

An Improved Intrusion Detection System Based on a Two Stage Alarm Correlation to Identify Outliers and False Alerts

Fatma Hachmi¹(✉) and Mohamed Limam²

¹ ISG, University of Tunis, Tunis, Tunisia
hachmi.fatma@gmail.com

² University of Tunis, Tunisia and Dhofar university, Salalah, Oman

Abstract. To ensure the protection of computer networks from attacks, an intrusion detection system (IDS) should be included in the security architecture. Despite the detection of intrusions is the ultimate goal, IDSs generate a huge amount of false alerts which cannot be properly managed by the administrator, along with some noisy alerts or outliers. Many research works were conducted to improve IDS accuracy by reducing the rate of false alerts and eliminating outliers. In this paper, we propose a two-stage process to detect false alerts and outliers. In the first stage, we remove outliers from the set of meta-alerts using the best outliers detection method after evaluating the most cited ones in the literature. In the last stage, we propose a binary classification algorithm to classify meta-alerts whether as false alerts or real attacks. Experimental results show that our proposed process outperforms concurrent methods by considerably reducing the rate of false alerts and outliers.

Keywords: Clustering · Binary classification · False positives · Intrusion detection systems · Outliers

1 Introduction

The ultimate goal of computer security is to protect networks against criminal activities such as violation of privacy, corruption of data and access to unauthorized information. In fact, computers are in need for powerful security technologies to secure the information system and to prevent hackers from destroying it. In fact, intrusion detection systems (IDS)s are considered as essential components for the protection of computer networks. Therefore, their accuracy depends on their ability to detect real threats on the network and to alarm the administrator about them. Despite the major role of an IDS as a component of the security infrastructure, it is still far from perfection since it tends to generate a lot of noisy alerts and a high rate of false alerts. A false alerts is defined as a signal triggered by an IDS reporting an attack but in reality it is just a normal network traffic. An outlier is defined as a noisy and inconsistent data point characterized by its dissimilarity from other observations in a given data set.

To enhance the accuracy of IDSs, we propose a two-stage process to eliminate outliers and to reduce the rate of false alerts. First, we clean the set of alerts by removing noisy meta-alerts or outliers. In fact, we evaluate the most cited outliers detection methods and then we select the best one to be integrated in our proposed process. Four outliers detection approaches are commonly used in the literature: the clustering-based, the density-based, the distance-based and the distribution-based approaches. In the second stage, we begin by clustering the set of cleaned alerts into a set of meta-alerts based on several attributes extracted from the alert database. Then, a binary classification algorithm (BCA) is proposed to identify false positives (FPs). The remainder of this paper is organized as follows. Section 2 gives an overview of the related works. Section 3 describes the proposed method for outliers detection and FPs reduction. Experimental results and performance comparisons are given in Sect. 4. Finally, conclusion and future work are given in Sect. 5.

2 Related Works

Many research works were conducted to reduce the rate of false alerts generated by intrusion detection systems. Reference [1] introduced a new alert correlation technique to extract attack strategies based on the causal relationship between alarms. This technique is based on two approaches: Multilayer perceptron (MLP) and support vector machine (SVM). In the experimental study they used DARPA 2000 to test their proposed technique. In fact MLP and SVM algorithm require a training set to build a model that will be used to predict the right decision for a new observation. Using elementary alerts from the training set is not valuable since one event may produce multiple alert signature. In order to improve the accuracy of an IDS, [2] proposed a two-stage alarm correlation and filtering technique. The first stage aims to classify the generated alerts based on the similarity of some attributes to form partitions of alerts using SOM with k-means algorithm. The second one aims to classify the meta-alerts created in the first stage into two clusters: true alarms and false ones. The binary classification is based on the collection of seven features extracted from the set of meta-alerts using SOM with k-means to cluster the input set. In the experimental study, they used DARPA 1999 to test their proposed technique. Unfortunately the use of SOM with k-means in the second stage is not efficient since the administrator should manually determine which cluster contains the true alarms by examining the two attributes: alarm frequency and time interval. Reference [3] introduced a network detection technique based on neural networks. First the detected network traffic is clustered using SOM. The partitions are displayed to the administrator so he can recognize attacks clusters. Then, MLP algorithm is applied to the output of SOM to efficiently model the network traffic. In the experimental study they used DARPA 1999 training data set along with three types of attacks to test their proposed technique: denial of service attacks, portscans attacks, and distributed denial of service attacks. Unfortunately, this technique requires a well experienced network administrator

to recognize the attacks. Moreover, this technique represents a real challenge if other attacks are added. Reference [4] developed a network detector based on SOM algorithm called NSOM. It allows the classification of real time network traffic. Once the detected alerts are normalized and the classification features are extracted, alarms that represent a normal behaviour will be grouped in one or more clusters and attacks will be placed outside. In fact, results show that NSOM allows the classification of normal traffic and attacks graphically and dynamically. But if there are different types of attacks NSOM will face a serious problem since it has to cluster each type of attack distinctly. Reference [5] introduced the decision support classification (DSC) alert classification. It collects the alerts generated in an attack-free environment. So, all alarms are considered as FPs in this environment and the recorded patterns in this case define the normal behavior and are called patterns of FPs. Then, DSC removes FPs based on these patterns. Reference [6] used a knowledge-based evaluation for the proposed post-processor for IDS alarms. This system uses background information concerning the hosts available in the network and generates a score for each alarm based on the exploited vulnerability. This score measures the importance of each alert. Then, based on the value of score threshold, a binary classifier groups the alert as real attacks or FPs. Reference [7] proposed a correlation framework that reduces the number of processed alerts in the first phases by removing the inconsistent and false alerts. Reference [8] proposed a Bayesian network model for classifying the alerts generated by IDSs as attacks or false alarms. Reference [9] used a Genetic Fuzzy Systems within a pairwise learning framework to improve IDSs. Reference [10] developed a novel approach called the cluster center and nearest neighbor (CANN) where two distance measures are computed then summed.

3 The Proposed Method

This work aims to improve the accuracy of IDSs by eliminating inconsistent alerts and false ones. To achieve this goal, we propose a two-stage process that begins with a cleaning the set of alerts to remove outliers. Once only consistent alerts remain, a clustering step is performed by applying k-means algorithm. Then, we propose a BCA that aims to identify FPs by comparing the similarity of meta-alerts with a labeled training set used as a classification model.

3.1 Outliers Detection

Outlier detection methods aim to clean databases from unusual objects. This helps building consistent data sets that can be used to extract knowledge in different domains. Four outliers detection approaches are commonly used in the literature. Distribution-based approach introduced by [11], consists of developing statistical models from a given data set for the normal behavior and then perform statistical tests to decide if an observation belongs to this model or not. This approach assumes that each data set has a distribution. [12] proposed the active outlier method (AO). AO invokes a selective sampling mechanism which is based

on active learning. However, this approach is not appropriate in multidimensional scenarios since they are univariate in nature and a prior knowledge about the data distribution is needed. Also, the construction of a probabilistic model based on empirical data is a difficult computational task and the chosen sample is not guaranteed to match the distribution law.

The Clustering-based approach introduced by [13], considers clusters of small sizes as clustered outliers. But if the separation between clusters is large enough, then all clusters are considered as outliers. So, the clustering approach by itself is not sufficient to detect outliers efficiently.

The distance-based approach introduced by [14], considers an object O in a data set as an outlier if there are less than M object, within the distance d from O . The major limitation of this approach is that it is difficult to set the values of M and d . An extended method based on the distance of an observation O from its k th nearest neighbor (KNN) is proposed by [15]. It sorts the top k vectors based on the distance between it and its KNN. Besides, [16] propose an algorithm that computes the outlier factor of each object as the sum of distances from its KNN. For large data sets, KNN-based methods are very time-consuming. Reference [17] introduced neighborhood approximation properties. Reference [18] proposed a new method called neighborhood outlier detection (NED) to detect outliers based on neighborhood rough set. But, the efficiency of this method depends on the appropriate selection of the neighborhood parameters.

The density-based approach originally proposed by [19], gives to each object a factor called the local outlier factor (LOF) to measure the degree of an object being an outlier.

To remove outliers from the set of meta-alerts, we use four methods, one from each approach, AO, k-means, NED and LOF. We evaluate them on different data sets and then based on the overall results of the proposed process, we select the best method in the context of intrusion detection.

3.2 False Positives Reduction

In this final stage, our interest is to reduce the rate of false alarms generated by IDSs. As an input set, we use the cleaned set of alerts which is generated by the previous stage. Then, we reduce the huge number of alerts by clustering the testing sets and finally a BCA is applied to identify the set of FPs.

The Clustering Step. The clustering step aims to reduce the number of generated alerts by correlating the similar ones together. The correlation is based on the similarity-based technique. It is based on maximizing the degree of similarity between objects in the same cluster and minimizing it between clusters. Therefore, the classification is based on the similarity of some selected attributes. Four attributes are used. In fact, IP addresses represent the identity of the hacker in the network and the timestamp define the time frame of a given event while the protocol defines its nature.

- Source IP address
- Destination IP address
- Timestamp
- Protocol

Before clustering the alerts, we normalize all the attributes to ensure having reliable clustering results. As a clustering algorithm, we propose to use k-means algorithm since it is a simple unsupervised learning method. K-means defines k centroids, one for each cluster and then clusters all data into the pre-defined k clusters. The grouping is done by computing the sum of squared Euclidean distances from the mean of each cluster. Moreover, k-means is very appropriate to ensure having reliable results from the next stages of our proposed process since it aims to maximize the distance between the clusters and to minimize the dispersion within them.

Since the detection of FPs requires wide knowledge about the network traffic and expertise in the domain, we use a training set to generate a model for the network traffic. But, the generated alerts are not valuable in the creation of the model since a single event may produce multiple alerts. So, the aforementioned attributes are used to cluster the training set. First, we split the training set on two big pre-clusters, the first one clusters the false alarms and the second one groups real attacks. Second, inside each pre-cluster we apply k-means algorithm to group similar alerts together. Therefore, the training set is transformed from a set of elementary insignificant alerts into a set of labeled clusters.

The Binary Classification Algorithm. The binary classification aims to identify FPs from the set of meta-alerts created by the clustering step. The attributes list is extracted from each consistent cluster. Four attributes are judged useful for the binary classification, namely

- Number of alerts in each cluster
- Signature type
- Protocol number
- Alert priority

Based on these extracted attributes, we propose a BCA to classify each meta-alert from the testing set whether as true alert or a false one. In fact, there is two labeled training clusters. The first one F includes the false meta-alerts and the second one T includes the true meta-alerts. Based on the aforementioned attributes, we compute the Euclidean distance between each meta-alert (MA) from the testing set and the centroid C_1 and C_2 of the two clusters F and T respectively. If the distance between MA and C_1 ($\text{Dist}(MA_i, C_1)$) is lower than MA and C_2 ($\text{Dist}(MA_i, C_2)$), then the probability that MA is a false alerts is high. To ensure that MA is correctly classified as a false alert, we propose to test its similarity with each false meta-alert (MT) inside F. Therefore, if the distance between MA and MT is lower than the maximum distance between false meta-alerts inside F, then MA is a false meta-alert otherwise it is a real attack. The proposed algorithm is detailed as follows.

Algorithm 1. Proposed classification algorithm**Input:** meta-alerts of the testing set, training clusters**Output:** False alarms**Begin**

N is the number of meta-alerts

F is the set of training false positives

T is the set of training true alerts

FN : number of clusters inside F

TN : number of clusters inside T

 C_1 : centroid of the set F C_2 : centroid of the set T**For** each meta-alert (MA) i from N **If** $\text{Dist}(MA_i, C_1) \preceq \text{Dist}(MA_i, C_2)$ **For** each meta-alert (MT) j from FN **If** $\text{Dist}(MA_i, MT_j) \preceq \text{AVG}(\text{Dist}(MT_j \text{ in } F))$ insert MA_i in the cluster F

FN=FN+1

End For C_1 is the new centroid of cluster F **End If****End For****End**

4 Experimental Results

To test the efficiency of the proposed technique, we used a public data set named DARPA 1999, commonly used for the evaluation of computer network sensors. Our experiments are based on the off-line evaluation sets:

- As a training set, we use the first and third weeks of the training data which are attacks free and the second week of the training data which contains a selected subset of attacks. The primary purpose of this training set is the detection of false alerts from the testing set.
- A selected sample from the fourth week is used as our first testing data set in order to evaluate our proposed process.
- A selected sample of the fifth week is used as our second testing data set.

4.1 The First Stage

To remove outliers from the clusters generated by the first stage, we use four methods namely, AO, K-means, NED and LOF.

Tables 1 and 2 give the number of outliers generated by each method for the first and second testing sets respectively.

We create four different subsets for each testing set. In the first one, we remove the outliers detected by LOF and then we apply BCA to evaluate its

Table 1. Number of outliers in testing set 1

Outliers detection methods	Detection rate
LOF	81
k-means	66
AO	52
NED	79

Table 2. Number of outliers in testing set 2

Outliers detection methods	Detection rate
LOF	72
k-means	66
AO	44
NED	81

performance with LOF. The second one includes the consistent alerts after eliminating the outliers detected by NED. In the third one we remove the noisy alerts identified by AO and the final one contains the alerts after removing the outliers defined by K-means. For each testing set, we evaluate the performance of the proposed BCA based on those four subsets and the best outlier detection method is used for our proposed process. The overall process is also compared to the methods SOM with k-means and DSC proposed by [2, 5] respectively.

4.2 The Second Stage

The Clustering Step Evaluation. In this step, we follow the approach used by [3] to select the appropriate k value because we don't know the exact number of clusters. The system performs 500 randomised trials and the best classifications are selected based on the minimal sum of squared errors and the highest frequency.

To ensure having the best clustering solution, we test the different values of k generated from the randomized trials until we get the optimal partitioning of the data. The latter is based on several validity measures to test the quality of the partitions. The first validity measure, separation index (SI), determines the average number of data and the square of the minimum distances of the cluster centers. Indeed, a small value of SI indicates an optimal partitioning. The second validity measure, Dunns index (DI), is used to identify whether clusters are well separated and compact or not. A big value of DI implies a good clustering. The third validity measure, Xie and Benis index (XB), quantifies the ratio of total variation within cluster and the separation of clusters. To have an optimal number of partitions, the value of XB index should be minimized.

Table 3 values of SI, DI, and XB for different values of k using the two testing sets. As illustrated, we notice that the best solution is when k is equal to 85

Table 3. Clustering evaluation of testing set 1

Testing set	Pairs of clusters	SI	DI	XB
1	k=66	5.6814e-004	0.0149	4.2338
1	k=85	4.5002e-004	0.0274	3.6083
2	k=79	5.8351e-004	0.0236	4.7062
2	k=77	3.6143e-004	0.0285	3.1397

for the first testing set. However, for the second testing set the best clustering solution is provided when k is equal to 77.

The BCA Evaluation. The effectiveness of our proposed technique is evaluated using true positives rate (TPR) which represent the false meta-alerts successfully classified as FPs. TPR is given by:

$$TPR = \frac{\text{number of detected false alerts}}{\text{Real number of false alerts}} \quad (1)$$

Table 4. TPR for testing set 1

	TPR
LOF, K-means and BCA	72.5
k-means, Kmeans and BCA	68.9
AO, K-means and BCA	54.8
NED, K-means and BCA	88.7
SOM with K-means	79.3
DSC	75.3
Rates are given in percentages	

Table 4 illustrates the TPR for the first testing set respectively. As shown the combination of NED, K-means and BCA outperforms concurrent methods for FPs reduction since it generates better results than SOM with K-means and DSC.

In fact, we notice that NED is the best outliers detection approach among the others in the context of intrusion detection.

As illustrated in Table 5 which summarizes TPR for the second testing set, the process AO, k-means and BCA has the lowest TPR for the two testing sets.

In fact, we deduce that AO is not efficient for outliers detection in the context of intrusion detection. Our proposed process NED, K-means and BCA outperforms concurrent methods since it generates the best TPR. In addition, it outperforms the method SOM with k-means and DSC for the second testing set.

Table 5. TPR for testing set 2

	TPR
LOF, K-means and BCA	74.2
k-means, Kmeans and BCA	66.3
AO, K-means and BCA	60.2
NED, K-means and BCA	79.1
SOM with K-means	75
DSC	70.3

Rates are given in percentages

4.3 Time Performance Evaluation

To evaluate the time performance of our proposed process, we compare its running times with competitor methods using the aforementioned testing sets. Table 6 shows the experimental results of running times for the three FPs detection methods. It is clear that the running time of our method is only a little higher than SOM with k-means. However, DSC has the highest running time among all other methods.

Table 6. Experimental results of running times

Methods	Testing set 1	Testing set 2
NED, K-means and BCA	0.22	0.28
DSC	0.35	0.40
SOM with K-means	0.20	0.25

5 Conclusion

An IDS is an essential part of any security package since it ensures the detection of intrusive activities if the information system has been hacked. However, an IDS tends to generate large databases where the majority of detected alerts are false alarms along with many outliers. In this work we propose a two-stage alarm correlation technique to improve the accuracy of an IDS. The aim of the first stage is to remove outliers from the set of alerts and the second one begins by a clustering step to reduce the cardinality of the testing set and ends by the identification of FPs. As our technique is tested using off-line data sets, it will be of interest to extend this work to study alarm correlation for multiple sensors.

References

1. Zhu, B., Ghorbani, A.: Alert correlation for extracting attack strategies. *Int. J. Netw. Secur.* **3**(3), 244–258 (2006)

2. Tjhai, C., Furnell, M., Papadaki, M., Clarck, L.: A preliminary two-stage alarm correlation and filtering system using som neural network and k-means algorithm. *Comput. Secur.* **29**, 712–723 (2010)
3. Bievens, A., Palagiri, C., Szymanski, B., Embrechts, M.: Network-based intrusion detection using neural networks. *Intell. Eng. Syst. Artif. Neural Netw.* **12**, 579–584 (2002)
4. Labib, K., Vemuri, R.: Nsom: A real time network-based intrusion detection system using self-organizing map. In: *Networks Security* (2002)
5. Zhang, Y., Huang, S., Wang, Y.: Ids alert classification model construction using decision support techniques. In: *International Conference on Computer Science and Electronics Engineering*, pp. 301–305 (2012)
6. Gupta, D., Joshi, P.S., Bhattacharjee, A.K., Mundada, R.S.: Ids alerts classification using knowledge-based evaluation. In: *International Conference on Communication Systems and Networks*, pp. 1–8 (2012)
7. Elshoush, H.-T., Osman, I.-M.: An improved framework for intrusion alert correlation. In: *WCE12: Proceedings of the 2012 World Congress on Engineering*, pp. 1–6 (2012)
8. Benferhat, S., Boudjelida, A., Tabia, K., Drias, H.: An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge. *Int. J. Appl. Intell.* **38**(4), 520–540 (2013)
9. Elhag, S., Fernandez, A., Bawakid, A., Alshomrani, S., Herrera, F.: On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems. *Expert Syst. Appl.* **42**, 193–202 (2015)
10. Lin, W.-C., Ke, S.-W., Tsai, C.-F.: Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowl. Based Syst.* **78**, 13–21 (2015)
11. Rousseeuw, P.J., Leroy, A.M.: *Robust regression and outlier detection*. John Wiley & Sons, New York (1987)
12. Abe, N., Zadrozny, B., Langford, J.: Outlier detection by active learning. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 504–509. ACM Press, New York, NY, USA (2006)
13. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Comput. Surv.* **31**(3), 264–323 (1999)
14. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: *Proceedings of the 24th International Conference on Very Large Databases*, New York, NY, pp. 392–403 (1998)
15. Ramaswamy, S., Rastogi, R., Kyuseok, S.: Efficient algorithms for mining outliers from large data sets. In: *Proceedings of the ACM SIDMOD International Conference on Management of Data*, pp. 211–222 (2000)
16. Angiulli, F., Pizzuti, C.: Fast outlier detection in high dimensional spaces. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) *PKDD 2002*. LNCS (LNAI), vol. 2431, pp. 15–27. Springer, Heidelberg (2002)
17. Wu, W.Z., Zhang, W.X.: Neighborhood operator systems and approximations. *Inf. Sci.* **144**, 201–217 (2002)
18. Chen, Y.M., Miao, D.Q., Zhang, H.Y.: Neighborhood outlier detection. *Expert Syst. Appl.* **37**(12), 8745–8749 (2010)
19. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying densitybased local outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, pp. 93–104 (2000)