Flow-level Network Anomaly Detection Method Based on Graph Connectivity

Ayaz Isazadeh, Jaber Karimpour and Roya Rastgar*
Department of Computer Science, Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

***Abstract*—In this paper, we propose a network-based anomaly detection approach using flow technology and graph theory. The concepts of the graph have long been used in the field of computer security especially in the area of Intrusion Detection System (IDS). Graph-based anomaly detection methods rely on the inspection of individual packets and, hence, they are not capable of being deployed on high-speed networks. To address this issue, this paper presents a flow based detection scheme using Traffic Dispersion Graph (TDG) clustering.**

In this work, flows are simulated from the network traffic traces. Subsequently, the Traffic Dispersion Graph (TDG) is applied for modeling flows over time. In order to prepare the universal cluster-based dataset, we employ a genetic algorithm for clustering TDGs. Eventually, a generic detection metric is introduced for diagnosing a wide range of attacks. We evaluate the proposed approach on network traffic traces from MIT DARPA99. The experimental results demonstrate that we are able to achieve appropriate evaluation criteria in terms of high detection rate, low false alarm, and high accuracy.

***Index Terms*—Anomaly Detection, Computer Security, Graph Clustering, Network Flow**

I. Introduction

DUE to the continued growth of computer networks such as the Internet, in terms of size, speed, and complexity, security of networks and detecting anomalies on network traffic is considered as a critical issue. The main reason of anomalies on network is malicious attacks like DoS (Denial of Service), port scan, buffer overflow and dictionary attack.

Because these abnormal behaviors on network can cause serious damages, it is an urgent task to diagnose and detect them by highest accuracy and lowest false alarm. Intrusion detection schemes are typically based on machine learning, data mining, and statistical analysis of network behaviors. These methods usually generate a large number of false alarms, and as a result, further work is essential requirement to improve detection performance.

As mentioned above, there are several methods to detect anomalies. One of the effective ways to achieve proper criteria of detection performance is applying graph theory in intrusion detection techniques [1]. Mainly, graphs provide a wide image of network, as well as draw patterns for normal and abnormal behaviors. Recently, a particular graph, which is called Traffic Dispersion Graph (TDG), has been identified to generally visualize the behavior of network.

Nowadays, using packet-based intrusion detection systems in high-speed networks is not efficient because analysis of packet payloads reduce the speed of systems. To encounter this issue, researchers have used flow instead of packet in detection schemes. In flow-based approaches, the set of packets that pass an exact point in the network and have common properties, which is considered as a flow, is inspected.

In this paper, we focus on employing the notion of data flow and TDG graph to present a generic intrusion detection method. In respect of this case, we use TDG for analyzing anomalous behavior on network. Two principal contributions of our approach are:

1. Model network traffic by processing flows and clustering TDGs over time followed by generate a cluster-based dataset.
2. Identify a generic detection metric by organizing the extracted features from cluster-based dataset.
3. Eventually, we use MIT's DARPA dataset in order to evaluate the effectiveness of our method.

The rest of the paper is organized as follows: in the next section, we briefly review previous intrusion detection schemes. In section III, we explained partially data preprocessing approach, followed by a detailed description of introducing a novel metric to detect anomalies in section IV. We evaluate our proposed approach in section V. In addition, the experimental results are presented in section VI. Ultimately, we conclude the research conducted in this paper and suggest alternative work for future in section VII.

II. Related Work

There are several types of research work that concentrate on network anomaly detection [2]. The goal of the presented paper is to propose an intrusion detection method using flow-based, graph-based and packet header-based detection techniques. Therefore, we pursue the previous work which is related to these three categories of approaches.

A. Flow-based intrusion detection

Instead of inspecting individual packets on network such as [3], [4], [5] and [6], network traffic flow is applied in several categories of detection scheme [7]. Generally, flow is identified as a set of packets which have a group of common properties.

Recently, a structured approach to detect anomalies in flow-based time-series is introduced in [8]. In this work, analyzing

the flow-level characteristics of network traffic shows the importance and inevitability of flow-based technologies in designing network-based intrusion detection for high-speed networks.

Hellemons et al. [9] designed and implemented a flow-based intrusion detection system for SSH dictionary attack. In this system, three phases are identified by analyzing flow-level SSH traffic. Additionally, a particular algorithm is proposed for detecting each phase of the attack. Ultimately, to describe the behavior of the network, two metrics are used in detection algorithm: packets-per-flow (PPF) and a minimum number of flow records which is collected at one minute time intervals. Although this approach is efficient on high-speed networks, it is limited to detecting at most one type of attack which is called SSH dictionary attack. In comparison, our introduced metric is able to detect several varieties of attacks.

B. Graph-based intrusion detection

Graph-based intrusion detection system was introduced for the first time in [10]. In this system, the data of activities among computers on a network is collected and this information is drawn on activity graph which shows an activity on the network. Additionally, this approach was implemented in [11] which was efficient for worm detection.

Nowadays, network traffic monitoring has become one of the challenging problems in intrusion detection approaches. Due to monitoring, analyzing and attaining a correct view of network, Traffic Dispersion Graph (TDG) was proposed in [12]. This type of graph is identified as a graphical perspective of alternative interaction among nodes. Related to IP networks, one node belongs to an entity which has a specific IP address. It is noteworthy to mention that the communication between two nodes is represented by the edge of the graph. One of the significant advantages of TDG is the power of visualizing attacks' structures.

In this regard, a novel anomaly detection scheme was introduced in [13]. In this work, several concepts of graph theory such as node degree and graph matching is applied. Moreover, TDG graph is used for modeling network traffic. This approach is comprised of two part: the first part focuses on statistical properties of a graph and the second part considers the dynamic aspects of graphs in time-series. One of the drawbacks of this method is that it is limited to detecting only one type of attacks called Distributed Denial of Service (DDoS) attack. Accordingly, it would not be sufficient on large networks with the wide range of attacks. Therefore, in this paper, we successfully diagnose several categories of attacks by employ a clustering method on TDG graph.

C. Packet-based intrusion detection

One of the important packet-based intrusion detection methods was proposed in [14] which uses the learning of normal properties of each packets' header to recognize anomalies. In several anomaly detection approaches, data mining and machine learning algorithms are utilized in which complex feature vectors are considered as the input of algorithm. It is demanding to extract particular features from network traffic. Furthermore, tools for extracting features are not publicly available.

As outlined before, deploying intrusion detection system based on feature vector has become difficult for network managers. To address this issue, raw network traffic data was applied in [15]. In this approach, information in each packet's header is inspected. The main goal of this method is modeling normal data and detecting anomalies rely on deviation from normal patterns. There is various types of normal data which have different behavior. Therefore, analyzing the behavior of each normal patterns and diagnosing them from the anomalous data is an important aspect in this approach. Besides, to accurately model behaviors, clustering and Support Vector Machine (SVM) is employed. To sum up, since this method is based on analyzing packets, its performance is not efficient in high-speed networks. To deal with this problem, we use flow technology and attempt to cluster graph instead of cluster data.

III. NETWORK TRAFFIC MODELING

One of the important aspects of evaluation intrusion detection systems is testing them by an appropriate dataset. Due to the lack of proper dataset for validating our proposed scheme, we need to preprocess traffic traces. As illustrated in Fig. 1, the overall process consists of five parts which we will describe them in the following.

A. Packet headers extraction

Features of a packet header is required for generating flows. In this work, six features were extracted including source IP address, destination IP address, protocol, packet time, source port, and destination port.

B. Flow simulation

There are several definitions of flow data [16] and [17]. In this paper, we follow the definition that proposed by IPFIX (IP Flow Information Export) [18]:

"A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties."

Common properties are source and destination IP address, source and destination port number and IP protocol. Using this definition, we simulate flows in 1 second time interval as detailed below:

Step1: Classifying extracted headers into 1 second time interval groups.

Step2: Categorizing headers with common properties in determined time interval as a flow. Following features are considered for each flow:
(Source IP address, destination IP address, number of packets, time, source port number, destination port number and the protocol type)

C. Traffic Dispersion Graph construction

By considering the regular time points $t_1, t_2, \dots, t_n$, the graphs are created using flows in alternative time-series. To construct a proper graph, the concept of Traffic Dispersion Graph (TDG)

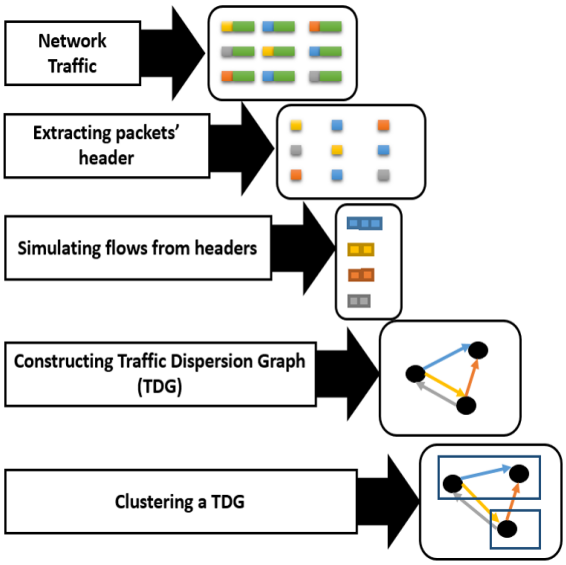

Fig. 1. Stages of preprocessing data

is required. TDG is a directed graph in which a node indicates an IP address and an edge represents the connection between two nodes (IP addresses). We assume the existence of flow or packet between two IPs as an edge in this paper.

As mentioned above, two types of TDGs are generated:

Flow-based TDG: A weighted edge is considered as the number of flows between two IPs.

Packet-based TDG: A weighted edge is considered as the number of packets between two IPs.

D. Time-series TDG clustering

There are several algorithms for graph clustering such as graph-based cluster algorithm (GB) [19] and [20]. However, this algorithm needs to have an initial amount as well as it works for undirected complete graphs. For this reason, we prefer to use a genetic algorithm for clustering TDG graphs.

We apply a graph clustering method based on genetic algorithm [21] that is suitable for our research work. Because of being applicable for weighted graphs, in the clustering algorithm, the weight of each edge is normalized between 0 and 1.

There are several steps in mentioned graph clustering algorithm that as following:

Step1: Creating an initial population. The encoding phase is based on generating random numbers, such that the length of each chromosome is equal to the number of graph nodes.

Step2: Calculating objective function which includes two phase: intra connectivity and extra connectivity.

- Calculate intra-connectivity

Measure the level of connection between nodes in a single cluster. High intra-connectivity represents that nodes in a cluster have high connection density which indicates a qualified clustering result.

This measurement is performed by formula as:

$$A_i = \frac{\mu_i}{N_i^2} \qquad (1)$$

In (1), μ_i and N_i represent the normalized weight of edges and the number of nodes in *i-th* cluster, respectively.

- Calculate inter-connectivity

Measure the relationship between clusters. In contrast to intra-connectivity, low inter-connectivity indicates the independence of the nodes of various clusters.

This measurement is performed by formula as:

$$E_{ij} = \begin{cases} 0 & \text{if } j = i \\ \dfrac{\varepsilon_{ij}}{2N_iN_j} & \text{if } j \neq i \end{cases} \qquad (2)$$

In (2), ε_{ij} represents the normalized weight of edges between *i-th* and *j-th* clusters and $2N_iN_j$ expresses the maximum weight of edges between two clusters.

Using (1) and (2), objective function is calculated as bellow in which *k* corresponds to the number of clusters:

$$MQ = \begin{cases} \dfrac{\sum_{i=1}^{k} A_i}{k} - \dfrac{\sum_{i,j=1}^{k} E_{ij}}{\frac{k(k-1)}{2}} & \forall k > 1 \\ A_i & k=1 \end{cases} \qquad (3)$$

Step3: The selection procedure is implemented by a proper function.

Step4: The crossover and mutation operators are employed.

Step5: The new population as well as the next generation are created by the replacement action.

At the end of the explained stages, following features are extracted:

1. **Time (T-series):**
   Regarding the time-series, which are indicated by t_1, t_2, … ,t_n, this attribute is represented using the numbers from 1 to n.
2. **Cluster ID (cl-num):**
   Clusters are identified by the numbers in each determined time.
3. **Inner nodes (node):**
   The number of nodes (IPs) in a cluster.
4. **Internal packets (in-wgt):**
   The volume of packets among the nodes in the cluster.
5. **External packets (ext-wgt):**
   The volume of packets among the clusters.
6. **Internal flows (in-flw):**
   The volume of flows among the nodes of clusters.
7. **External flows (ext-flw):**
   The volume of flows among the clusters.

According to above features, the cluster-based dataset is prepared. Fig. 2 demonstrate the five time-series of our proposed dataset which is related to DARPA dataset and the network traffic from the 4th week, Tuesday [23].

IV. Proposed Detection Metric

Statistical metrics are accounted as the essential part of behavioral-based intrusion detection schemes to diagnose anomalous behavior. Indeed, metrics are required for analyzing the different behavioral characteristics on the network.

As mentioned before, a TDG provides a general explicit visualization of the network. In addition, the performance of attacks would be accurately identified using the graph theory and clustering concept.

Fig. 3 represents an instance of clustered TDG in which an

T-series	cl-num	node	in-wgt	ext-wgt	in-flw	ext-flw
1	1	2	10	2	4	2
1	2	2	0	12	0	1
1	3	3	1	1	1	1
1	4	2	23	11	2	1
2	1	3	0	25	0	4
2	2	2	8	1	6	1
2	3	3	2	13	2	2
2	4	2	37	39	6	5
2	5	2	1	0	1	0
3	1	2	2	0	2	0
4	1	2	23	0	4	0
4	2	2	2	0	2	0
4	3	2	1	0	1	0
4	4	2	6	0	2	0
5	1	1	0	0	0	0
5	2	2	2	12	2	1
5	3	2	24	25	4	4
5	4	4	0	37	0	4
5	5	2	17	5	11	5
5	6	2	1	0	1	0
5	7	2	22	24	4	4

Fig. 2. 5 time-series of the cluster-based dataset

edge indicates a relationship between two IPs. Extensively, in this figure, the cluster with the maximum weight of internal edges, which is assumed as intra-connectivity of the cluster, points out the strong connectivity among IPs. The high intra-connectivity would be the indicator of several types of attacks behavior such as DoS, User to Root (U2R) and Remote to Local (R2L) [22]. Furthermore, a cluster with the maximum weight of external edges indicates the high association among the IPs of two separate clusters, which is considered as inter-connectivity of the cluster. The analysis of various attack strategies shows that the high inter-connectivity would be equivalent to the port scan attack behavior.

In respect of explained notion above, we use the features which are extracted from proposed cluster-based dataset and suggest a generic metric that is named Weight Ratio (WR). To attain Weight Ratio metric, calculating two parameters are required: Average Distribution of Internal Flows (ADIF) and Relative Wight of External Flows (RWEF).

Average Distribution of Internal Flows Indicates high intra-connectivity over a distinctive time. The parameter, $ADIF_i$, is computed for each *i-th* time in considered dataset by (4). In this regard, $maxflow_i$ points out the weight of internal flows which belongs to the cluster with maximum internal flow volume and N_i indicates the number of nodes in the same cluster.

$$ADIF_i = \frac{maxflow_i}{N_i} \tag{4}$$

Regarding the performance of various attacks such as DoS along with the conception of fig. 3, we expect that this

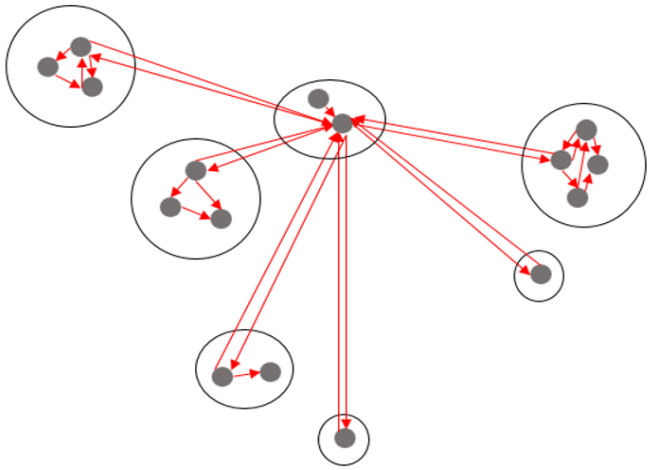

Fig. 3. A hypothetical clustered TDG

parameter would detect the distinctive types of attack by passing a specific threshold value.

Relative Wight of External Flows is introduced for demonstrating the inter-connectivity among clusters. A cluster with the maximum inter-connectivity would represent that a node in it connects to alternative nodes in the other clusters. In this case, the node which exists in the cluster with the high volume of external flows is considered as an intruder and the other nodes which associate with the intruder is assumed as the target nodes.

In order to describe the Relative Weight of External Flows (*RWEF*) in *i-th* time, we explain formula (6) as follows:

1. We refer F_j as the average distribution of external flows of *j-th* cluster. As shown in (5), in order to attain F_j, the weight of external flows EXF_j of *j-th* cluster is divided into the number of nodes in the same cluster N_j.

$$F_j = \frac{EXF_j}{N_j} \tag{5}$$

2. We select the maximum of F_j in *i-th* time as $MaxF_i$. Finally, $MaxF_i$ is divided into the summation of F_j for other clusters (except the cluster belongs to $MaxF_i$).

$$RWEF_i = \frac{MaxF_i}{\sum_{j=1}^{k} F_j - MaxF_i} \tag{6}$$

Respect to the behavior of port scan attacks and the concept of fig. 3, we anticipate that the external flows would exist among several clusters, as well as the summation of average distribution of external flows for other clusters (F_j) might be more than $MaxF_i$. In this instance, if the value of $RWEF_i$ is between 0 and 1, an attack would occur.

Weight Ratio metric in *i-th* time (WR_i), which employs the Average Distribution of Internal Flows ($ADIF_i$) and Relative Wight of External Flows ($RWEF_i$) for detecting the wide range of various attacks, is calculated as follows:

$$WR_i = \frac{AVGF_i}{AVGFE_i} \tag{7}$$

According to the analysis of the performance of anomalies and fig. 3 as well, we expect an attack would occur if a threshold level exceeds a specific value in terms of Weigh Ratio.

V. Evaluation

After the stages of data processing and calculating generic metric, a comprehensive model for anomaly detection, which is depicted in Fig. 4, is proposed. In this section, we evaluate the suggested model using publically available DARPA99 dataset [23]. The dataset includes five weeks in which three weeks are considered as training data and two weeks are introduced as testing data. In this paper, the testing data of third and fourth weeks are used.

It should be noted that DARPA dataset [23], and [24] contains five types of attacks: DoS, scan, Remote to Local (R2L), User to Root (U2R) and data [25]. Table 1 represents the number of each types of attacks in DARPA99.

TABLE I
THE NUMBER OF DIFFERENT TYPES OF ATTACKS IN DARPA99

Attack	DoS	Scan	R2L	U2R	Data
	17	8	17	14	1

We evaluate and compare the introduced detection scheme in terms of Detection Rate (DR), false alarm and accuracy. These performance criteria is computed using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) as follows:

$$DR = \frac{TP}{TP + FN} \tag{8}$$

$$FalseAlarm = \frac{FP}{TN + FP} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{10}$$

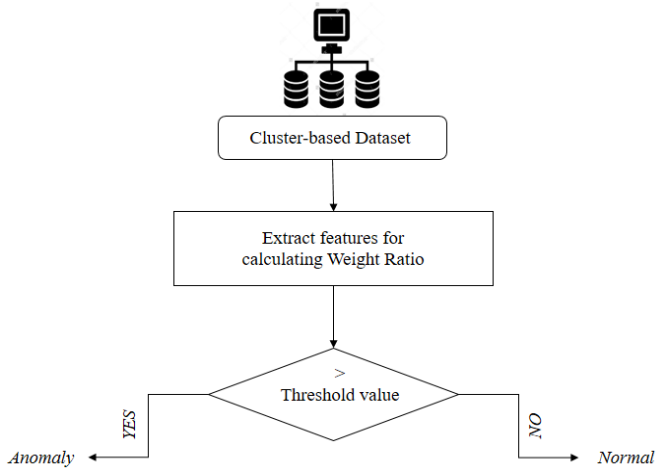

Fig. 4. Stages of preprocessing data

TABLE II
THE NUMBER OF NORMAL AND ANOMALOUS POINTS IN EACH TIME-INTERVAL

Time-interval	Normal	Anomalous	Total
30 sec	20206	2734	22940
60 sec	11721	1479	13200
90 sec	7382	1028	8410

We tend to illustrate the performance of proposed approach by reaching the maximum detection rate, minimum false alarm rate and, consequently maximum accuracy. To achieve this goal, our method is tested by several time-intervals and threshold values.

Due to showing how the time interval selection affects the detection performance, we examine three alternative time-intervals: 30 seconds, 60 seconds and 90 seconds. In table II, the number of normal and anomalous points of fourth and fifth weeks of dataset is represented for each determined time-interval.

A. Threshold and time-interval

To specify the threshold value, we study network traffic records of the fifth week from DARPA99. In this work, we use static threshold values, which are determined from the comparison of normal and anomalous traffic. In this case, the threshold is confirmed 3 times for each time intervals.

In this process, we sample the network traffic in terms of regular time points $t_1, t_2, \ldots, t_i, \ldots, t_n$ in which t_i represents the *i-th* time point over time-series. Ultimately, the detection model presented in figure \ref{model} is employed to diagnose normal and abnormal points of data. Continuously, the result of this part of the experiment for 30 seconds, 60 seconds and 90 seconds is shown as below respectively.

30 seconds time-interval:

Table III shows the experimental results on 30 seconds time-series. We set threshold value to 30, 60 and 50 Weight Ratio (WR). The observations represent that both 50 and 60 threshold values possess high accuracy. Although the 60 threshold value reaches the maximum accuracy as well as minimum false alarm, it does not achieve the highest detection rate. As a result, we opt 50 threshold value to pursue the rest of experiment.

TABLE III
COMPARISON OF DIFFERENT THRESHOLD VALUES IN 30 SECONDS TIME-INTERVAL

Threshold	Detection rate	False alarm	Accuracy
>30	0.996	0.093	0.905
>50	0.986	0.038	0.963
>60	0.926	0.031	0.965

Fig. 5 illustrates the histogram of Weight Ratio metric for 30 seconds time-series from DARPA99, 5th week, Tuesday. The time-series between 430 and 460 time points as well as 1220 and 1230 points demonstrate DoS attacks which hold more than 200 Weight Ratio. In addition, the time points between 1060 and 1064 represent port scan attacks with greater than 60

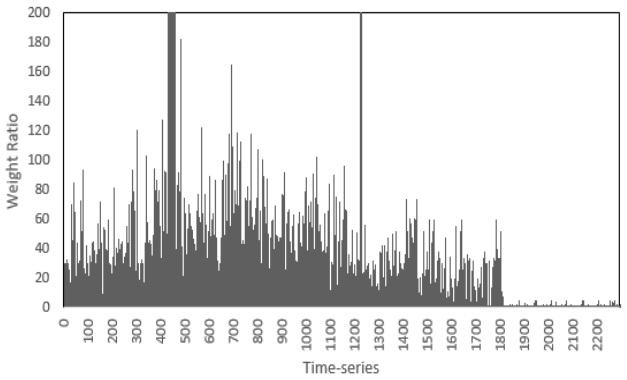

Fig. 5. Time-series of Weight Ratio in 30 seconds time-interval

Weight Ratio. According to 30 threshold value, we observe high false alarm.

As shown in table IV, the maximum Detection Rate (DR), which is calculated for 50 Weight Ratio in 30 seconds time-interval, is related to U2R and DoS attacks.

TABLE IV
DETECTION RATE OF EACH TYPES OF ATTACK IN 30 SECONDS TIME-INTERVAL AND 50 THRESHOLD VALUE

Attack	DoS	Scan	R2L	U2R	Data
DR	0.988	0.975	0.976	0.997	0.925

60 seconds time-interval:

We determine 3 threshold values for Weight Ratio over 60 seconds time-series. Relating to table V, 100 threshold value leads to the maximum accuracy; however, it has low detection rate. Following the observations, 80 threshold value is selected.

TABLE V
COMPARISON OF DIFFERENT THRESHOLD VALUES IN 60 SECONDS TIME-INTERVAL

Threshold	Detection rate	False alarm	Accuracy
>60	0.958	0.087	0.914
>80	0.933	0.060	0.939
>100	0.850	0.042	0.951

Fig. 6, represents the histogram of Weight Ratio for 60 seconds time-interval from DARPA99, 5th week, Tuesday. The time points between 213 and 219 as well as the points between 619 and 622 represent the DoS attacks. Moreover, port Scan attacks occur in 537 and 538 points. Relying on evidence, 80 threshold value seems appropriate for this time interval.

After selecting 80 Weight Ratio as a proper threshold value, we calculate the Detection Rate (DR) for each type of attack. Table VI represents that the maximum detection rate is related to U2R and DoS attacks.

TABLE VI
DETECTION RATE OF EACH TYPES OF ATTACK IN 60 SECONDS TIME-INTERVAL AND 80 THRESHOLD VALUE

Attack	DoS	Scan	R2L	U2R	Data
DR	0.968	0.819	0.794	0.997	0.950

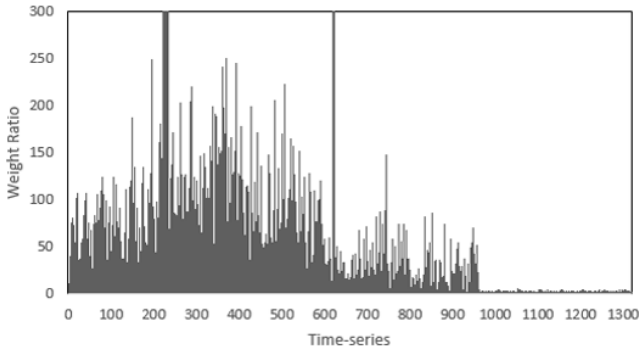

Fig. 6. Time-series of Weight Ratio in 60 seconds time-interval

90 seconds time-interval:

Similar to 30 and 60 seconds time-intervals, we experiment the proposed method on 90 seconds time-interval using 3 threshold values. As illustrated in table VII, 200 threshold value with minimum false alarm and minimum detection rate does not provide the best performance result. Likewise, 100 threshold value with maximum detection rate is affected by the maximum false alarm. Therefore, 150 threshold value is selected for this time-interval.

Fig. 7, represents the histogram of Weight Ratio for 90 seconds time-interval from DARPA99, 5th week, Tuesday. DoS attacks happen in time points between 145 and 154 and the points between 409 and 412. In addition, 355 and 356 points demonstrate the occurrence of port scan attacks. Based on the results in table \ref{tab:7} and the histogram in figure \ref{90s},

TABLE VII
COMPARISON OF DIFFERENT THRESHOLD VALUES IN 90 SECONDS TIME-INTERVAL

Threshold	Detection rate	False alarm	Accuracy
>100	0.971	0.082	0.927
>150	0.846	0.039	0.955
>200	0.714	0.016	0.969

TABLE VIII
DETECTION RATE OF EACH TYPES OF ATTACK IN 90 SECONDS TIME-INTERVAL AND 150 THRESHOLD VALUE

Attack	DoS	Scan	R2L	U2R	Data
DR	0.806	0.580	0.523	0.972	0.933

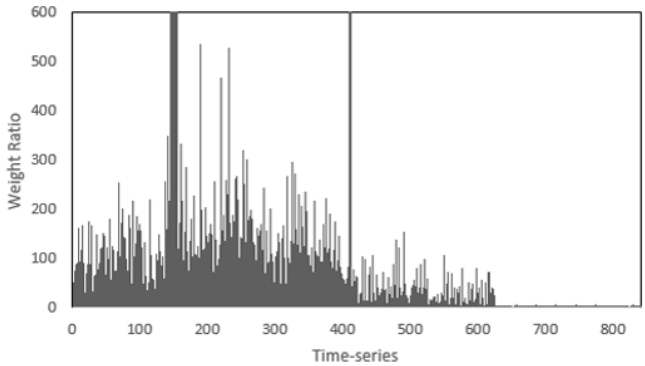

Fig. 7. Time-series of Weight Ratio in 90 seconds time-interval

we choose 150 as an appropriate threshold value in 90 seconds time-interval.

Using 150 threshold value in 90 seconds time-interval, we summarize the result of Detection Rate (DR) for each type of attack in table VIII. In consequence, the satisfied detection rate is obtained for U2R and Data attacks.

Comparison of 3 time-intervals:

As mentioned above, we selected the relevant threshold value for each time-interval. Therefore, table VIIII, analyze the performance of the time-intervals. The maximum detection rate and accuracy as well as the minimum false alarm of 30 seconds time-interval cause to opt this time-interval to continue the comparison process. In addition, the ROC in Fig. 8 shows the qualification of detection performance in 30 seconds time-interval.

VI. Experimental Results

TABLE VIIII
COMPARISON OF PERFORMANCE OF 3 TIME-INTERVALS

Time interval	Detection rate	False alarm	Accuracy
30 sec	0.986	0.038	0.963
60 sec	0.933	0.060	0.939
90 sec	0.846	0.039	0.955

In this part of our research work, we evaluate the proposed

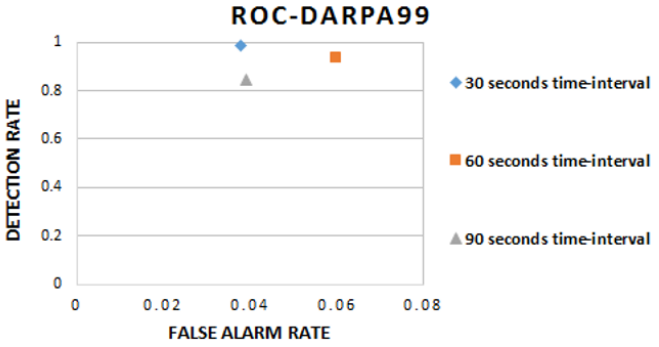

Fig. 8. ROC for 3 time-intervals

detection model to illustrate the power of it. In order to achieve this goal, we show the merits of using flows and graph clustering techniques by comparing the flow-level and packet-level results as well as comparing the performance of the method before and after clustering. In addition, we analyze the difference between proposed method and intrusion detection systems such as Snort [25] and Cisco [26] in terms of the number of detected attacks. Ultimately, the comparison of the results of our approach and the other work, which employ a packet-based technique, is represented.

According to the results of testing different time-intervals, the 30 seconds time-interval is used to evaluation and comparison.

A. Comparing flow-level and packet-level anomaly detection

As mentioned in section II, Hellemons et al. [9] introduced the number of Packets Per Flow (PPF) criterion to detect the SSH dictionary attack. In this paper, we want to prove that using the number of flows has better detection performance rather than using the number of PPF. Indeed, in this stage, the number of flows and the number of PPF are considered as the comparison metrics.

In this part of comparison, after testing different threshold values for each metric, 30 threshold value is selected for the number of flows metric and 200 threshold value is opted for the number of PPF.

Based on the determined time-interval (30 seconds) and threshold values, the results of the best case for each approach are shown in table X.

As stated by table X, lowest detection rate and accuracy, as well as highest false alarm rate in packet-level method based on PPF represents that applying flow based technique in our work has persuasive results. This comparison claims that our method

TABLE X
COMPARISON OF DETECTION PERFORMANCE BY THE NUMBER OF PPF AND THE WEIGHT OF FLOWS

Method	Detection rate	False alarm	Accuracy
Flow	0.485	0.089	0.887
PPF	0.454	0.119	0.847

is effective than the proposed method in [9].

B. Comparing anomaly detection before and after clustering

As pointed before, we propose a generic detection metric, Weight Ratio, which is generated after graph clustering process. In order to evaluate detection performance of both before and after clustering approaches, the maximum number of flows is used as the comparison metric. Therefore, the maximum number of flows is considered as a metric for anomaly detection before clustering, as well as the maximum number of internal flows of clusters is applied as a metric for diagnosing anomalous behavior after clustering.

Regarding the procedure of determining the threshold value, which is described in section V, we employ 30 threshold value for before clustering method and 50 threshold value for after clustering approach. As the reported results in table XI, the significant increase in detection rate after clustering shows the

TABLE XI
COMPARISON OF DETECTION PERFORMANCE BEFORE AND AFTER CLUSTERING

Method	Detection rate	False alarm	Accuracy
After	0.816	0.087	0.904
Before	0.485	0.089	0.887

effect of clustering in detection methods.

Due to proving the superiority of Weight Ratio metric, we compare it with the maximum weight of internal flows metric in after clustering approach. According to table XII, the Weight

Ratio metric has a slight growth in terms of detection rate and accuracy and a slight decline in terms of false alarm. These results demonstrate the necessity of using Weight Ratio metric in cluster-based detection method.

TABLE XII
COMPARISON OF DETECTION PERFORMANCE BY WEIGHT RATIO METRIC AND THE MAXIMUM WEIGHT OF INTERNAL FLOWS

Method	Detection rate	False alarm	Accuracy
Weight Ratio	0.986	0.038	0.963
Internal flows	0.816	0.087	0.904

Eventually, to show the evolution of our proposed approach, we provide a ROC which is depicted in Fig. 9.

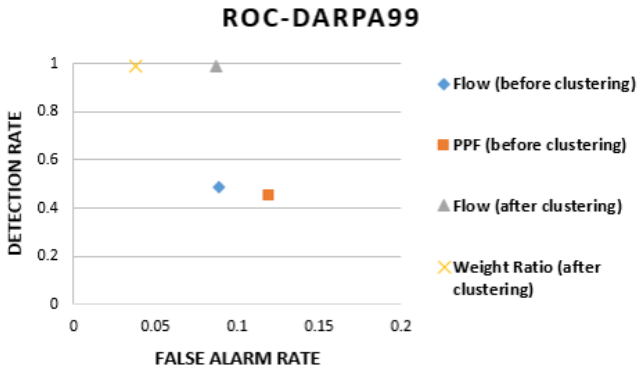

Fig. 9. ROC for evolution of our approach

C. Comparing proposed approach with commonly used IDSs

To show the reliability and practicality of our proposed approach, we compare it with two realistic intrusion detection systems, Snort [25] and Cisco [26]. In this phase of the experiment, the number of detected attacks, which are detected by 100 percent detection rate, is considered as evaluation criterion. The mentioned IDSs are evaluated by the number of detected attacks on DARPA99 dataset in [27].

The results in table XIII demonstrate that the total number of detected attacks by our approach is more than the number of attacks detected by Snort [25] and Cisco [26]. Overall, Snort can detect more attacks than proposed method just in terms of the number of detected R2L attacks. The deviation of behavior pattern of R2L by our method could be the reason of this case.

TABLE XIII
THE NUMBER OF DETECTED ATTACKS BY DIFFERENT IDSS IN DARPA99

Method	DoS	Scan	R2L	U2R	Data	Total
Our method	13	3	9	9	1	35
Cisco	3	1	5	1	0	10
Snort	9	4	13	5	1	32

It is essential to state the fact that Snort and Cisco are signature-based IDSs and only is used for comparing the proposed method with practical IDSs.

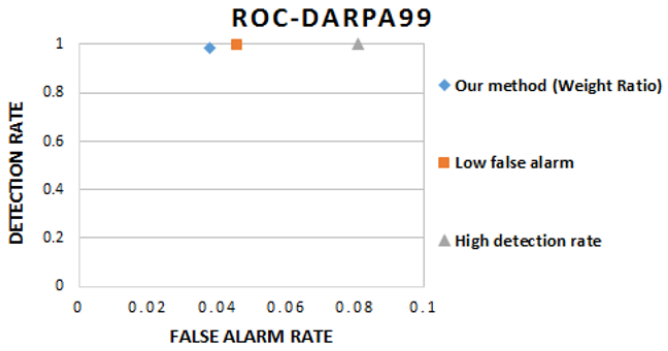

Fig. 10. ROC for our approach and other detection method

D. Comparing proposed approach with a packet-based anomaly detection

Hereafter, we compare our proposed method with a packet-based anomaly detection method which is introduced in [15]. It should be noted that this packet-based technique inspect the packet header and diagnose the attacks which occur on TCP protocol. There is two version of detailed results for packet-based method: Based on low false alarm and based on high detection rate.

The results of the experiment are drawn in Fig. 10 and the details of the experimental results are presented in table XIIII.

As stated in the details of table XIIII, in terms of false alarm and accuracy, our proposed method has satisfactory results rather than the packet-based method. It is necessary to mention that the packet-based method is able to detect just 80 percent of attacks, which occur on TCP protocols; however, our method is capable of detecting 100 percent of attacks of DARPA99 dataset. As a result, this could be the reason for the slight decrease in detection rate in our method.

TABLE XIIII
PERFORMANCE CRITERIA FOR OUR METHOD AND A PACKET BASED METHOD

Method	Detection rate	False alarm	Accuracy
Our method	0.986	0.038	0.963
Low false alarm	0.998	0.046	0.954
High DR	1.00	0.081	0.918

VII. CONCLUSION AND FUTURE WORK

Primarily, the goal of this paper was to propose network anomaly detection scheme based on data flow and graph concepts. Furthermore, generating flow-level and cluster-based dataset was the other goal of this work. Ultimately, a generic detection metric, which is named Weight Ratio (WR), was identified to diagnose various attacks on network with high detection rate and accuracy as well as low false alarm. The presented method is evaluated by DARPA99 dataset. Additionally, we compared it with alternative techniques to show the power of our proposed approach.

During the process of this work for solving the network-based anomaly detection problem, several issues appeal our attention as future work. We briefly referred to them as following.

In the first place, part of our future work fall into deploying

our proposed final detection model in order to test the performance of it on real-time traffic traces. The next desirable point of future work is to introduce an advanced clustering algorithm which is specific for clustering the TDG. The experimental results show that an accurate clustering method can lead to have proper evaluation criteria. Another issue is that we used the features of dataset to propose a static metric for detecting anomalies. To improve the detection metric (Weight Ratio), we suggested utilizing dynamic metrics which measure the distance of features over time. The last challenging work is to employ machine learning concepts for enhancing the performance of our detection method. We will manipulate the features of cluster-based dataset to make them appropriate for mentioned future work.

REFERENCES

[1] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," Data Mining and Knowledge Discovery, vol. 29, no. 3, pp. 626–688, 2015.

[2] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, pp. 303–336, 2014.

[3] J. Hoagland, "Spade, silican defense (2000)."

[4] H. S. Javitz, A. Valdes, and C. NRaD, "The nides statistical component: Description and justification," Contract, vol. 39, no. 92-C, p. 0015, 1993.

[5] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in Proceedings of the 2003 ACM symposium on Applied computing, pp. 346–350, ACM, 2003.

[6] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in Recent Advances in Intrusion Detection, pp. 203–222, Springer, 2004.

[7] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," IEEE communications surveys & tutorials, vol. 12, no. 3, pp. 343–356, 2010.

[8] A. Sperotto, "Flow-based intrusion detection," in Ph.D. thesis, University of Twente, 2010.

[9] L. Hellemons, L. Hendriks, R. Hofstede, A. Sperotto, R. Sadre, and A. Pras, "Sshcure: a flow-based ssh intrusion detection system," in Dependable Networks and Services, pp. 86–97, Springer, 2012.

[10] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle, "Grids-a graph based intrusion detection system for large networks," in Proceedings of the 19th national information systems security conference, vol. 1, pp. 361–370, Baltimore, 1996.

[11] D. R. Ellis, J. G. Aiken, A. M. McLeod, D. R. Keppler, and P. G. Amman, "Graph-based worm detection on operational enterprise networks," tech. rep., Citeseer, 2006.

[12] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese, "Network traffic analysis using traffic dispersion graphs (tdgs): techniques and hardware implementation," 2007.

[13] D. Q. Le, T. Jeong, H. E. Roman, and J. W.-K. Hong, "Traffic dispersion graph based anomaly detection," in Proceedings of the Second Symposium on Information and Communication Technology, pp. 36–41, ACM, 2011.

[14] M. V. Mahoney and P. K. Chan, "Phad: Packet header anomaly detection for identifying hostile network traffic," 2001.

[15] P. Manandhar and Z. Aung, "Towards practical anomalybased intrusion detection by outlier mining on tcp packets," in Database and Expert Systems Applications, pp. 164–173, Springer, 2014.

[16] B. Claise, "Cisco systems netflow services export version 9," 2004.

[17] B. Claise, "Specification of the ip flow information export (ipfix) protocol for the exchange of ip traffic flow information," tech. rep., 2008.

[18] G. Sadasivan, "Architecture for ip flow information export," Architecture, 2009.

[19] Z. Mingqiang, H. Hui, and W. Qian, "A graph-based clustering algorithm for anomaly intrusion detection," in Computer Science & Education (ICCSE), 2012 7th International Conference on, pp. 1311–1314, IEEE, 2012.

[20] S.-n. Yin, Z.-g. Chen, and S.-R. Kim, "Ldfgb algorithm for anomaly intrusion detection," in Information and Communication Technology, pp. 396–404, Springer, 2014.

[21] D. Doval, S. Mancoridis, and B. S. Mitchell, "Automatic clustering of software systems using a genetic algorithm," pp. 73–81, 1999.

[22] K. Kendall, "A database of computer attacks for the evaluation of intrusion detection systems," tech. rep., DTIC Document, 1999.

[23] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," Computer networks, vol. 34, no. 4, pp. 579–595, 2000.

[24] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, et al., "Evaluating intrusion detection systems: The 1998 darpa offline intrusion detection evaluation," in DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings, vol. 2, pp. 12–26, IEEE, 2000.

[25] M. Roesch et al., "Snort: Lightweight intrusion detection for networks.," in LISA, vol. 99, pp. 229–238, 1999.

[26] S. Convery and B. Trudel, "A security blueprint for enterprise networks," 2000.

[27] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of darpa dataset for intrusion detection system evaluation," in SPIE Defense and Security Symposium, pp. 69730G–69730G, International Society for Optics and Photonics, 2008.