Hybrid Data Mining to Reduce False Positive and False Negative Prediction in Intrusion Detection System

Bala Palanisamy(✉), Biswajit Panja, and Priyanka Meharia

Department of Computer Science, Eastern Michigan University, Ypsilanti, MI 48197, USA
{bpalanis, bpanja, pmeharia}@emich.edu

Abstract. This paper proposes an approach of data mining machine learning methods for reducing the false positive and false negative predictions in existing Intrusion Detection Systems (IDS). It describes our proposal for building a confidential strong intelligent intrusion detection system which can save data and networks from potential attacks, having recognized movement or infringement regularly reported ahead or gathered midway. We have addressed different data mining methodologies and presented some recommended approaches which can be built together to enhance security of the system. The approach will reduce the overhead of administrators, who can be less concerned about the alerts as they have been already classified and filtered with less false positive and false negative alerts. Here we have made use of KDD-99 IDS dataset for details analysis of the procedures and algorithms which can be implemented.

Keywords: Intrusion Detection Systems · Data mining · Intrusion detection
Anomaly detection · SVM · KNN · ANN

1 Introduction

With rapid developments and innovations in computer technology and networks, the number of people using technology to commit cyber-attacks is also increasing. In order to prevent this, we must take preventive measures to stop these crimes and stay secure. A strong computer system can prevent potential attacks by having a good Intrusion Detection System (IDS) in place. Intrusion Detection Systems are used to preserve data availability over the network by detecting patterns of known attacks which are defined by experts. These patterns are usually defined by a set of rules which are validated with a set of common occurring events and probable intrusion sequences. There are many Intrusion Detection Systems available in the market, based on which environment and system they are used for. IDSs can be used in small home networks or in huge organizations which have large systems in multiple locations across the globe. Some of the well-known IDSs are Snort, NetSim, AIDE, Hybrid IDS, Samhain, etc.

The Internet has become an indispensable tool for exchanging information among users and organizations, and security is an essential aspect in this type of

K. Arai et al. (Eds.): FICC 2018, AISC 887, pp. 1–12, 2019.
https://doi.org/10.1007/978-3-030-03405-4_1

communication. IDSs are often used to sniff network packets to provide a better understanding of what is happening in a particular network. Two mainstream preferences for IDSs are (1) host-based IDSs, and (2) network-based IDSs. Correspondingly, the detection methods used in IDS are anomaly based and misuse based (also called signature or knowledge based), each having their own advantages and restrictions. In misuse-based detection, data gathered from the system is compared to a set of rules or patterns, also known as signatures, to describe network attacks. The core difference between these two techniques is that anomaly-based IDS uses collections of data containing examples of normal behavior and builds a model of familiarity, therefore, any action that deviates from the model is considered suspicious and is classified as an intrusion in misuse-based detection, attacks are represented by signatures or patterns. However, this approach does not contribute much in terms of zero-day attack detection. The main issue is how to build permanent signatures that have all the possible variations and non-intrusive activities to lower the false-negative and false-positive alarms.

An Intrusion Detection System screens a computer system or networks for any malignant movement, crime, or illegal violation over the network. Any recognized infringement is regularly reported ahead or gathered midway. It consolidates activities from different sources, and uses alerting methods to distinguish malignant movement from false alarms. One way of building a highly secure system is by using a very good Intrusion Detection System by recognizing Illegal utilization of those systems. These systems work by monitoring strange or suspicious actions that are most likely indicators of crime activity.

Although an extensive variety of approaches are used to secure data in today's composed environment, these systems regularly fail. An early acknowledgment of such occasions is the key for recovering lost or affected data without much adaptable quality issues. As of now, IDSs have numerous false cautions and repetition alerts. Our approach is to reduce the number of false positive and false negative alerts from the Intrusion Detection System by improving the system's proficiency and precision. In such scenarios the global rule set will not be applicable to some of the common usage pattern in some surroundings. Separating approved and unapproved acts is a troublesome issue. Marks pointing to an intrusion may also correspond to approved system use, bringing about false alerts. This is especially troublesome while implementing business sector corresponding Intrusion Detection Systems.

Here we propose the design of an intelligent Intrusion Detection System which makes use of Data Mining techniques to detect and identify the possible threats from the threats suggested by a currently implemented Intrusion Detection System by reducing the False Positive and False Negative alerts. We apply data mining techniques such as Decision Trees, Naïve Bayes Probability classifier, Artificial Neural Networks (ANN), and K-Nearest Neighbor's algorithm (KNN). Data mining is helpful in identifying obtrusive behavior and typical normal behavior.

Currently we can differentiate between normal actions and illegal or strange acts from data mining continuous information updates. The administrators can be less concerned about the alerts as they have been already classified and filtered. We also allow making rapid addition and acquiring of knowledge which will be more helpful in terms of maintenance and operational administration of Intrusion Detection System.

Today's developments in technology have created huge breakthroughs in internet networks which have also increased computer related crime. Thus there is a huge demand for developments in cyber security to preserve our data, detect any security threats and make corrections to enhance the security over the sharing networks. We must make our environments secure with use of secured firewall and antivirus software, and a highly sophisticated Intrusion Detection System. Some of the common types of attacks in a secured network are Probe or Scan, Remote to Local, User to Root, and Denial of Service attacks. Organizations make use of IDSs to preserve data availability from these attacks.

The current Intrusion Detection Systems send a security alert whenever any sequence of events occurs that is similar to the ruleset defined for the common generic environment. Large organizations which use these systems quite often receive numerous alerts which are false alarms, because it is of a generic type, which may be the organization's common and typical work sequence. This means the administrators must validate a large number of sequences. For example: An online marketing company is vulnerable of internet attacks and is also entirely dependent on the live values of the items. Due to this reason they face problem of security alerts raised very often saying the possibility of an attack on network. But the heavy user traffic is a typical part of their operation during starting and end of the day scenario. Having a generic Intrusion Detection System in place causes an overhead for the network administrators due to frequent false positive and false negative alarms and also they do not have an ability to identify new unknown attacks. So there is a need for an Intelligent Intrusion Detection System which can overcome this problem with less false positive and false negative cases.

In this proposed approach we have applied data mining techniques to common generic predictions made by the Intrusion Detection system. This allows us to further analyze the current system and try to classify and cluster our predictions and improve them on the system events. Using such a system will create an intelligent IDS, which can provide more stable predictions and the flexibility of having customizable systems based on specific business sectors of an organizations, allowing it to better identify a possible attack on the system.

In order to build a data mining machine learning system, we must have the data evidently which has clearly labeled or collected data from the Intrusion Detection System. IDSs usually have a huge amount of data available, regarding type of attack, current network traffic, source port, destination port, type of service, region, traffic sequence number, acknowledge number, message length, activity time, etc. This data must be collected on normal working scenarios over a period of time and can then be used to identify events and determine whether they are regular traffic or an actual intrusion and attack on the system.

Data mining operations are done by two approaches in the proposed approach: supervised and unsupervised learning. Supervised learning, also known as classification, is a method that uses training data obtained from observations and measurements classify events. This requires clearly labeled data indicating the operation type and class of severity of the action which triggered event. Unsupervised learning, also known as clustering, is a method used when the training data is unclear and only consisting of observations and measurements. It identifies clusters in data and uses a

technique called clustering to find patterns in the high-dimensional unlabeled data. It is a non-supervised pattern encounter method whenever the data is assembled together with a comparison quantity.

For analyzing our proposed approach, we have used KDD 99 IDS dataset. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. It provided a standard set of data to be audited, which included a wide variety of intrusions simulated in a military network environment. The 1999 KDD intrusion detection contest uses a version of this dataset.

Attacks in the dataset fall into four main categories: DOS: denial-of-service, e.g. syn flood; R2L: unauthorized access from a remote machine, e.g. guessing password; U2R: unauthorized access to local super user (root) privileges, e.g., various "buffer overflow" attacks; Probing: surveillance and other probing, e.g., port scanning.

2 Related Work

Xu et al. [1] discussed privacy preserving data mining (PPDM) in this paper, where the basic idea is to do data mining approach efficiently devoid of conceding data. A protected data like ID card numbers and cellphone numbers which shouldn't be used with mining and results which is disclosed will consequence in confidentiality defilement. Data Provider is the source for the data which is chosen through the data mining task. The provider must handle the sensitive data which some time he may hide. Data collector has major role on protecting sensitive data. They suggest making use of association rule mining or decision tree mining technique to predict the customer buying habits by spending associations between different items that customers or to make random substitutions using decision trees.

Yu et al. [2] proposes approach for the prevention of crimes by predicting the hotspots and time where crime can happen using datamining classification techniques using existing crime data. Classification is made on the data set to identify the hotspots and heating up location in the grid. They are classified as either sociological norms or the social signal. The predictions are made based from the probability of crime incidents that happened in previous month or pattern. The variation of hotspot and cold spot are identified based on the no True positives and False positives & negatives.

Hajian et al. [3] discussed usage of data mining machine techniques effectively on the datasets which are commonly used in intrusion detection systems (IDS) applications by avoiding the data factors in datasets that causes discrimination (gender, age, race, etc.). They introduce anti-discrimination with cyber security and proposes its ideas of its success in judgment prevention with its data feature. Discrimination discovery is the identification of the discriminating factors that influence the decisions and the degree of discrimination which is affected by every factor. So they calculate the probability of occurrence of a decision which is derived as the support and confident factor for each discrimination factors used from the dataset.

Xu et al. [4] concentrates on the identification of malware or any security violation by using data mining technique for mobile application. They propose solution to the problem with a cloud computing platform implemented with data mining which analyses the android apps with (ASEF) which is an automated tool which works as a virtual machine and can install applications on it and (SAAF) analysis method which provides information on applications. They have described ways to static and dynamic behavior analysis patterns of applications and then they apply machine learning technique to analyze and classify the Android software by monitoring things happening on kernel level. They have used PART, Prism and nNb classifiers to identify the software either it is original or malicious.

3 Data Normalization and Pre Processing

Data from real-time systems will often be mostly unformatted, and in case of network data the data usually consists of multiple formats and ranges. It is therefore necessary to convert the data into a uniform range, which can help the classifier to make better predictions. Data cleaning is required to filter accurate information from invalid or unimportant data and distinguish important data fields from insignificant ones [5]. In cases where organizations use multiple systems we must remove the redundant data from the data set consolidated from the different systems. Missing value estimation must also be performed for some of the fields which are not available. In many cases, when we collect the data from the Intrusion Detection System we will have certain data fields missing based on the particular sequence of events or scenarios. Missing value estimation can be used to estimate probable values for those missing fields; however information sections with excessively numerous missing qualities are probably not going to convey much helpful data. Subsequently, information sections with a number of missing qualities that exceeds a given limit can be expelled. The more stringent this limit is, the more data that is expelled.

The KDD Cup99 dataset available in three different files, the KDD Full Dataset which contains 4,898,431 instances, the KDD Cup 10% dataset which contains 494,021 instances, and the KDD Corrected dataset which contains 311,029 instances.

Each sample of the dataset represents a connection between two network hosts according to network protocols. The connection is described by 41 attributes, 38 of which are continuous or discrete numerical quantities and 3 of which are categorical qualities. Each sample is labeled as either a normal sequence or a specific attack. The dataset contains 23 class labels, 1 for normally occurring events and the remainder for different attacks. The total 22 attack labels fall into the four attack categories: DOS, R2L, U2R, and Probing.

Preprocessing this data consists of two steps, removing duplicate records and normalizing the data. In the first step, we removed 3,823,439 duplicate records from the 4,898,431 instances of the KDD Full Dataset, leaving 1,074,992 distinct records. Similarly, 348,435 duplicate records were removed from the KDD Cup 10% dataset's 494,021 instances, obtaining 145,586 distinct records, a reduction of about 70%.

Data Normalization was performed by performing substitution for the columns pertaining to protocol type, service, flag, and land. We replaced the different types of

attack classes with the 4 main classes, DOS, R2L, U2R and probing. We then performed min–max normalization based on the values in respective columns, as the data was discontinuous with different ranges for each column. The attributes were then scaled to a range of 0–1.

4 Feature Selection

We currently have 41 features in KDD Dataset [9, 10] which are to be studied and selected to obtain better performance by reducing the false positive and false negative errors in the prediction and improve the accuracy. We can identify the most important features and other least important features to improve our accuracy and prediction. We first noted that several columns, including Num_outbound_cmds, and Is_hot_login, were zero valued for all the records, so these columns could be omitted when performing the data analysis.

We also studied the entropy of each column with a list of features for which the class is most relevant, and selected the important features which were good indicators for identifying the class of the attacks.

5 Dimension Reduction of Data

Having a large dataset makes predictability an issue as many of the data fields have different scales of variability. Hence, in order to make the parameter rescaling easier we make use of dimensional reduction. Figure 1 provides architecture of network intrusion detection system.

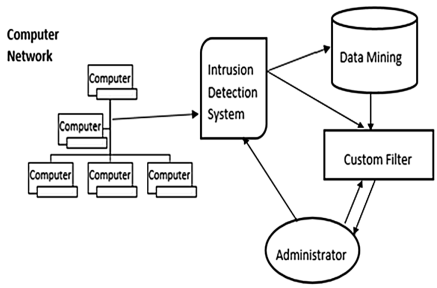

Fig. 1. Network intrusion detection system data mining process diagram.

5.1 Principal Component Analysis

Dimensional reduction is a measurable methodology that orthogonally changes the first n directions of an information set into another arrangement of n directions. As an aftereffect of the change [6], the primary part has the biggest conceivable fluctuation of values each succeeding segment has the most elevated conceivable difference under the

limitation that it is orthogonal to the previous segments. Keeping just the main m < n segments diminishes the information dimensionality while holding the vast majority of the variety in the information.

5.2 Random Forests/Ensemble Trees

Random forests/Ensemble trees are one of the possible ways to deal with dimensional reduction. It is implemented by reducing a vast decision tree, built in arrangement of trees against an objective trait and based on every nodes utilization measurements to locate the most instructive subset of components. In particular, we can produce a huge set of decision trees, with every tree being prepared on a little portion of the aggregate number of qualities. On the off chance that a trait is frequently chosen as best part, it is in all probability an instructive element to hold. A score computed on the characteristic utilization insights in the irregular branch which shows in respect to alternative qualities – which are the most prescient properties.

Once the preparation of the data is finished we can apply the supervised classification.

This paper proposes to use well known data mining classification algorithms such as Artificial neural network [7–9] (ANN), Decision tree learning (DT), Naïve Bayes classifier(NBC), Support vector machines (SVM), Nearest Neighbor Algorithm (KNN). There are many significant reasons for combining multiple methods in our analysis, mainly it provides a better performance and prediction supported by several methods will give us better accuracy and will reduce false positive and false negative cases.

After we have specific data we need to separate the data sets into the training dataset and the testing dataset. Training datasets are used for learning the patterns of the system to classify or cluster the data, and testing datasets are used to test the accuracy of the system which we have trained.

Step 1: Convert the symbolic attributes protocol, service, and flag to numerical ones.
Step 2: Normalize data to [0, 1]
Step 3: Separate the instances of 10% KDD training dataset into five categories: Normal, DoS, Probe, R2L, and U2R.
Step 4: Apply modified K - means on each category and create new training datasets.
Step 5: Train SVM and others with these new training datasets.
Step 6: Test model with corrected KDD dataset.
TN (True Negatives): Indicates the number of normal events successfully labeled as normal.
FP (False Positives): Refer to the number of normal events being predicted as attacks.
FN (False Negatives): The number of attack events incorrectly predicted as normal.
TP (True Positives): The number of attack events correctly predicted as attack.

6 Classification Process

6.1 Naïve Bayesian Classifier

Naïve Bayesian classifier uses the Bayes theorem to apply the posterior probability of one event on other. It accepts that the impact of a probability on a given class is autonomous of the estimations of different characteristics. Given a case, a Bayesian classifier can anticipate the likelihood that the tuple has a place with a specific class. Bayesian system or network system is a probability based graphical approach that speaks to the factors and the connections between each of the classes. The system is developed with hubs representing the different or constant arbitrary factors and coordinated classes as the connections within them, setting up a coordinated non-cyclic diagram. Bayesian systems are fabricated utilizing master learning or utilizing productive calculations that perform surmising.

When we use the Naive Bayes, we must assume the dataset fields and features are independent for each event and it is the most accurate because it is based on the probability and can be represented as Bayesian network by combining graph theory and probability. Bayesian networks can be used for learning the inference between the real time intrusion scenarios and common regular scenarios.

6.2 Support Vector Machine

Support Vector Machine (SVM) is a well-known classification method. SVM utilizes a nonlinear mapping to change the first train data into a higher measurement. Inside this new measurement, SVM hunt down a direct ideal isolating class distribution by utilizing support vectors and edges. The SVM is a classifier in light of finding an isolating class distribution in the component space between two classes in a manner that the separation between the class distribution and the nearest information purposes of every class is improved.

The approach depends on a minimized classification chance instead of on ideal order. SVMs are outstanding for their speculation capacity and are especially valuable when the quantity of components, m is high and the quantity of information focuses, n is low (the value of m greater than n). At the point when the two entities are not distinguishable, loose factors are included and a cost feature is relegated for the covering information focuses. SVM among quick calculations is not stable when the quantity of qualities is high.

We need to perform SVM classification training by selecting the features such that it maximizes the prediction, picking the feature which is mutual from the attack case and common case.

Here we have used a subset of KDD Dataset after multiple levels of normalization we have now classified using the complete dataset and we have now able to obtain a 96% accuracy in prediction of dataset classes as normal or an intrusion scenario.

6.3 Artificial Neural Networks

Artificial Neural Networks is a well-known methodology based on how the neurons in our brain works. It consists of interconnected fake neurons equipped for specific calculations on their information sources. The data information actuates the neuron network nodes in the main layer of the system whose yield is passed to another node of neurons in the system. Every layer transfers its yield to the next node and the final node yields the outcome. Nodes in the middle of the info and yield nodes are alluded to as concealed layers. At the point which an Artificial Neural Network is utilized as a classifier, the yield nodes produce the last order classification.

ANN can create nonlinear models. The back-engendering highlight of the ANN makes it conceivable to create an EX-OR rationale. When modifying in this node with intermittent, feed forward, and convolutional NNs, ANNs are picking up in ubiquity once more, and in the meantime winning many leads in late scenarios acknowledging the. Since the propelled adaptations of ANNs require much additionally handling power, they are actualized ordinarily on graph making units.

When ANN was applied to the 10% KDD Dataset, we were able to obtain an accuracy of 94% when trained with the complete dataset.

6.4 K-Nearest Neighbours

KNN is a non-parametric classification algorithm which makes use of regression. It attributes the closest neighbors grouping, the yield is a particular class classification. KNN is example based learning, or slow realizing, where the capacity is just approximated locally and all calculation is conceded until characterization.

The k-NN calculation [11, 12] is a relatively simple machine learning calculation. In the training phase, it saves only the vectors and the field types, using the most frequent data to influence the major decisions in this algorithm. This algorithm can be varied by selecting different values for k, the neighbor limit number to a particular sequence of data.

When KNN was applied to the 10% KDD Dataset, we were able to obtain an accuracy of 70% when trained with the complete dataset with parameters k = 5.

6.5 Decision Trees

A decision tree is a simple algorithm which resembles a tree and branches in its connections of elements and groupings. A model is ordered by testing its quality values against the branches of the tree. At a particular node when constructing the decision tree and at every hub of the tree, we must pick the appropriate element that is most successfully suitable in its arrangement of cases into sub nodes. The leaf node is the standardized data and where the quantity with the most elevated standardized data pick up is settled on the root. The decision trees are the common representation where we can denote high order precision, and straightforward execution. The fundamental hindrance is that for information incorporating clear cut factors with an alternate number of levels, data pick up qualities are one-sided for elements with high density of

branches. The decision tree is implemented by improving the data reachability at every branch, results in data rank hierarchy.

Once the data is classified, we must compare the results. Our proposal is to have a comparison graphs with the Accuracy, Classification limits, Time taken for the Model which is known and Time taken for model that is unknown sequence of data.

Several models have been proposed to design multi-level IDS. Some models classified DoS and Probe attacks at the first level, Normal category at the second level, and both R2L and U2R categories at the third or last level. By contrast, other models classified Normal, DoS, and Probe at the first level and R2L and U2R at the second level. Figure 2 shows the scattered plot and Fig. 3 parallel coordinate plots. Table 1 show that SVM is more accurate than KNN and ANN.

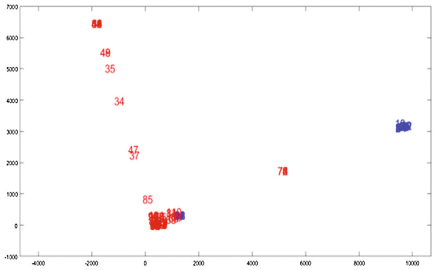

Fig. 2. Scatter plot of results of KDD 10% Dataset where normal sets are in blue color and intrusion sets are in red color.

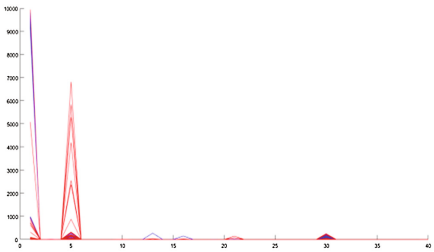

Fig. 3. Parallel coordinate plot of data value distribution of 39 different features.

Table 1. Accuracy of KNN, ANN, SVM

KDD 10% Dataset Results with each algorithm			
	KNN	ANN	SVM
Accuracy	70%	94%	96%

7 Conclusion

In this paper we have discussed data mining techniques which all we can be applied to build a strong and intelligent Intrusion Detection System. We have mainly targeted data mining procedures to reduce false positive and false negative alerts, which will reduce the overhead of a network administrator in an organization. It is also a very difficult to collect the training dataset which is a crucial task for building the system. Even though our methodology has not been proved for cybercrimes and attacks, this approach will help us to build custom sector based Intrusion Detection System rather than having a generic system which can be very confusing to handle.

The ideas which we have presented are introductory and in progress, so there must be significant investment of further work to detail procedures on how to normalize the data and how unrelated details of some of the specific attack scenarios can be filtered. There are also additional methods to reduce the dimensions of the data that can be explored. Sometimes it also depends on the type of intrusion detection systems as they have a variety of formats that they can produce as the report and we also have to see how we can extract the report to a data set which can used for mining. The classification methods Naïve Bayes and Decision Trees will produce a straight forward results though in terms of other methods K-Nearest Neighbors we have a difficulty of selecting a proper k value which may have a lots of testing cases to be run to identify the right value similar difficulties will also be for support vector and artificial neural network method.

References

1. Xu, L., Jiang, C., Wang, J., Yuan, J., Ren, Y.: Information security in big data: privacy and data mining. IEEE Access **2**, 1149–1176 (2014)
2. Yu, C.H., Ward, M.W., Morabito, M., Ding, W.: Crime forecasting using data mining techniques. In: 2011 IEEE 11th International Conference on Data Mining Workshops, pp. 779–786. IEEE (2011)
3. Hajian, S., Domingo-Ferrer, J., Martinez-Balleste, A.: Discrimination prevention in data mining for intrusion and crime detection. In: 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 47–54. IEEE (2011)
4. Xu, J., Yu, Y., Chen, Z., Cao, B., Dong, W., Guo, Y., Cao, J.: Mobsafe: cloud computing based forensic analysis for massive mobile applications using data mining. Tsinghua Sci. Technol. **18**(4), 418–427 (2013)
5. Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C.: Data mining for credit card fraud: a comparative study. Decis. Support Syst. **50**(3), 602–613 (2011)
6. Hu, Y., Panda, B.: A data mining approach for database intrusion detection. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 711–716. ACM (2004)
7. Ravisankar, P., Ravi, V., Rao, G.R., Bose, I.: Detection of financial statement fraud and feature selection using data mining techniques. Decis. Support Syst. **50**(2), 491–500 (2011)
8. Erskine, J.R., Peterson, G.L., Mullins, B.E., Grimaila, M.R.: Developing cyberspace data understanding: using CRISP-DM for host-based IDS feature mining. In: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, p. 74. ACM (2010)

9. Lee, W., Stolfo, S.J., Mok, K.W.: A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999, pp. 120–132. IEEE (1999)
10. Feng, W., Zhang, Q., Hu, G., Huang, J.X.: Mining network data for intrusion detection through combining SVMs with ant colony networks. Future Gener. Comput. Syst. **37**, 127–140 (2014)
11. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001) (2001)
12. Julisch, K., Dacier, M.: Mining intrusion detection alarms for actionable knowledge. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 366–375. ACM (2002)