}{t_n + f_n} \quad (18)$$

$$ER = \frac{f_n + f_p}{t_p + t_n + f_p + f_n} \quad (19)$$

$$MR = \frac{f_n}{t_p + f_n} \quad (20)$$

$$FAR = \frac{f_p}{t_n + f_p} \quad (21)$$

The large number of flows in Table II needs a long training phase. Therefore, initially, two smaller datasets are randomly selected from the CAIDA and DARPA datasets respectively. Each of these datasets has 200,000 flows including benign and malicious flows. The ratio of malicious to benign flows in both selected datasets is the same, 1:3, and is equal to that of the original flow-based DARPA dataset in Table II. The selected datasets are shown in Table III.

TABLE III. SELECTED FLOW-BASED DATASETS

	Flow-based DARPA		Flow-based CAIDA	
	Malicious	Benign	CAIDA DDoS	CAIDA Traces2013
Randomly selected	45,610	154,390	45,610	154,390
Total flows	200,000		200,000	
Sampling Rate	20,000		20,000	

TABLE IV. PERFORMANCE MEASURES OF GSA-BASED ANOMALY DETECTOR COMPARED TO OTHER ALGORITHMS

Flow-based Dataset	Detector	Sampling	Recall (%)	ER (%)	MR (%)	FAR (%)
DARPA	GSA-based MLP	···	98.56	2.44	4.43	1.64
	PSO-based MLP[20]	...	97.41	3.28	6.31	2.03
	EBP-based MLP	…	93.86	4.51	8.12	2.01
	BBNN[40]	...	99.92	3.18	...	5.14
	SVM (RBF)[40]	...	92.07	6.56	…	5.20
	SVM(sigmoid)[40]	...	99.73	3.54	…	5.59
	Na ïve Bayes [40]	...	46.83	23.02		2.49
CAIDA	GSA-based MLP	···	96.20	1.83	2.17	1.51
	PSO-based MLP	···	94.71	2.15	2.45	1.63
	EBP-based MLP	···	93.14	3.14	4.11	2.20

TABLE V. PERFORMANCE OF GSA-BASED ANOMALY DETECTOR WITH DIFFERENT SAMPLING METHODS

Flow-based dataset	Detector	Sampling	Recall (%)
DARPA	GSA-based MLP	Original traffic	98.56
		Selective sampling	24.5
		Smart sampling	10.11
		Random sampling	17.35
CAIDA	GSA-based MLP	Original traffic	96.20
		Selective	22.35
		Smart	8.19
		Random sampling	20.12
Trace flows	Volume anomaly detection	Smart sampling [25]	18
		Random sampling [25]	6

First, the GSA-based anomaly detector is trained and tested with selected flow-based DARPA datasets and flow-based CAIDA with no sampling applied. GSA is compared with a number of heuristic algorithms in the optimisation of MLP weights. The performance of GSA and these training algorithms in tuning MLP weights are investigated and the experiments are repeated 10 times. Table VI shows the averaged results. The iteration

number is 600 for all of the methods. Limiting FAR in anomaly detection is a priority [3, 13]. According to TABLE IV, GSA has the lowest FAR. Additionally, GSA creates the highest recall compared to other methods. Therefore, GSA-based MLP is selected to be evaluated in sampled traffic.

In the second step, GSA-based MLP will be evaluated with three flow sampling methods: random flow sampling, smart sampling and selective sampling (see Table V). In the sampled traffic, the false negative is f_{ns} and it is as in (22), where u_m is the number of unsampled malicious flows. Therefore, f_n in (18) to (21) will be calculated using (22) in sampled traffic.

$$f_{ns} = f_n + u_m \tag{22}$$

In NetFlow, the sampling rate is static and is based on the worst situation [1]. A fixed sampling rate, 0.1, is selected in this study (see Table III). Table V compares the impact of different sampling methods on the recall of the GSA-based anomaly detector. The results are also compared with another study [25]. Due to a 0.1 sampling rate, 90% of the traffic will be lost. As Table V shows, in sampled traffic there is a significant drop in the recall of the anomaly detector. While selective sampling provides the best recall, large malicious flows are detected by smart sampling [34].

V. DEALING WITH SAMPLING TRAFFIC

The distribution of malicious flows in flow-based CAIDA DDoS and DARPA datasets is shown in Fig. 3. As shown in this figure, a large number of malicious flows are small in size and a small number of them are large in size. In spite of the small number of large flows, they carry significant numbers of packets. Selective sampling only focuses on small malicious flows, while smart sampling is suitable for large malicious flows [34]. An optimized version of selective sampling is proposed in this study to cover both types of malicious flows.

A. Proposed Sampling Method

Small malicious flows carry a large number of packets owing to their frequency. In addition, large malicious flows have a lot of packets due to their size. Fig. 4 shows the number of packets carried by different flow sizes in DARPA and CAIDA DDoS datasets. This figure clearly shows that the majority of packets are carried by a small number of flows. Therefore, flow sizes can be sampled with greater priority if their carried packets are larger than a particular threshold.

The proposed FST in Fig. 1 initially extracts information about each flow size based on the total number of packets carried by flows with that size. Then, it uses the proposed OSS method to sample, based on the distribution of flow sizes (see Fig. 1). The proposed OSS method, which can sample both small and large malicious flows, is the optimized version of selective sampling.

OSS is a probabilistic method in which each flow is sampled with the probability *p(x)* as shown in (23), where *x* is the flow size in packet number, *y(x)* is the number of packets received by flow size *x*, *t* is the threshold, and $0 < b \le 1, m \ge 1$.

$$p(x) = \begin{cases} y(x)/mt & y(x) < t \\ b & y(x) \ge t \end{cases} \tag{23}$$

Each flow size which carries packets more than the threshold is sampled with greater priority and it is sampled with a constant number, *b*. Our proposed method can sample small and large malicious flows which carry a large number of packets.

This study chooses a static sampling rate which is adjusted by the threshold. In the following sections, the performance of the OSS method is compared with those of other sampling methods with the same sampling rate. The OSS method improves the performance of flow-based anomaly detection. It also preserves more malicious packets compared with other methods.

B. Results

Selective sampling focuses on small flows which are the possible source of a large number of anomalies. According to Table V, selective sampling gives the highest rate of detected malicious flows (recall). However, some anomalies generate large flows carrying a large number of packets. Fig. 3 shows that a number of malicious flows have large size. These anomalies are not sampled in the selective sampling method due to their size.

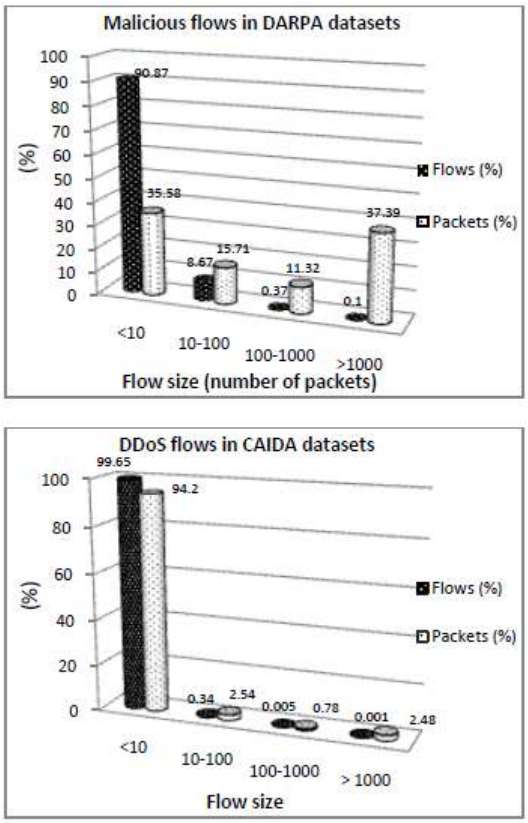

Figure 3. The distribution of malicious flows in DARPA and CAIDA DDoS dataset

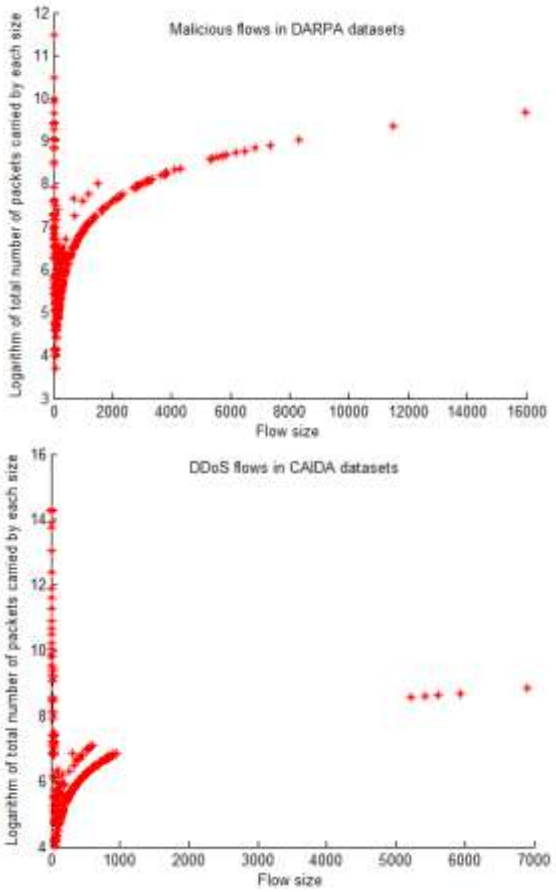

Figure 4. Total number of packets carried by each flow size in DARPA and CAIDA DDoS datasets

Smart sampling is an appropriate method for large anomalies. Smart sampling has the minimum rate of recall in Table V, as there are a small number of large malicious flows in flow traffic. The proposed OSS method optimizes selective sampling to sample large flows. Therefore, it has the advantages of both selective and smart sampling methods.

In terms of recall, the results of OSS are compared with other sampling methods (see Fig. 5). Fig. 5 compares the percentage of recall by flows, i.e. the rate of detected malicious flows, in different sampling methods [35]. It is shown that OSS has the highest rate of recall by flows. The recall of OSS is also better than the recall of another study in Table V.

The OSS method helps to detect most malicious flows and its recall by flow is better than that of selective sampling. Smart sampling has the lowest recall by flows; therefore, it loses a lot of attacks.

To improve anomaly detection, a sampling method should be able to preserve as many malicious packets as possible. Recall by packets shows the number of malicious packets carried by detected flows. Fig. 5 shows that the OSS method has a high recall by packets.

Smart sampling samples all large malicious flows with probability 1; therefore, it can preserve a lot of malicious packets. According to Fig. 5, the results from OSS and smart sampling are very close in terms of recall by packets. Our proposed sampling method has merit in sampling both small and large flows. Thus, its recall by packets is more than that of selective sampling. The high performance of the OSS method is shown in Fig. 5.

OSS gives a high priority to a flow size, when the number of carried packets by that flow size is larger than the threshold (Fig. 4). The OSS method can sample large flows, which are mostly ignored in selective sampling. Monitoring tools usually sample large flows; therefore, OSS is a good option for monitoring purposes. In addition, OSS can sample both small and large malicious flows. Thus, it captures flows required for flow-based anomaly detection.

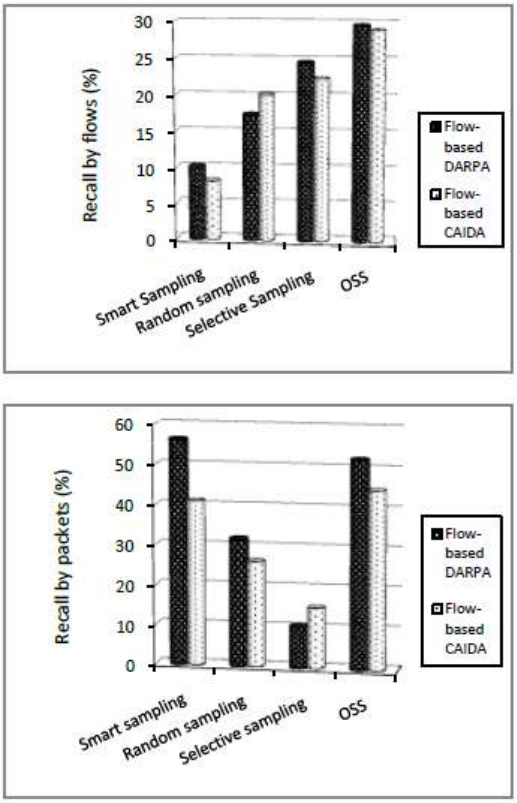

Figure 5. Comparison of recalls of the anomaly detector with different sampling methods (sampling rate 0.1)

To provide a comprehensive evaluation, OSS is examined with different sampling rates and its performance is compared with other sampling methods (see Fig. 6). In all sampling rates, the recall by flows in OSS is the highest, and it is close to that of selective sampling. OSS can significantly improve the percentage of recall by packets compared to selective sampling and its result is close to that of smart sampling.

Fig. 6 also shows the impact of the sampling rate on the detection rate of malicious flows. An increase in the sampling rate helps OSS to save more malicious flows; therefore, the recall of the flow-based anomaly detector will be increased.

VI. CONCLUSION

Flow-based anomaly detection was introduced to handle the analysis of the high volume of traffic in high-speed networks. Recently, sampling methods have been

widely employed to decrease the large numbers of flows in modern networks, but they seriously destroy flow-based anomaly detection. In this study, an optimized MLP was developed to detect anomalies in sampled flow traffic. Then, we proposed an optimized selective sampling method in which flows were sampled based on the number of received packets. In contrast to traditional methods, our method could sample both small and large malicious flows. The proposed sampling method improved the detection rate of our MLP-based anomaly detection by about 5%. A number of flow-based datasets were generated to evaluate our anomaly detector and the proposed sampling method. The future aim of this study is to develop an adaptive sampling rate.

ACKNOWLEDGMENT

The authors would like to thank CAIDA for the two datasets used to generate flow-based CAIDA datasets. They are also grateful to UNSW@ADFA for providing the flow-based DARPA datasets.

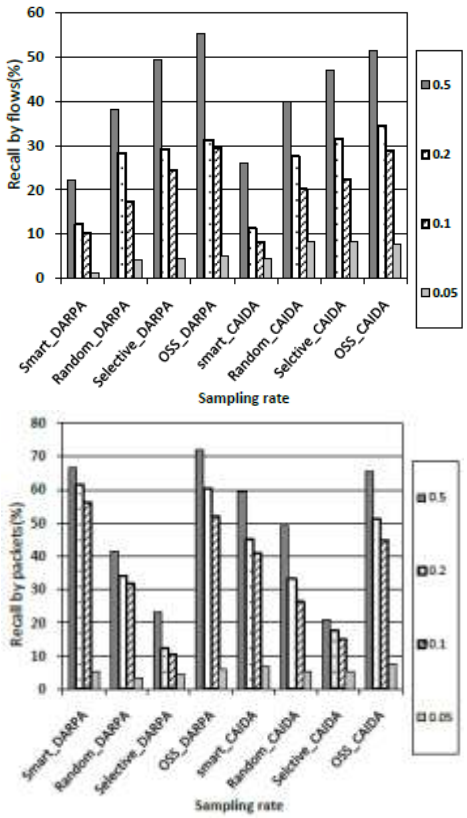

Figure 6. Sampling methods with different sampling rates

REFERENCES

[1] 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.

[2] 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

[3] 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.

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

[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] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *in Local Computer Networks (LCN), 2010 IEEE 35th Conference on, 2010*, pp. 408-415.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] S. Amato, B. Apolloni, G. Caporali, U. Madesani, and A. Zanaboni, "Simulated annealing approach in backpropagation," *Neurocomputing*, vol. 3, pp. 207-220, 1991.

[16] 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.

[17] 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.

[18] 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.

[19] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences.2009;* 179: 2232–2248.
[20] 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.
[21] 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.
[22] 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.
[23] A. Bahrololoum, H. Nezamabadi-Pour, H. Bahrololoum, and M. Saeed, "A prototype classifier based on gravitational search algorithm," *Applied Soft Computing,* vol. 12, pp. 819-825, 2012.
[24] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications,* vol. 36, pp. 567-581, 2013.
[25] 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.
[26] K. Bartos and M. Rehak, "Towards efficient flow sampling technique for anomaly detection," *in Traffic Monitoring and Analysis, ed: Springer,* 2012, pp. 93-106.
[27] 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.
[28] K. Bartos and M. Rehak, "IFS: Intelligent flow sampling for network security–an adaptive approach," *International Journal of Network Management,* 2015.
[29] 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.
[30] R. Lin, O. Li, Q. Li, and K. Dai, "Exploiting Adaptive Packet-Sampling Measurements for Multimedia Traffic Classification," *Journal of Communications,* vol. 9, 2014.
[31] J. M. Khalife, A. Hajjar, and J. Díaz-Verdejo, "Performance of OpenDPI in identifying sampled network traffic," *Journal of Networks,* vol. 8, pp. 71-81, 2013.
[32] 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.
[33] C. Estan and G. Varghese, "New directions in traffic measurement and accounting" vol. 32: ACM, 2002.
[34] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network anomaly detection and classification via opportunistic sampling," *Network, IEEE,* vol. 23, pp. 6-12, 2009.
[35] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, "Analysis of the impact of sampling on NetFlow traffic classification," *Computer Networks,* vol. 55, pp. 1083-1099, 2011.
[36] G. Androulidakis and S. Papavassiliou, "Improving network anomaly detection via Selective flow-based sampling," *Communications, IET,* vol. 2, pp. 399-409, 2008.
[37] The CAIDA UCSD "DDoS Attack 2007" Dataset http://www.caida.org/data/passive/ddos200708nct04_dataset.xml, as of April 2014.
[38] The CAIDA UCSD Anonymized Internet Traces 2013 http://www.caida.org/data/passive/passive_2013_dataset.xml, as of April 2014.
[39] 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.
[40] 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.
[41] http://www.mindrot.org/projects/softflowd/,as of Jun 2014.
[42] http://www.mindrot.org/projects/flowd/, as of Jun 2014.