An Enhanced Intrusion Detection System Based on Clustering

Samarjeet Borah, Ranjit Panigrahi and Anindita Chakraborty

Abstract The aim of a typical intrusion detection framework is to recognize attacks with a high discovery rate and low false alarm rate. Many algorithms have been proposed for detecting intrusions using various soft computing approaches such as self-organizing map (SOM), clustering etc. In this paper, an effort has been made to enhance the intrusion detection algorithm proposed by Nadya et al. The proposed enhancement of the algorithm is done by adding the SOM training process. Clustering of the data is done to differentiate abnormal data from the normal data. The clustered data may sometime contain both normal and abnormal data thus leading to false alarms. In this regard, k-means algorithm is further used to detect those abnormal data and reducing the rate of false positive. The SOM is trained using the neural network toolbox present in Matlab R2010b. The enhanced algorithm yields desired results both in terms of higher detection rates and removal of false positives.

Keywords Intrusion detection · Attack · Clustering · MatLab
False positive · False negative · SOM

1 Introduction

The action of intrusions gradually weakens the confidentiality of resources and information present or to hamper the integrity and availability of behavior in a host or in a network environment. Therefore, intrusions are any sets of activities threatening the veracity, confidentiality, or accessibility of a network resource [1–3]. It can also provide unofficial access to important and useful information and unauthorized file modification which are reasons behind harmful activities.

S. Borah (✉) · R. Panigrahi · A. Chakraborty
Sikkim Manipal Institute of Technology, Sikkim Manipal University,
Rangpo, Sikkim, India
e-mail: samarjeetborah@gmail.com

K. Saeed et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*,
Advances in Intelligent Systems and Computing 564,
https://doi.org/10.1007/978-981-10-6875-1_5

To counter such malicious activities, an intrusion detection system (IDS) comes into action. The IDS identifies these unauthorized activities and takes appropriate action, thus preventing them at real time [4–8]. It is used for detecting real-time monitoring system activities and real-time aggressive behavior, and takes corrective measure to avoid or minimize the occurrence of attacks. The IDS monitors the events that are occurring in the system or networks and analyzes them for intrusion. It collects information related to events taken place in a system and trigger an alarm, when an intrusion is detected, thus preserving the data integrity from attacks. It also helps in handling and monitoring of audit trails, assessment of their system, and networks which is an important part of security management.

This paper discusses the refinement of an existing algorithm [9] which can be used for host-based intrusion detection. In this research work, KDD99 cup dataset is used which consists of both normal and abnormal data and so we first use the k-means clustering algorithm to differentiate the normal from the abnormal data and then train the SOM according to the dataset.

2 Motivation

This work is motivated by the work of El Moussaid et al. [9]. They proposed an improved k-means clustering algorithm. They have tested the algorithm with KDD’99 dataset. Four different types of attacks were identified such as DOS, U2R, R2L, and Probe. In their approach, out of the 41 features of the dataset, 13 features were selected reducing the noise, dataset dimension, and time complexity. Normalization of dataset is done by calculating the mean absolute deviation and standardized measured. The normalized data is then subjected for cluster number initialization. This is done by calculating the similarity and clustering them in one group, while clustering if records remained to be cluster is less than ten percentages of total records is clustered in the same group. This may sometimes lead to a combined cluster of both normal and abnormal data resulting in large number of false-positive alarms. To reduce these alarms, k-means clustering is used by calculating the Euclidean distance and density of each connection record. They have labeled the clusters by calculating the percentage of abnormal connection “ɵ”. If the member of cluster is less than or equal to the product of ɵ, then total records N are labeled as anomaly otherwise as normal. It has been found that the algorithm gives better results for DOS and R2L attacks and the false-positive rate is 30%.

The main advantage of the approach is that it is less time consuming compared to other such approaches. But for Probe and U2R attacks, the rate of detection is low as compared to other attacks. The proposed enhancement of the algorithm is done by adding the SOM training. It tries to reduce the false-positive alarm rate by using k-means clustering.

3 Working Methodology

In the proposed enhancement, the approach consists of several modules, each for different purposes. The modules are described below:

3.1 *Dimensionality Reduction*

The dataset considered is subjected to dimensionality reduction which is done using the Principal Component Analysis (PCA) (Fig. 1).

3.2 *Clustering*

The initial clustering is performed as per the existing method. For each feature of the dataset taken, the minimum and maximum values are found out, and next it helps in finding the upper and lower limits of every feature; if each value of the features has a value between lower and upper limits, then they are grouped into one cluster. Here, one condition is observed; if the amount of connection records is fewer than 10% of the total records of the dataset, then they are grouped into same cluster. In this module, we are trying to define the number of cluster before it is subjected to k-means algorithm. This helps us in getting an idea as to how many clusters a particular dataset may contain. Thus, it may be possible that a particular cluster or more may contain the intrusion information and it may not be necessary to check all the clusters available. The Euclidian distance for each feature is calculated for every connection record. The k-means algorithm is used for the clustering process. The initial centroids are assumed randomly. Distance between the centroid and all the data points are calculated, and the minimum distance is considered resulting as

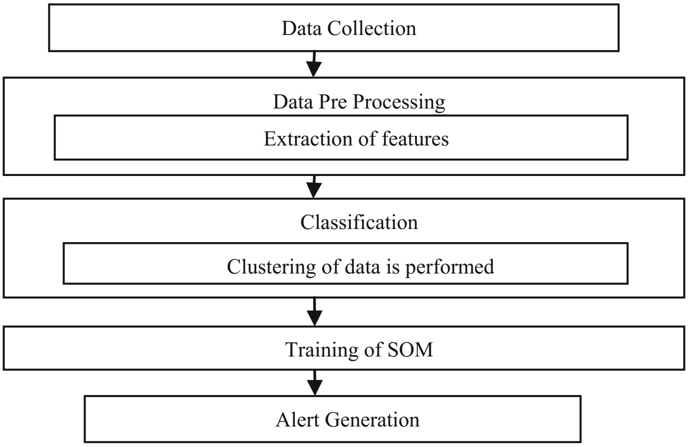

Fig. 1 Architectural diagram

the next centroid leading to cluster updation. This process continues till a maximum repetition is reached defined beforehand [10].

3.3 Training of Self-organizing Map

Self-organizing map is trained using batch algorithm in neural network toolbox of Matlab R2010b. The training runs for maximum of 200 epochs. During training, the weight vector associated with each neuron is moved to the center of the cluster, thus reducing high dimensionality input into two dimensions of topology. This is quite a beneficial tool because after training of SOM there are various options which help us in visualizing the resulting clusters. The various options available are SOM topology, SOM neighbor connections, SOM neighbor distance, SOM weights, SOM sample hits, and SOM weight positions.

4 The Proposed Algorithms

4.1 Data Clustering

```
START
Calculate Max (fi) and Min (fi) of each feature fi.
Calculate upper limit (UL) and lower limit (LL) of
features (fi) as:
    UL= [(Max+Min)/2]
LL= [(Max-Min)/2]
For each feature fi
    if value of LL ≤  fi  ≥ UL
          Insert 1 in table
    Else
          Insert 0 in table
    End if
    For k = 1 to max_row
          For m = k+1 to max_row
                Compare (k, m)
                if similar
                      Keep in same cluster
                Else
                      Keep in different cluster
                End if
          End for
    End for
    If records remained < 10% of total records
          Keep in same cluster
    Else
         Repeat from line 7
    End if
End for
STOP
```

4.2 False-Positive Reduction

```
START
Input_Data= mixed data obtained from clustering of data
in 5.1
Apply K-means clustering algorithm to input data
This step continues till MaxRepeat (defined beforehand)
STOP
```

4.3 Self Organizing Map (SOM) Training

The training phase is having the following steps:

(a) Apply Principal Component Analysis (PCA) for dimentionality reduction.
(b) Consider the score with highest significance.
(c) Train SOM using the batch algorithm.

5 Implementation

The refined algorithm is implemented in Matlab on windows platform. The detection process involves several steps. The preprocessing step is performed prior to the cluster number initialization. The SOM is trained using neural network toolbox. Preprocessing reduces the dimensions of data. The batch algorithm is used for SOM training. Here, the whole training set is trained only once and after that the map is updated having the net effect of the samples. The default training algorithm for SOM is batch algorithm as it is very much faster to calculate than the normal sequential algorithm (in Matlab).

Dataset used for this experiment is KDD’99 cup dataset. A nice description of the same is found in [11, 12]. It has a set of 41 features out of which there are few nominal features. Thus, in order for appropriate clustering, the nominal features are converted into numeric values as follows (Table 1).

Table 1 Transformations table

Name	Value
UDP Protocol	1
ICMP Protocol	2
TCP Protocol	3
Flags	4–16
Services	7–15

6 Results

While comparing, two points are considered. If the remaining records are less than 10% of the total records, they are grouped into the same cluster; the rest of the records were grouped into cluster which results in false-positive alarm generation.

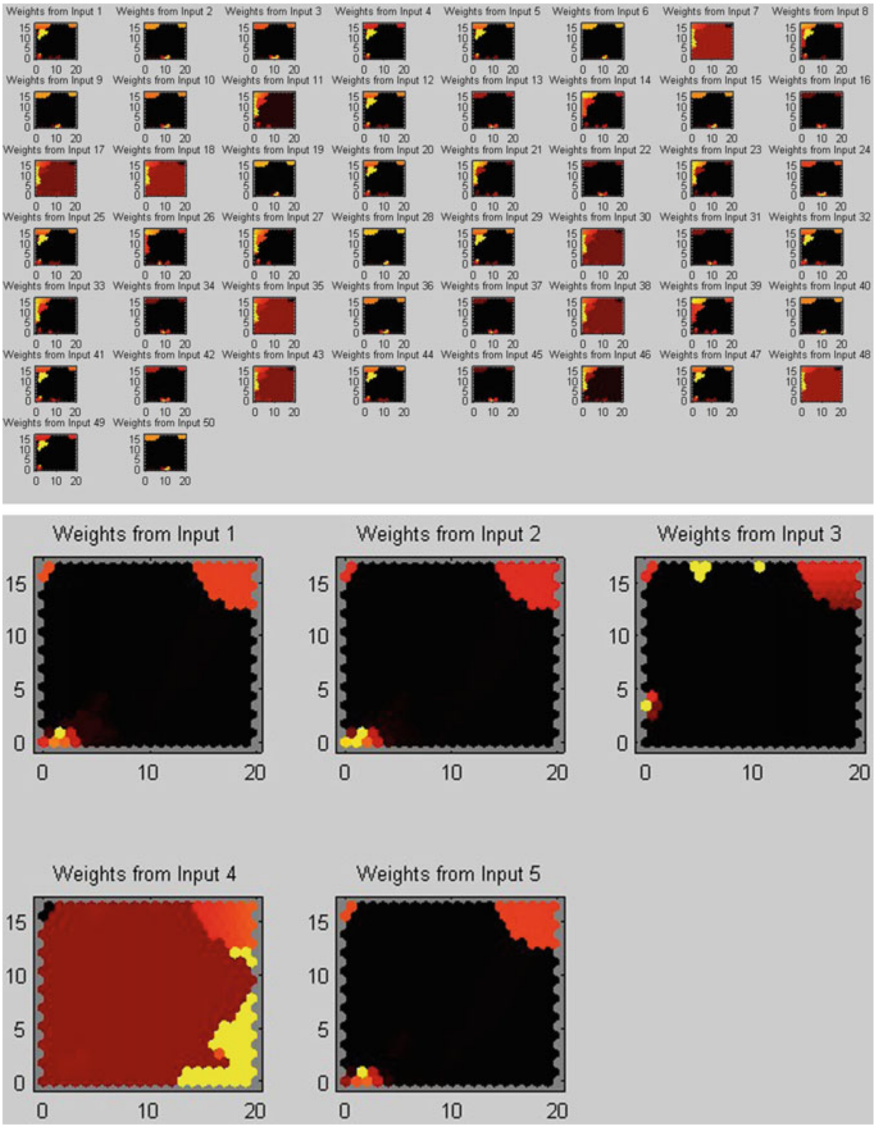

Fig. 2 SOM input planes

K-means clustering is applied to cluster and thus the abnormal data is differentiated from the normal data by clustering the normal data in cluster3 and the abnormal data in cluster1. But still some normal data is still considered as abnormal data, and as in record 4, is normal data but is considered as abnormal data.

The training of the SOM is done using the batch processing algorithm in neural network toolbox.

The SOM input planes show for each input features a weight plane. They are weight visualizations which connect the every input with the each of the neurons. The samples having dark color represent large weight and the light color represent smaller weight. Any two or more samples having same weight may have their color be same. The SOM input plane for the clustering of data and re-clustering using k-means is shown in Fig. 2.

To evaluate the detection rate for host-based intrusion, the dataset considered is clustered by the k-means and is trained using self-organizing map. The algorithm was tested with both the labeled and unlabeled data.

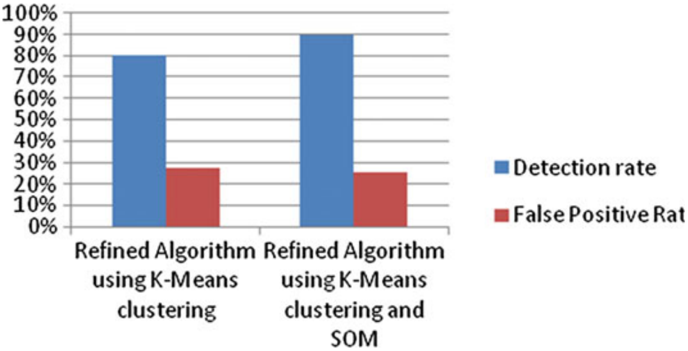

Fig. 3 Detection rate for refined algorithm

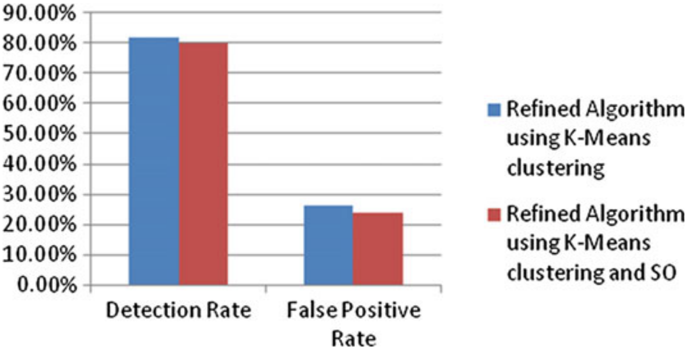

Fig. 4 False-positive rate for refined algorithm

Table 2 Results found for labeled data

For labeled data	Detection rate (%)	False alarm rate (%)
Refined algorithm using K-means clustering	80	27.33
Refined algorithm using K-means clustering and SOM	90	25

Table 3 Results found for unlabelled data

For labeled data	Detection rate (%)	False-positive rate (%)
Refined algorithm using K-means clustering	81.75	26
Refined algorithm using K-means clustering and SOM	80	24

Following figures show the detection rate of the host-based intrusion while being clustered by k-means clustering and trained using labeled data and unlabelled data. Thus, we try to reduce the false-positive alarm rate using this process as shown in figure.

The comparative performance analysis is shown in Figs. 3 and 4, and Tables 2 and 3.

7 Conclusion

The initialization of cluster in the beginning helps us in working with a finite number of clusters. K-means clustering helps in fast computing since the number of clusters is defined beforehand. However, the initialization plays an important role as different values will give different results. The k-means clustering algorithm is able to reduce the percent of false-positive alarm rate.

In this paper, detection rates for k-means clustering with and without SOM training are shown for refined algorithm. The approach leads to the conclusion that when SOM is trained for detecting unlabelled intrusions, it may or may not be able to generate good results. Thus, the SOM must be trained repeatedly using different numbers of nodes until the improved results are found. The false-positive alarm rates can be increased by clustering the data efficiently so that it is able to differentiate correctly between the normal data and abnormal data and the training of SOM must be done accordingly.

References

1. Luo, N., Yuan, F., Zuo, W., He, F., Zhou, Z.: Improved unsupervised anomaly detection algorithm. In: Proceedings of Third International Conference, RSKT 2008, Chengdu, China, 17–19 May 2008. Springer Rough Sets and Knowledge Technology Series (2008)
2. Youssef, A., Emam, A.: Network intrusion detection using data mining and network behaviour analysis. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **3**(6), 87–98 (2011)
3. Suryavanshi, M., Akiwate, B., Gurav, M.: GNP-based fuzzy class-association rule mining in IDS. Int. J. Emerg. Trends Technol. Comput. Sci. (IJETTCS) **2**(6), 179–183 (2013). ISSN 2278-6856
4. Beal, V.: Intrusion Detection (IDS) and Prevention (IPS) Systems. http://www.webopedia.com/DidYouKnow/Computer_Science/intrusion_detection_prevention.asp (2005). Accessed 15 July 2005
5. Kazienko, P., Dorosz, P.: Intrusion Detection Systems (IDS) Part I—(network intrusions; attack symptoms; IDS tasks; and IDS architecture). http://www.systemcomputing.org/ssm10/intrusion_detection_systems_architecture.htm (2003). Accessed 07 Apr 2003
6. Borah, S., Chakravorty, D., Chawhan, C., Saha, A.: Advanced Clustering based Intrusion Detection (ACID) Algorithm, Advances in Computing and Communications, Springer CCIS series, Vol. 192, Part 1, ISSN: 1865:0929, pp. 35–43, (2011) http://dx.doi.org/10.1007/978-3-642-22720-2_4
7. Borah, S., Chakraborty, A.: Towards the Development of an Efficient Intrusion Detection System. Int. J. Comput. Appl. **90**(8), 15–20 (2014)
8. Dutt , I., Borah, S., Maitra, I.: Intrusion Detection System using Artificial Immune System. Int. J. Comput. Appl. **144**(12),19–22 (2016)
9. El Moussaid, N., Toumanari, A., Elazhari, M.: Intrusion detection based on clustering algorithm. Int. J. Electron. Comput. Sci. Eng. **2**(3), 1059–1064. ISSN- 2277-1956
10. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, pp. 281–297. MR 0214227. Zbl 0214.46201 (1967)
11. Olusola, A.A., Oladele, A.S., Abosede, D.O.: Analysis of KDD ’99 intrusion detection dataset for selection of relevance features. In: Proceedings of the World Congress on Engineering and Computer Science 2010 Volume I, WCECS 2010, 20–22 Oct 2010, San Francisco, USA (2010). ISBN: 978-988-17012-0-6, ISSN: 2078-0958
12. Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I.: Selecting features for intrusion detection: a feature relevance analysis on KDD 99. In: Third Annual Conference on Privacy, Security and Trust (PST), 12–14 Oct 2005, The Fairmont Algonquin, St. Andrews, New Brunswick, Canada (2005)