A Novel Algorithm for Network Anomaly Detection Using Adaptive Machine Learning

D. Ashok Kumar and S. R. Venugopalan

Abstract Threats on the Internet are posting high risk to information security and network anomaly detection has become an important issue/area in information security. Data mining algorithms are used to find patterns and characteristic rules in huge data and this is very much used in Network Anomaly Detection System (NADS). Network traffic has several attributes of qualitative and quantitative nature, which needs to be treated/normalized differently. In general, a model is built with the existing data and the system is trained with the model and then used to detect intrusions. The major and important issue with such NADS is that the network traffic changes over time; in such cases, the system should get trained automatically or retrained. This paper presents an adaptive algorithm that gets trained according to the network traffic. The presented algorithm is tested with Kyoto University's 2006+ Benchmark dataset. It can be observed that the results of the proposed algorithm outperform all the known/commonly used classifiers and are very much suitable for network anomaly detection.

Keywords Intrusion • Anomaly • Network traffic • Normalization
Performance metrics • Adaptive algorithm • Kyoto 2006+ • Naïve Bayes classification

1 Introduction

Internet has brought huge potential for business and on the other hand, it poses lots of risk for the business. Internet is a global public network [1]. Intrusion is a deliberate, unauthorized, illegal attempt to access, manipulate, or take possession of

D. Ashok Kumar (✉)
Department of Computer Science, Govt. Arts College, Tiruchirappalli, Tamilnadu, India
e-mail: akudaiyar@yahoo.com

S. R. Venugopalan
Aeronautical Development Agency (Ministry of Defence, GoI), Bengaluru 560017, India
e-mail: venu_srv@yahoo.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_7

information system to render them unreliable or unusable. Intrusion detection is the process of identifying various events occurring in a system/network and analyzing them for the possible presence of intrusion. Intrusion Detection Systems (IDS) can be classified into three types based on the method on which intrusion is detected namely signature-based, anomaly-based, and hybrid. Statistical methods and clustering are used for anomaly detection systems [1]. The availability of higher bandwidth and sophisticated hardware and software, the need to detect intrusions in real-time, and the adaptation of the detection algorithm to the ever-changing traffic pattern are a big challenge. IDS should adapt to the traffic behaviors and learn automatically. In this paper, an algorithm is proposed for network anomaly detection. The results, i.e., performance metrics of the experiment, are encouraging. The proposed algorithm can detect new/unknown attacks and can learn and adapt automatically based on the network traffic.

The organization of the paper is as follows: Sect. 2 gives the background and the literature surrounding IDS with necessary performance metrics. The problem description and the algorithm development are discussed in Sect. 3. In Sect. 4, the dataset, data preprocessing, data normalization, and the training and test dataset generation used in this study are discussed. The experiment and the results are discussed in Sect. 5. Conclusions and future work are in given in Sect. 6.

2 Background and Related Work

Panda M. et al. proposed Naïve Bayes for Network Intrusion Detection and found that the performance of Naïve Bayes is better in terms of false-positive rate, cost, and computational time for KDD ’99 datasets, and same was compared with backpropagation neural networks approach [2]. Jain et al. in their work have combined information gain with Naïve Bayes for improving the attack detection and have observed higher detection rate and reduced false alarm [3]. Muda Z. et al. in their work have used k-means to cluster the data and used Naïve Bayes classifier to classify the KDD Cup99 [4] data and have achieved better performance than Naïve Bayes classifier [5]. They have achieved 99.7% accuracy, a detection rate of 99.8%, and 0.5 false alarm rate.

FVBRM model is proposed by the authors of [6] for feature selection and compared it with other selection methods by reducing the features of the dataset and then classifying with Naive Bayes classifier. There is no mention about how the qualitative and quantitative attributes are treated. The authors of [7] have compared the results of Naïve Bayes algorithm with decision tree and concluded that from the performance point of view Naïve Bayes provides competitive results for KDD 99 [8] dataset. K-means clustering algorithm was applied for intrusion detection and concluded that k-means method is very efficient in partitioning huge dataset and has better global search ability [9, 10]. K-means clustering is a good unsupervised algorithm used to find out structured patterns in the data but the computational complexity is high for its application in intrusion detection. A novel density based k-means cluster was proposed for signature-based intrusion detection [11] where results show improved accuracy and detection rate with reduced false-positive rate. It is not very clear that which normalization technique was used and how the discrete and continuous data was treated. Sharma et al. [12] proposed k-means clustering via Naïve Bayes for KDD Cup '99 dataset. This approach outperforms the Naïve Bayes in terms of detection rate and higher false positives which is a concern.

S.M. Hussein et al. in their work compared the performance of Naïve Bayes, Bayes Net, and J48graft, and recorded that Naïve Bayes performs better in terms of rate of detection and time to build model, whereas J48 was better in terms of false alarm rate [13]. Earlier works which were reviewed in this section tried in achieving higher performance with the help of preprocessing/feature reduction and have achieved performance improvements. The study of the existing literature reveals the need for a novel algorithm to detect unknown attacks because they have not considered the following points: (a) Ever-changing network traffic/speed, new attacks, and the need for the algorithm to adapt itself and learn/get trained automatically from the changing traffic; (b) The ability of the algorithms/methods described in the literature to perform well for datasets other than the tested ones. The algorithms were tested with the only one dataset; (c) Either attack or normal data is used for training and not both; (d) Network traffic data contains features that are qualitative or quantitative nature and has to be treated differently and have to use different preprocessing/normalization technique; and (e) Earlier works have measured accuracy, detection rate, and false alarm rate only as a performance measure which may not be sufficient; measures such as F-score and sensitivity are required for evaluating an algorithm/method.

2.1 Metrics for Intrusion Detection Performance

The choice of NADS for a particular environment is a general problem, represented precisely as intrusion detection system's evaluation [14]. For an anomaly detection system, False Alarm Rate (FAR) and the Detection Rate (DR) are basic factors and their trade-off can be analyzed with Receiver Operating Characteristic (ROC) curve. The above-mentioned basic factors FAR and DR are not sufficient to evaluate the performance of IDS [15]. So the evaluation of IDS should take into account the environment where the IDS is being deployed, its maintenance costs, operating environments, likelihood of attacks, cost toward false alarm and missed detections, etc. [14]. The following section explains the performance metrics, which needs to be considered while deploying/deciding on IDS/anomaly detection system and these measures are used for evaluation of the algorithm proposed. Attacks that are detected correctly as attacks are referred as True Positives (TP) and normal connections detected as normal connections are True Negatives (TN). The following Table 1 is the general confusion matrix used in intrusion detection evaluation. The values in the matrix represent the performance of the prediction algorithm. TP rate determines the security requirement and the number of FP's determines the

Table 1 Confusion matrix

Confusion matrix		Predicted value	
		Attack	Normal
Actual value	Attack	True Positives (TP)	False Negatives (FN)
	Normal	False Positives (FP)	True Negatives (TN)

Table 2 Performance measures used to evaluate IDS

S. no.	Performance metric	Description	Formula
1.	Detection Rate/Positive Prediction Value/Precision	Proportion of the predicted positives which are actual positive (or) fraction of test data detected as attack which is actually an attack	(TP + FP)
2.	Accuracy	Measure to test the overall accuracy. It can be delineated as the percentage of correct prediction among the whole dataset	(TP + FP + FN + TN)
3.	False Alarm Rate	False-positive rate (FPR) also known as false alarm rate (FAR) refers to the proportion of normal packets being falsely detected as malicious	(FP + TN)
4.	Sensitivity/True Positive Rate/Recall	The fraction of attack class which is correctly detected (or) proportion of actual positives which are predicted as positives	(TP + FN)

usability of the IDS. There is always a trade-off between the two metrics, precision and recall. For an IDS to be effective, the FP and FN rates should be minimized and accuracy, and TP and TN rates to be maximized [16].

Table 2 gives the details about the various performance measures for the evaluation of IDS.

F-Score

The harmonic mean between precision and recall is called as F-score/F-measure. F-score is considered as a measure of the accuracy of a test. Good IDS performance is achieved by improving both precision and recall. Both precision and recall are considered for computing F-score. An F-score of 1 is considered as best and 0 as worst:

$$F - \mathrm{Score} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{1}$$

3 Problem Description and Approach/Algorithm Development

Supervised algorithms significantly outperform unsupervised algorithms in detecting known attacks. For those problems where the test data is drawn from different distributions, semi-supervised learning methods offer a promising future [17]. The dramatic increase in the speed of the networks has made the existing policies and network anomaly intrusions detection systems vulnerable to intrusion than ever before. Thus, making the existing IDS useless unless they adapt to the new trends, i.e., adapt to the ever-changing network traffic and learn automatically. Adaptive Network Anomaly Detection Algorithm (ANADA) proposed in this study uses labeled dataset for initial learning and adapts itself to the changing traffic patterns. The proposed ANADA algorithm used simple statistical measures such as mean, median, and norm (distance measure). This algorithm uses normalized data, i.e., the normalization of training data is described in the data preprocessing section. The uniqueness of the algorithm is given below:

- The algorithm uses both attack and normal data for training;
- The algorithm adapts itself the new traffic by modifying the training dataset with the test dataset;
- At each test instance, the algorithm decides whether the test data is worth being included/replaced with an instance of training data;
- The algorithm is very simple and can be easily parallelized for performance improvements; and
- This algorithm uses a new distance measure, i.e., 0.8 norm (given in Eq. 2).

3.1 Adaptive Network Anomaly Detection Algorithm (ANADA)

Input: Training dataset and testing dataset: a—attack training dataset; n—normal training dataset; and t—testing dataset.

Output: Anomaly detection performance metrics such as detection rate, FAR, sensitivity, F-score, etc.

Generate initial population/training dataset that has equal number (5000) of attacks and normal traffic features.

Training Phase: The training dataset is grouped based on the label as attack and normal sessions. 5000 attack records and 5000 normal records are used for training. Find the centroid of the attack class and normal class. For numerical attributes, the mean (or) average is calculated and for the categorical attributes, median is calculated. The centroid will be a set of values.

Testing Phase: For each record in the testing data, the following steps are followed:

BEGIN

1. *Initialize the necessary variables such as counters and loop index etc.*
2. *Read the attack and normal traffic data. // attack data is referred as a[5000] [7] and normal data as n[5000] [7].*
3. *Evaluate mean for first 12 attributes and median for next 2 attributes for both attack and normal data //ma referred as mean of train attack data and mn referred as mean of train normal data.*
4. *Read the test data // test data is referred as t[5000] [7] 15th column is the actual label and 16th column will be used for computed label.*
5. *Compute the distance between the test data and the centroid of the attack/normal dataset using 0.8-norm as given in Eq.* 2:

$$|X| = \sqrt[0.8]{\sum_{k=1}^{n} |ai - ti|^{0.8}} \qquad (2)$$

6. *If the test data is closer to normal centroid and the distance between test data and normal centroid is less than 1.5 times of the distance between the normal and attack centroid, then it is labeled as normal else an attack.*
7. *After labeling the test data, decision has to be made whether to replace the test data with the training data.*
8. *If the new test data is attack/normal, the decision has to be made whether the new data has to be replaced with the attack/normal training data or not. This is done by calculating the distance between the test data and the attack/normal centroid and the ith (counter used for replacement) row of attack data and the centroid of the attack/normal. The distance is calculated using 0.8-norm as given in Eq.* 2. *If the new test data is closer to the centroid than the ith data, then replace the ith data with the new one.*
9. *Repeat the above steps for all the test data. The algorithm is given in the next Sect.* 3.1.
10. *Calculate the TP, TN, FP, FN, sensitivity, specificity, FAR, accuracy, detection rate, F-score, etc.*

END *//end of algorithm.*

4 Datasets for Experimentation

In this paper, the publicly available dataset Kyoto 2006+ datasets is used for experimentation.

4.1 Kyoto 2006+ Dataset

Kyoto 2006+ [18] dataset is a network intrusion evaluation/detection dataset which was collected from various honeypots from November 2006 to August 2009. Real network traffic traces were captured in this dataset. This data has 24 statistical features, which include 14 features which were there in KDDCUP '99 dataset and additional ten features for effective investigation. This study uses August 31, 2009, data and has used the first 14 features (conventional features) and the label which indicates whether the record is an attack or normal. As the study does not distinguish between the known and unknown attack, both are represented as attack only. The unknown attacks in this dataset are very minimal and that is also another reason for not distinguishing known and unknown attack.

4.2 Data Preprocessing

Raw data needs to be preprocessed before fed into any learning model and the most used technique is normalization [19]. Network traffic data contains features that are qualitative or quantitative nature and have to be treated differently. The values of attributes with high values can dominate the results than the attributes with lower values [20]. The dominance can be reduced by the process of normalization, i.e., scaling the values within certain range. The quantitative attributes can be normalized by various techniques, such as (1) mean-range normalization, (2) frequency normalization, (3) maximize normalization, (4) rational normalization, (5) ordinal normalization, (6) softmax scaling [21], and (7) statistical normalization, whereas applying the above normalization techniques for qualitative data will not be meaningful. For qualitative data, the general approach is to replace the values with numerical values. Though this seems simpler, it does not consider the semantics of the qualitative attributes. In this study, the following probability function is used for normalizing the qualitative data [2, 20]:

$$\mathrm{fx}\ (\mathrm{x}) = \mathrm{Pr}\ (\mathrm{X} = \mathrm{x}) = \mathrm{Pr}(\{\mathrm{s} \in \mathrm{S}: \mathrm{X}(\mathrm{s}) = \mathrm{x}\}) \tag{3}$$

Based on the above equation, the qualitative data are converted into quantitative data in the range of [0–1]. In this study, for quantitative attributes, mean-range normalization is used [22]:

$$Xi = \frac{(vi) - \min(vi)}{\max(vi) - \min(vi)} \tag{4}$$

The reason for choosing the mean range (for quantitative attributes) and probability function (for qualitative attributes) is because this normalization technique yields better results in terms of time and classification rate [2 and 8]. There are two qualitative attributes, i.e., flag and service; and all the other 12 attributes are quantitative. The mean-range normalization is applied for quantitative attributes and the above probability function is used for qualitative attributes.

4.3 Dataset Generation for Training and Testing

The framework used in the study uses both normal and malicious (attack) data for training. In general, the system is trained using either normal data or attack data. This is one of the unique characteristics of the algorithm which makes it suitable for adaptive learning, i.e., the system is automatically trained based on the testing/network traffic data. The data pertaining to date August 31, 2009 of Kyoto 2006+ dataset is used for this study and this dataset has 134665 records, out of which 44257 (32.9%) are normal and the 90408 (67.1%) are attack data records. There were a lot of duplicate records (42.2%) which were removed before the experimentation. *From the above statistical information, it can be observed that the attack data dominates the dataset which is not a general case and there are a lot of duplicates.*

In this study, the procedure was devised in selecting the testing/training data in such a way that the above observations do not dominate the detection procedure and this can be used for all the datasets. In this study, the training dataset consists of 5000 attack and 5000 normal records. Four sets of testing records were generated in the following manner for Kyoto 2006+ dataset. These records were chosen in random using SPSS Statistics V20 after removing the duplicates.

Dataset1 (Test Case-1) consists of 10000 records of which 10% are attack and the rest 90% are normal records.

Dataset2 (Test Case-2) consists of 10000 records of which 20% are attack and the rest 80% are normal records.

Dataset3 (Test Case-3) consists of 20000 records of which 10% are attack and the rest 90% are normal records.

Dataset3 (Test Case-4) consists of 20000 records of which 20% are attack and the rest 80% are normal records.

The reason for choosing the above configuration was that in general, the number of attacks will not be more than 20% of the records.

5 Experimental Results and Discussions

The ANADA described earlier in this study is implemented using Matlab version 7.12.0.635 (R2011a). The experiments were carried out on a system with Intel Core i3 2.53 Ghz CPU and 4 GB memory running Window 8 Professional 64-bit Operating System. Microsoft Office Professional Plus 2010 & SPSS Statistics V20 were used for data preprocessing.

Table 3 IDS performance comparison of ANADA with Naïve Bayes (Kyoto 2006+)

Kyoto dataset		Detection rate	Accuracy	False alarm rate	F-score
Test Case-1	NB Train and Test	0.7229	0.9616	0.0426	0.8388
	NB 10-Fold	0.7223	0.9615	0.0423	0.8380
	ANADA	0.8861	0.9773	0.0127	0.8866
Test Case-2	NB Train and Test	0.8499	0.9646	0.0441	0.9187
	NB 10-Fold	0.8512	0.9642	0.0435	0.9175
	ANADA	0.9402	0.9750	0.0149	0.9373
Test Case-3	NB Train and Test	0.7244	0.9619	0.0422	0.8398
	NB 10-Fold	0.7266	0.9621	0.0417	0.8404
	ANADA	0.8085	0.9727	0.0251	0.8744
Test Case-4	NB Train and Test	0.8484	0.9641	0.0446	0.9176
	NB 10-Fold	0.8525	0.9644	0.0430	0.9178
	ANADA	0.9336	0.9666	0.0159	0.9148

Kyoto 2006 dataset is preprocessed as given above and the training data was fed to the algorithm for learning. There are four test cases namely test-case1, test-case2, etc. The test cases are fed one by one and the results are recorded. The results are given in Table 3 and Fig. 1. Table 3 clearly depicts the various anomaly detection evaluation performance measures of ANADA algorithm for Kyoto 2006+ dataset. The results need to be compared with the other techniques. Naïve Bayes classification was used because of the reason that it is a simple classification scheme and

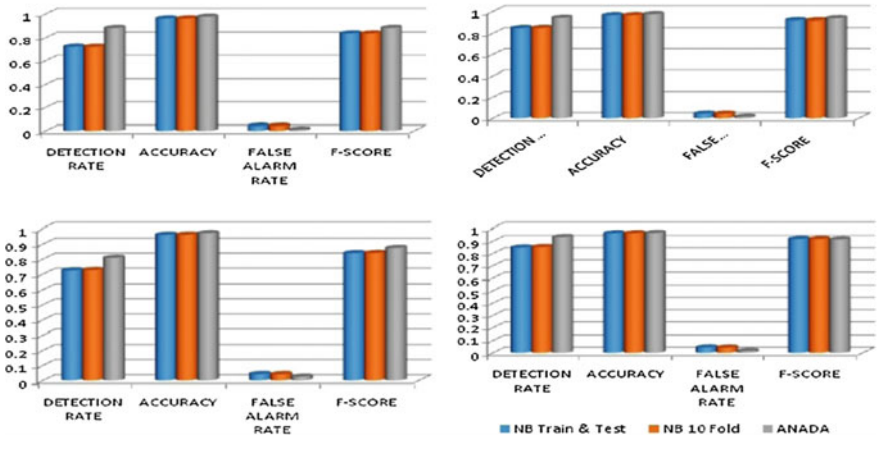

Fig. 1 Performance comparison of ANADA with NB and NB 10 Fold (Kyoto 2006+ dataset)

provides better results in terms of detection rate and FAR. Naïve Bayes is a supervised algorithm based on Bayes' theorem with the "Naïve" assumption that the features are strongly independent and mathematically this is given in Eq. 5.

$$P(X1, \ldots, Xn|Y) = \pi P(Xi|Y) \tag{5}$$

Naïve Bayes model was built using the same training set with 5000 attack and 5000 normal vectors. All the four test cases were re-evaluated with the model built and the results are tabulated. In addition to above, the test cases were evaluated using Naïve Bayes (NB) 10-fold cross-validation. The cross-validation is a process of repeatedly carrying out the experiment 10 times so that each subset is used as test set at least once. This is used to estimate the accuracy and this has been found to be effective when there is sufficient data. The results of the NB Train and Test, NB 10-fold cross-validation, and ANADA are given in Table 3 and the same is depicted as graphs in Fig. 1.

From the above table, it can be clearly observed that DR and accuracy of ANADA are higher in all the cases and F-score of ANADA is also higher in all the cases except for test case-4 which is marginally low. False Alarm Rate (FAR) is lower than NB's Train and Test and 10-fold cross-validation in all the cases which qualifies the usability of the algorithm.

6 Conclusions and Future Work

In this study, a novel adaptive algorithm has been proposed. The proposed method uses the labeled dataset for training but can adapt/learn itself and can detect new attacks. The performance measures of the algorithm can still be improved by combining this algorithm with feature weights. The algorithm has good potential to be parallelized. The future work shall focus on parallelizing the algorithm using GPGPU processors for achieving performance as energy efficiency has become the prime concern for the computer industry. Different sensors for different protocol types can be used for performance improvements. The authors are working on improving the algorithm and modifying it for flow-based anomaly detection.

References

1. https://www.sans.org/reading-room/whitepapers/detection/intruion-detection-systems-definition-challenges-343. Accessed on 06 Jan 2016
2. Panda, M., Patra, M.R.: Network intrusion detection using naive bayes. Int. J. Comput. Sci. Netw. Secur. **7**(12), 258–263 (2007)
3. Jain, M., Richariya, V.: An improved techniques based on Naïve Bayesian for attack detection. Int. J. Emerg. Technol. Adv. Eng. **2**(1), 324–331 (2012)

4. The UCI KDD Archive: KDD Cup 1999 Data, Information and Computer Science, University of California, Irvine. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (1999). Accessed 2 February 2014
5. Muda, Z., Yassin, W., Sulaiman, M.N., Udzir, N.I.: A K-Means and Naive Bayes learning approach for better intrusion detection. Inf. Technol. J. **10**(3), 648–655 (2011)
6. Mukherjee, S., Sharma, N.: Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technol. **4**, 119–128 (2012)
7. Amor, N.B., Benferhat, S., Elouedi, Z.: Naive bayes vs decision trees in intrusion detection systems. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 420–424 (2004)
8. MIT Lincoln Lab., Information Systems Technology Group: The 1998 Intrusion detection off-Line Evaluation Plan. http://www.ll.mit.edu/ideval/files/id98-eval-ll.txt (1998)
9. Münz, G., Li, S., Carle, G.: Traffic, Anomaly detection using K-Means Clustering. In: GI/ITG Workshop MMBnet, Sept 2007
10. Jianliang, M., Haikun, S., Ling, B.: The application on intrusion detection based on k-means cluster algorithm. In: International Forum on Information Technology and Applications, 2009. IFITA'09, pp. 150–152 (2009)
11. Randeep, B., Sharma, N.: A novel density based K-Means clustering algorithm for intrusion detection. In: J. Netw. Commun. Emerg. Technol. **3**(3), 17–22 (2015)
12. Sharma, S.K., Pandey, P., Tiwari, S.K., Sisodia, M.S.: An improved network intrusion detection technique based on K-means clustering via Naïve Bayes classification. In: 2012 International Conference on Advances in Engineering, Science and Management (ICAESM), proceedings, 30–31 Mar 2012. IEEE, Piscataway, NJ (2012)
13. Hussein, S.M., Ali, F.H.M., Kasiran, Z.: Evaluation effectiveness of hybrid IDs using snort with naive Bayes to detect attacks. In: 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP). IEEE (2012)
14. Thomas, C: Performance Enhancement of Intrusion Detection Systems using Advances in Sensor Fusion, Phd Thesis. Supercomputer Education and Research Center, Indian Institute of Science Bangalore, India (2009)
15. Gaffney Jr., J.E., Ulvila, J.W.: Evaluation of intrusion detectors: a decision theory approach. In: 2001 IEEE Symposium on Security and Privacy, 2001. S&P 2001. Proceedings, pp. 50–61. IEEE (2001)
16. Mokarian, A., Faraahi, A., Delavar, A.G.: False positives reduction techniques in intrusion detection systems-a review. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **13**(10), 128 (2013)
17. Laskov, P., Düssel, P., Schäfer, C., Rieck, K.: Learning intrusion detection: supervised or unsupervised? In: Image Analysis and Processing–ICIAP 2005, 1 Jan 2005, pp. 50–57. Springer, Berlin (2005)
18. Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., Nakao, K.: Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In: Proceedings of the 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, Salzburg, 10–13 Apr 2011, pp. 29–36. ACM 2011 (2011). http://dx.doi.org/10.1145/1978672.1978676
19. Ammar, A.: Comparison of feature reduction techniques for binominal classification of network traffic. J. Data Anal. Inf. Process. (2015) http://dx.doi.org/10.4236/jdaip.2015.32002
20. Ihsan, Z., Idris, M.Y., Abdullah, A.H.: Attribute normalization techniques and performance of intrusion classifiers: a comparative analysis. Life Sci. J. **10**(4), 2568–2576 (2013)
21. Chavez, A.R., Hamlet, J., Lee, E., Martin, M., Stout, W.: Network Randomization and Dynamic Defence for Critical Infrastructure Systems, Sandia National Laboratories, New Mexico. SAN2015-3324 (2015)
22. Wang, W., Zhang, X., Gombault, S., Knapskog, S.J.: Attribute normalization in network intrusion detection. In: 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN), 14 Dec 2009, pp. 448–453. IEEE (2009)