Hybrid Intrusion Detection in Information Systems

David Pierrot, Nouria Harbi, Jérôme Darmont

Hybrid Intrusion Detection in Information Systems

David PIERROT, Nouria Harbi, Jérôme Darmont
Univ Lyon, Lumière ERIC EA 30835 F90676 Bron Cedex France
{david.pierrot1,nouria.harbi,jerome.darmont}@univ-lyon2.fr

***Abstract*—The expansion and democratization of the digital world coupled with the effect of the Internet globalization, has allowed for individuals, countries, states and companies to interconnect and interact at incidence levels never previously imagined. Cybercrime, in turn, is unfortunately one the negative aspects of this rapid global interconnection expansion. We often find malicious individuals and/or groups aiming to undermine the integrity of Information Systems for either financial gain or to serve a cause. Our study investigates and proposes a hybrid data mining methodology in order to detect abnormal behavior that could potentially threaten the security of an Information System, in a simple way that is understandable to all involved parties, whether they are security experts or standard users.**

***Index Terms*—Intrusion, Detection, Firewall, Security.**

I. Introduction

Nowadays, it is very easy to communicate, exchange ideas, acquire content and develop knowledge by using the Internet. The operational maintenance of Information Systems is therefore an essential criterion for any business, government and/or individual seeking to use this medium to deliver content, offer services, or simply wishing to communicate with others. Unfortunately, an often experienced negative aspect of this Information System's global expansion is a phenomenon called Cybercrime. Malicious individuals and/or groups aim to attack and harm individuals, companies and/or even government branches for monetary reward and/or in pursuit of a cause. The objective of this paper is to first analyze and explain the current state of intrusion detection practices, and secondly discuss the work that we carried out to facilitate Information System data flow visualization and first level intrusion/attack detection (scanning, brute forcing). The main contribution of this paper is the use and analyze firewall logs to detect misuses and abuse with data mining methods. It is important to note that we do not use any network packet inspection tools or attack classification made by an IDS. This paper is organized as follows. In section II, we present the overview of some related researches in the aera of intrusion detection systems. Our motivations for choosing this approach are detailed in section III. The data sets and the results used and obtained from our experimentation are presented in section IV. Finally, our conclusions and the scope of potential future work is presented in section V.

II. Related works

A. Intrusion detection systems

There are a variety of available tools (IDS, IPS, HIDS, Firewalls[1]) that allow for scanning and ensuring the relative security of an entire system. However, these tools themselves can be vulnerable, as they often misinterpret real-time observations, fail to report abnormal behavior, and can become quickly outdated, which can potentially result in a treat or attack to the individual system components mentioned above[1][2]. It is therefore appropriate and desirable to be able to respond in a timely manner from the instant an intrusion is detected, to deploy adequate countermeasures to respond swiftly to a potential cyberattack. The scope of current Intrusion Detection System solutions can be classified into three groups.

- NIDSs (Network Intrusion Detection Systems) can monitor data collected from their own network segment and signal abnormal transactions/behaviors[3].
- HIDSs (Host-Based Intrusion Detection Systems) are designed to monitor and detect irregularities in individual hosts. They monitor both inbound and outbound network activity, and moreover are capable of checking the system, software, and any relevant peripheral devices (such as USB storage).
- Hybrid IDSs, combine the different characteristics of NIDS and HIDS, Allowing for the possibility to check both network and application layers.

IDS' mission is to detect intrusion attempts as soon as they happen. All these various solutions are typically based around two concepts (attack signatures and known profiles/ behavioral model) which are generally implemented by almost all IDS. To keep the system current, it is possible to add new signatures/attributes or manually add new behaviors to keep the system current and to minimize false positives. Table I attempt to summarize the advantages and disadvantages of intrusion detection solutions. The main limitation of existng solutions reside in the fact that they do not take into account the almost continuous evolution of an Information System components.

TABLE I: Classic IDS advantages and disadvantages

	Avantages	Disadvantages
NIDS	Alarm in case of anomalies Multiple positioning Real time	Signatures update needed Absorption network traffic Ineffective for encrypted stream False positives Expert needed
HIDS	Workstation protection	Ineffective against attacks on multiple hosts Configurations depending on systems
Hybrid	Decrease false positives Real time	More sources, management and interpretation Event correlation more difficult for events and alarms

[1]A firewall is a network security system that controls the incoming and outgoing network traffic based on an applied rule set

B. Data mining-based IDSs

Since the initial studies by Denning [4], detection systems have continued to evolve. Data mining offers various solutions for the detection and analysis of computer attacks[5]. Lee and Stolfo use data mining methods[6], which do not require an experts intervention. These methods tend to generate a lot of association rules and therefore exponentially increase the system's level of complexity. This work is based on network capture tool tcpdump and Unix audit data. To use this method, it is compulsory to install a sensor network with a large memory storage space and it is only for Unix systems. Using a hybrid (combine different data mining methods) approach is very interesting because it takes care about alert management[7][8]. H.Nguyen et al.[9] use an ensemble system of classifiers, called CBE based on K-mean algorithm but the classe number must be fixed a priori [10]. This may take time depending on the number of servers to be monitored. A.R. Ajiboye et al. proposed to use the density-based DSB-CAN algorithm[11]. In fact, even DBSCAN[12] or OPTICS algorithm[13] are capable of determining the cluster number. However, it is compulsory for us to define two hard-to-estimate input parameters chiefly the radius of the cluster and the minimum required points inside the cluster for DBSCAN and fix the minipoints argument for OPTICS. Moreover with such a large dataset, these algorithms require large memory and computing resource. Using PCA model with K nearest neighborhood for intrusion detection (on a KDD 99 dataset)[14] or Hierarchical Agglomerative Clustering (HAC)[15] have the advantage of providing good graphic visualization to assist with finding the relevant numbers of cluster. The studies and works presented in the current section are all based on data flow from either KDD99 based models[16]. We feel there are several disadvantages of using this process. First, it is necessary to capture network traffic and this can affect memory storage and time analysis.

III. Motivations and proposal

A. General principle

We believe security systems should be simple enough and automated as much as possible as to be understood and deployed when needed by all users of a specific system. It is critical for Information System supervision tools to allow for behavior based decision support methods (behavior of users, services, servers, etc.), so that accurate event data streams and attack analysis can happen in real-time. Considering these ideas should result in the ability to predict risks and threats, not only from known and monitored assets (servers), but also from the evolution of the Information System. Our study is based on four phases:

- Phase 1: Monitoring and visualization of network data. Graphical representation of computer network activities via a data model.
- Phase 2: Behavior analysis and alert based on data mining methods.
- Phase 3: Risk scoring and evaluation.
- Phase 4: Action plan building.

Phase 1 and 2 could be merge, the vizualtion phase is finaly useful when an alert is received or to make a diagnostic. This model is very similar to the Defense Life-cycle[18] and to the intrusion response systems of Kanoun et al.[17]. We added a graphic part and we do not have any attack classifications made by an IDS. Hence, we opted for a monitoring/visualization phase which is a conventional approach and then an assessment of the risk is determined by an analysis of behavior. At the end, an action plan will help to stop or to prevent an abnormal action. Before introducing the concepts of intrusion detection or behavior analysis, let us explain usual procedures. Typically, a network intrusion is based on the following five steps.

1) Reconnaissance: use different sources from the internet for gathering potential target information (address IP, site and DNS name owner, email address, social network).
2) Scanning the port: find and record ports, services, operating system and versions used.
3) Gaining access: analyze vulnerabilities for potential exploitation.
4) Maintaining access: keep the system or network permanently engaged.
5) Covering tracks: delete and remove traces from the network and/or system.

The first step is extremely difficult to detect because it is dependent on the digital footprint of a business and its staff. Thus, our work focuses on steps 2 and 3 which correspond to the phases 1-3 of the cyber kill chain 3.0 model[19]. All computer attacks usually start by gathering intelligence. Step 2's ultimate goal is to receive a scanning service offer from a server. A successful information request provides a list of available services (HTTP, HTTPS, FTP, mail, etc.), the version of the services used (Microsoft Internet Information Services or Apache for a Linux server), and the base of the operating system. Scanning activities represent more than 80% of detected cyberattacks[20]. Once this information is known, it is not difficult to find and/or create malicious tools/scripts to exploit known vulnerability in the target server. Hence, it is essential to be able to detect quickly whenever this kind of scanning and enumerating phase is taking place. The methods for information gathering have also evolved. It is now possible to use multiple IP addresses[2], or use stretched intelligence to limit the arrival of access requests to different ports on a server. The attacker's goal is to receive the required information while minimizing traces of any exchanges. These are used to identify the information needed for preparing potential attacks and for detecting technical anomalies and vulnerabilities.

Although this paper focuses on Phase 2, we still need to quickly provide a summary of Phase 1. This phase is used to provide full data visualization to relevant users (security ingeneer, network analyst, chief information security officer). It is a crucial phase in terms of decision-making for security readiness. Additionally, they are the preamble to the data mining process that is conducted in phase 2. We have decided

[2]Firewall/IDS Evasion and Spoofing, http://nmap.org/book/man-bypass-firewalls-ids.html

to use a firewall as a capture data source. Due to its placement, a firewall can potentially offer full and complete data flow visibility. It can also open the opportunity to trace and keep information to highlight these flows on events that have been either authorized and/or prohibited. Moreover, the firewall is the most security component used, the business equipment rate was 95% in 2010[21]. Pre-treatments is made from logs connection filtering equipment (log option) and they are sent to the Syslog-ng server[3]. Through its Perl compatible regular expression(PCRE) pattern matching and filtering options, all flows are saved on the databaseThereafter, computation is carried out via a Perl script. Finally, Graphviz Suite[4] and Afterglow [5] script components are used to create graphs.The terms of the variables listed below are exported to data containers.

- source IP address, destination IP address
- destination port, protocols (UDP and TCP)
- date and time
- firewall policy rule number matching
- firewall action (accept or reject)

Choosing these variables, we offer the possibilty to integrate several types of firewall of different brands. Futhermore, a network probe implementationis no longer necessary.

B. Phase 2: Behavior analysis and alert

Phase 2 is intended to allow for data analysis and the detection of abnormal behavior (if any, which in turn will trigger an alert (if needed). We feel it would be wise to use firewall event data mining methods for anomaly detection. Besides, Our work uses session data from firewall Logs as explained in section IV. Our main goal is to predict risks and threats according network transactions. After consideration of the data studied in Phase 1, we have to perform preprossing to reduce the available data range. This is required because, for example, the destination port variable includes 65535 possibilities. We opted for grouping variables in the three following categories:

- well-known ports (under 1024) accept and deny;
- port 1024 through 65535 accept and deny;
- administration port (portadm), port activity for server or database administration accept and deny.

Preprocessing aggregates of IP source addresses and combines the occurrences according to the categories mentioned above. Furthermore, it would probably be wise to consider the total number of transactions carried out by the same source IP address and the number of flows rejected (action deny) and allowed (action permit) by the firewall. This can be achieved from the data already collected, which is available in different data containers. Once preprocessing was complete, we sought the advice of experts (five in total) to determine whether the observed behavior could definitely be defined as a potential risk or threat. From labeled training data set, we want to predict intrusion from the aggregates of IP source addresses. The supervised learning give us the possiblily to construct an estinator which is able to predict the risk. Experts analysis give us a picture of the security policy. So, we intraged this data set training into a framework that we made and called D113[22]. Throughout the supervised leaning process and when using the aggregated data obtained through preprocessing, we focuse only on the IP source classification problem. This method does however allow for a few potential risks when compared to a security policies defined by experts (which are not necessarily exhaustive). For example, some servers could add or suppress event data as the Information System evolves. In our view, it would be prudent to analyze the suffered behavior by the different servers in order confirm the supervised learning result of. By using two different learning methods[23], it should be possible to have a quick overview of the IP source and the inherent misuse risks and servers deviation behavior. We want finaly group behavoir into different groups of activities. This method opens up the opportunity to detect abnormal and misuse activity events. On the other hand, we decided to look for abnormal activity on the destination address (server). With this kind of process, it is possible to detect and monitor any and all behavior on the server. Our first step is to create a data frame and extract every data flow received on each server. The next step is to identify and show that behavior is different. There are a large number of methods to execute unsupervised learning. We want to group IP source behavior according the destination server IP. This cluster analysis revolves around the concept of placing a set of objects in the same group or cluster. By following this methodology, it should be possible to identify and verify any behavior deviation. Again, there are various algorithms for achieving this. The main problem with clustering methods is how to determine the numbers of clusters (classes) to be used. Both these solutions have the advantage of providing good graphic visualization to assist with finding the relevant numbers of cluster. Thus, to find the best method for determining the correct cluster number, we use internal validation. We have tested different algorithms and after analyzing the results, we opted for using the Partitioning Around Medoids (PAM)[24], as this provides the best set of results. But the relevant numbers of cluster have to be fixed a priori as with the other methodologies. To solve this problem, using the PAMK (Partitioning around medoids with estimation of number of cluster) method automatically gives the correct class number by estimating the optimum average.

IV. Validation / Proof Of Concept

A. Use case

We work in our study on architecture from a health industry public company with 92,000 employees. Our analysis focuses on three interconnected networks within an extended network (WAN: Wide Area Network) geographically remote and protected by filtering equipment. The objective is to be able to monitor firewall related events and export them to a data container. To simplify references to different networks, the following nomenclature is used:

- production site: SP1;
- qualification site: SQ1;

[3]Log management solution, http://www.balabit.com/network-security/
[4]Graph Visualization Software, http://www.graphviz.org
[5]Link Graph Visualization, http://afterglow.sourceforge.net/

TABLE II: Flow treated with SP1, SQ1, SAB1 in numbers of lines

	Daily flows processed	Average per minute
SP1	9 886 928	6 865
SAB1	572 272	397
SQ1	20 670	14

- office and remote administration site: SAB1;

These three sites are currently operational, and the information processed and analyzed in the following sections corresponds to real-time data production. IP addresses have been anonymized for privacy reasons. Network SP1 has its own pasteboard outcome of events sent in real-time by the firewall. Network SP1 provides services to 14 million people. Data used is financially sensitive, and needs 9.2 terabytes for storage related to transactions several million Euros a day. Data are heterogeneous and received from several different sources. Table II summarizes the volume in number of lines processed by the filtering equipment. In order to strengthen the integrity of this work, we requested for several users from several companies who work as Information Systems security managers to review and critique our implementation.

B. Phase 2 use case results

Execution of Phase 1 provides us with the opportunity to review network activity. This, in turn, allows for easy understanding when compared to reviewing raw data and events (Figure 1). It is important to note that our study focuses only on session type information (source IP address, destination IP address, protocols and services). After preprocessing, we

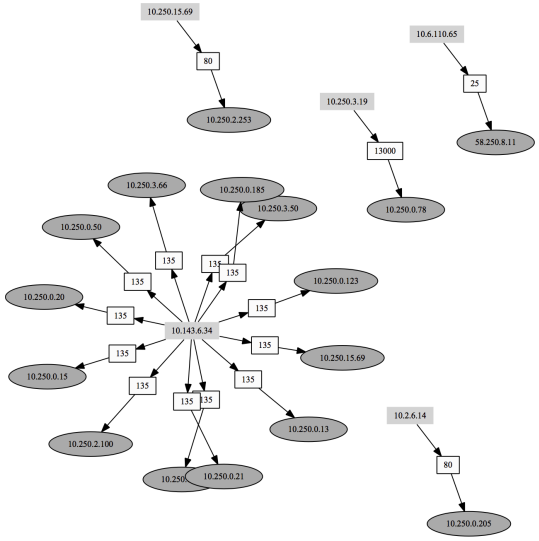

Fig. 1: D113: Example of rejected flow

integrate the new data and opte for supervised learning. We can detect an malicious activity as shown in line 3 of Table III, so the 0.7% permit flow could be defined as a marker/trigger for an intrusion. This should be sufficient information to attempt a server access attack. We have tested different kinds of methods for information gathering, such as classic scan (one shot scan), decoy scan (add multiple source of the scan), zombi (use a compromise victim to send the scan), distributed scan (use different machines for the same scan) with different delays (increase the delay when scanning ports; in fact, we saw that a long interval port scanning can hardly find or detect events). In order to have the best detection accuracy, we also teste different supervised learning algorithms by using cross validation. We see that the best results are achieves when using the Random Forest algorithm (Table IV).

TABLE III: Aggregated flow with risk analysis supervised learning

Sum	Action deny	Action Permit	Inf 1024	Sup 1024 port	Adm	Risk
16	0.0	100.0	0.0	0.0	0.0	No
12	0.0	100.0	0.0	0.0	0.0	No
3296	99.3	**0.7**	61.9	38	0.1	Yes
36	100.0	0.0	0.0	100	0.0	No

TABLE IV: Supervised algorithms error rate comparison

Algorithms	Error rate %
CART	9.09
C4.5	11.3
C5.0	8.5
Random Forest	8.2

For the second step of phase 2, we must to find the best method for determining the correct cluster number. We have tested different algorithms and after analyzing the results, we opted for using the Partitioning Around Medoids (PAM) algorithm[24], as this provides the best set of results. But the relevant numbers of cluster have to be fixed a priori as the other methodologies. To solve this problem, the PAMK (Partitioning around medoids with estimation of number of cluster) method automatically gives the correct class number by estimating the optimum average. We compared the PAMK methodology output versus priori methods (Figure 2), and we noticed that the results were positive for all servers involved in our study. For instance, the computation time for 53 servers is approximately 0.353 seconds. We have built a training data set and saved the class attribution result for reference. Which means that each server has a specific class number, and any derivation could be potentially considered as abnormal behavior. Further, we create a test data set for different kind of scans and we launched several brute force and path traversal attacks (Phase 2 obtain access) [6]. Bydoing this, we saw that the number of class was directly detected in the target servers. For instance, the main server shifted from 6 to 2 classes. This source of behavior deviation is automatically identified as abnormal activity. We then extract data in order to trigger an alert. However, it is possible for an attacker to send massive amounts of data as decoy to try and hide a discreet attack. So, we extract that to make new computations until a similar class number is obtained. Therefore, we can use a variation coefficient (CV) to generate a score[25]. For the moment, we do not manage the score, but the variation coefficient provides a general overview which is good enough for the training dataset when compared to the test data set. We believe that

[6]https://www.owasp.org/index.php/Path_Traversal

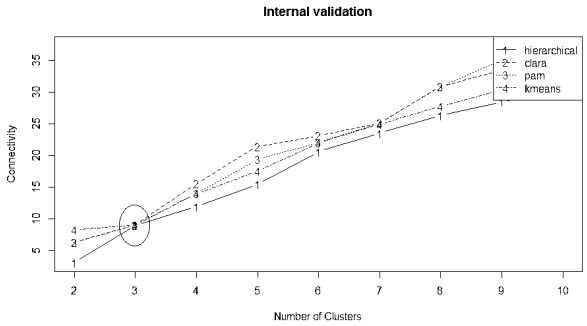

Fig. 2: Example of internal validation for one server

with this kind of process, it will be possible to detect and identify destination servers that have undergone changes in behavior. Once we know this, we can then have a specific overview based on server class variation with CV, which allows to focus on the most critical activity. Our approach allows us to detect large scan, brute force attacks and path tranversal attacks, without any help from experts.

C. Experts feedback

From the point of view of experts (2 senior security ingeneer, 1 security consultant,1 chief information security officer, 1 senior network analyst), the D113 tool give us the possibility of obtaining access to all the rejected flow including port scanning and brute force attacks, etc... In addition of that, this tool is very useful to make diagnostics based on verification of security filtering rules. It was possible to report and alert abnormal access to the intrusion detection team. In fact, the diffrents IDS did not identify these actions. With the data mining learning methods, experts were able to visualize the different behaviors on servers. However, the provisional alert thresholds does not appear to correspond to all firewall according to their volumetric flow. The establishment of effecive variation coefficient is expected to change this.

V. Conclusion and perspectives

The work done on Phase 2, allows us to check internal security by using Random Forest supervised method with an experts help. The PAMK unsupervised method allows us to detect different kind of information gathering, potential attacks, and other access methods. By using the variation coefficient (CV) it is possible to score the behavior change levels. In order to make Phase 4 operational, we have worked on a simple method for interpreting this scoring and triggering alerts. Additionally, the work completed in Phase 1 has to be improved as to result in a specific and accurate overview of abnormal behavior and misuse. We suggest that other supervised methods, like bagging or boosting, are tested in order to have improved accuracy. These phases will generate association rules based on the assets covered by the several attack vectors. A consideration of techniques from supervised and unsupervised learning allow for better account of the attacks within the automatic definition of reported thresholds. We believe this would create a scalable and adaptive real-time system for identifying changes in behavior due to intrusions and attacks.

References

[1] A. Valdes and H.Alvaro., "Flips: Hybrid adaptive intrusion prevention," *8th International Symposium, RAID 2005*, vol. 3858, pp. 82–101, 2005.

[2] U.Sandhu and al., "A survey of intrusion detection & prevention techniques," *International Conference on Information Communication and Management*, vol. 16, pp. 66–71, 2011.

[3] H. Bhruyan and al, "Survey on incremental approaches for network anomaly detection," *International Journal of Communication Networks and Information Security*, vol. 3, pp. 226–238, 2011.

[4] D. Denning, "An intrusion-detection model," *Journal IEEE Transactions on Software Engineering - Special issue on computer security and privacy*, vol. 13, pp. 222–232, 1987.

[5] A. J. Deepa and V. Kavitha, "A comprehenseive survey on approaches to intrusion detection system," *International Conference on Modelling Optimization and Computing*, vol. 38, pp. 2063 – 2069, 2012.

[6] W. Lee and al., "A data mining framework for building intrusion detection model," *IEEE symposium on security and privacy*, pp. 120–132, 1999.

[7] M. Siraj and al., "A hybrid intelligent approach for automated alert clustering and filtering in intrusion alert analysis," *International Journal of Computer Theory and Engineering*, vol. 1, pp. 539–545, 2009.

[8] E. Bahri and N. Harbi, "Real detection intrusion using supervised and unsupervised learning," *International Conference on Soft Computing and Pattern Recognition*, pp. 321–326, 2013.

[9] H. Nguyen and al, "An effcient local region and clustering-based ensemble system for intrusion detection," *15th International Database Engineering and Applications Symposium*, vol. 185-191, 2011.

[10] R.Lavanya and al., "Improving Network Intrusion Detection Based on Multi Objective Criteria," *International Journal of Advanced Research Trends in Engineering and Technology*, vol. 2, pp. 34–39, 2015.

[11] A. Ajiboye and al., "Anomaly Detection in Dataset for Improved Model Accuracy Using DBSCAN Clustering Algorithm," *African Journal of Computing and ICTs*, vol. 8, pp. 39–46, 2015.

[12] M.Ester and al., " A Density-Based Algorithm for Discovery Clusters in Large Databases with Noise," *KDD 96*, pp. 226–231, 1996.

[13] M.Ankerst and al., "OPTICS: ordering points to identify the clustering structure," *International conference on Management of data*, vol. 28, pp. 49–60, 1999.

[14] Z. Elkhadir and al., "Intrusion Detection System Using PCA and Kernel PCA Methods," *International Journal of Computer Science*, vol. 43, pp. 72–79, 2016.

[15] K.Debabrata and al., "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM," *11th International Conference on Distributed Computing and Internet Technology*, vol. 8956, pp. 377–390, 2015.

[16] "Kdd cup 1999 data," 1999-01-02, URL: http://kdd.ics.uci.edu/databases/kddcup99/ [accessed: 2016-06-27].

[17] W.Kanoun and al., "Automated reaction based on risk analysis and attackers skills in intrusion detection systems," *10th International Conference on Risks and Security of Internet Systems*, pp. 117–224, 2008.

[18] A.Shameli-Send and al., "Taxonomy of intrusion risk assessment and response system," *Journal Computers and Security*, vol. 14, pp. 1–16, 2014.

[19] S. Yadav and D.Mallari., "Technical aspects of cyber kill chain," *3rd Communications in Computer and Information Science*, vol. 536, pp. 438–452, 2016.

[20] Molina and al., "Operationnal experiences with anomaly detection," *Computers & Securyt*, vol. 31, pp. 273–285, 2012.

[21] CLUSIF, "Menaces informatiques et pratiques de sécurité en france," p. 32, 2012, [accessed: 2016-07-27]. [Online]. Available: http://cyber-serenite.fr/uploads/documents/clusif-rapport-2010.pdf/

[22] D.Pierrot and al., "D113 : une plateforme open-source dédié à l'analyse des flux et à la détection des intrusions," *Extraction det Gestion des Connaissances*, vol. 28, pp. 449–454, 2015.

[23] S. Meesala and B. Xavier, "A Hybrid Intrusion Detection System Based on C5.0 Decision Tree and One-Class SVM," *International Journal of Current Engineering and Technology*, vol. 5, pp. 59 – 70, 2015.

[24] F. Lamiaa and Manal.H., "Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning," *International Journal of Computer Science Issues*, vol. 9, pp. 10–25, 2013.

[25] R. Breunig, "An almost unbiased estimator of the coefficient of variation," *Economics Letters*, pp. 15–15, 2001.