THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Intrusion Detection Using Neutrosophic Classifier

V. Jaiganesh
Assistant professor, Department of computer Science, Dr. N.G.P. Arts and Science College, Coimbatore, India
P. Rutravigneshwaran
Research Scholar, Department of Computer Science, Dr. N.G.P. Arts and Science College, Coimbatore, India

Abstract:
Neutrosophic logic has been applied to network intrusion processing problems recently. A novel approach for intrusion thresholding is proposed by defining neutrosophic set in network domain. Neutrosophic is applied to network processing by defining a neutrosophic domain. An intruders region growing are noticed based on neutrosophic logic is implemented for system traffic. A new approach for network demonizing based on neutrosophic set can also be used. In this dissertation, a neutrosophic set is applied to the field of classifiers where an SVM is adopted as the example to validate the feasibility and effectiveness of neutrosophic logic. This brand new function of neutrosophic set consists of neutrosophic set that is integrated into a formulate SVM, and the concert of the achieve classifier N-SVM is evaluated under a network intrusion systems.

Keywords: *IDS, Decision Tree, Network Intrusion Detection, Neutrosophic logic, svm, Fuzzy logic, Intrusion Attacks*

1. Introduction

An intruder can be defined as somebody attempting to break into an existing computer. This person is popularly termed as a hacker, blackhat or cracker. The number of computers connected to a network and the Internet is increasing with every day. This combined with the increase in networking speed has made intrusion detection a challenging process. System administrators today have to deal with larger number of systems connected to the networks that provide a variety of services. Overall intrusion detection involves defense, detection, and importantly, reaction to the intrusion attempts. An intrusion detection system should try to address each of these issues to a high degree. [Balakumar et al.2014]. An insider is a one who has legitimate access to your network or computer and is trying to misuse his privileges [Balakumar et al., 2014]. Insider intrusion is usually an attempt to alleviate privileges or to gain information by probing misconFigured services or just to create mischief. On an average, 80% of security breaches are committed by insiders. Insider attacks are extremely difficult to detect because they happen within a protected and mostly unsuspicious environment. Currently, many IDSs are rule-based systems where the performances highly rely on the rules identified by security experts. Since the amount of network traffic is huge, the process of encoding rules is expensive and slow. Moreover, security people have to modify the rules or deploy new rules manually using a specific rule-driven language [Jiong Zhang et al., 2008]. To overcome the limitations of rule-based systems, a number of IDSs employ data mining techniques. The intention of this papers is (i) To improve the performance of the classifier in terms of true positive, true negative, false positive, false negative, sensitivity and specificity (ii) To improve the classification accuracy of the classifier using fuzzy logic based neutrosophic classifier.

2. Literature Review

In order to improve detection accuracy and efficiency, a new Feature Selection method based on Rough Sets and improved Genetic Algorithms was proposed in Yuteng Guo et al., 2010 for Network Intrusion Detection. Intrusion detection was the act of detecting unwanted traffic on a network or a device. Umak et al., 2014 tried to present MSPSO-DT intrusion detection system. Where, Multi Swam Particle Swarm Optimization (MSPSO) was used as a feature selection algorithm to maximize the C4.5 Decision Tree classifier detection accuracy and minimize the timing speed. Traditional signature-based intrusion detection methods cannot find previously unknown attacks. Juvonen and Sipola. , 2013 aims to combine unsupervised anomaly detection with rule extraction techniques to create an online anomaly detection framework. Aizhong Mi and Linpeng Hai., 2010 applied pattern recognition approach based on classifier selection to network intrusion detection and proposed a clustering-based classifier selection method. In the method, multiple clusters are selected for a test sample. Then, the average performance of each classifier on selected clusters was calculated and the classifier with the best average performance was chosen to classify the test sample. Om and Kundu., 2012 proposed a hybrid intrusion detection system that combines k-Means, and two classifiers: K-nearest neighbor and Naive Bayes for anomaly detection. It consists of selecting features using an entropy based feature selection algorithm which selects the important attributes and removes the irredundant attributes. Katkar and Kulkarni., 2013 evaluates variation in performance of Naive Bayesian classifier for intrusion detection when used in combination with different data pre-processing and feature selection methods. Due to the effective data analysis method, data mining was introduced into IDS. Nadiammai and Hemalatha., 2012 brought an idea of applying data mining algorithms to intrusion detection database. Natesan and Rajesh., 2012 introduced a new approach called cascading classification model based on AdaBoost and Bayesian Network Classifier that can

improve the detection rate of rare network attack categories. In this approach they trained two classifiers with two different training sets. The KDD Cup99 dataset was splitted into two training sets where one contains full of non rare attacks datasets and other contains datasets of rare attack categories.

3. Proposed Neutrosophic Classifier

Fuzzy logic extends classical logic by assigning a membership function ranging in degree between 0 and 1 to variables. As a generalization of fuzzy logic, neutrosophic logic introduces a new component called "indeterminacy" and carries more information than fuzzy logic. One could expect that the application of neutrosophic logic would lead to better performance than fuzzy logic. Neutrosophic logic is so new that its use in many fields merits exploration. In this paper, for the first time, neutrosophic logic is applied to the field of classifiers. A neutrosophic set is a generalization of a classical set and a fuzzy set. Generally, a neutrosophic set is denoted as <T, I, F>. An element x(t, i, f) belongs to the set in the following way: it is t true, i indeterminate, and f false in the set, where t, i, and f are real numbers taken from sets T, I, and F with no restriction on T, I, F, nor on their sum m=t+i+f. Figure.1 shows the relationship among classical set, fuzzy set and neutrosophic set. In a classical set, i = 0, t and f are either 0 or 1. In a fuzzy set, i = 0,0≤t,f≤1 and t + f = 1. In a neutrosophic set, 0≤t,i,f≤1.

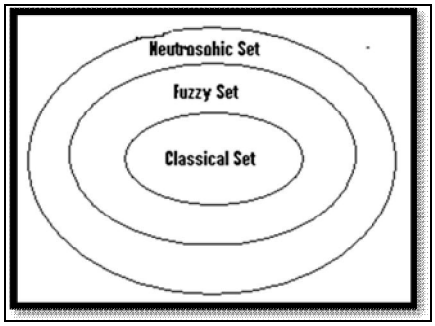

Figure 1: Relationship among classical set, fuzzy set and neutrosophic set

Neutrosophic logic has been applied to network intrusion detection. A novel approach for intrusion thresholding is proposed by defining neutrosophic set in network domain. Neutrosophy is applied to network processing by defining a neutrosophic domain. A intruders region growing are noticed based on neutrosophic logic is implemented for network traffic. A novel approach for network denoising based on neutrosophic set can also be used. In this paper, for the first time, a neutrosophic set is applied to the field of classifiers where an SVM is adopted as the example to validate the feasibility and effectiveness of neutrosophic logic. This brand new application of neutrosophic logic consists of neutrosophic set that is integrated into a reformulated SVM, and the performance of the achieved classifier N-SVM is evaluated under an network intrusion system.

3.1. Background of SVM

Given a training set S containing n labeled points (x1, y1),…, (xn, yn), where xj∈RN and yj∈{-1, 1}, j=1, …, n. Suppose the positive and negative samples can be separated by some hyperplane. This means there is a linear function between the positive and negative samples with the form:

$$d(x)=w.x + b \quad (1)$$

For each training sample x_j, $d(x_j) \geq 1$ if $y_j = 1$; $d(x_j) < -1$, otherwise. This function is also called as decision function. A test sample x can be classified as:

$$y=sign(d(x)) \quad (2)$$

For a given training dataset, many possible hyperplanes could be found to separate the two classes correctly. SVM aims to find an optimal solution by maximizing the margin around the separating hyperplane. The solution for a case in two-dimensional space has an optimal separating line, as shown in Figure.2.

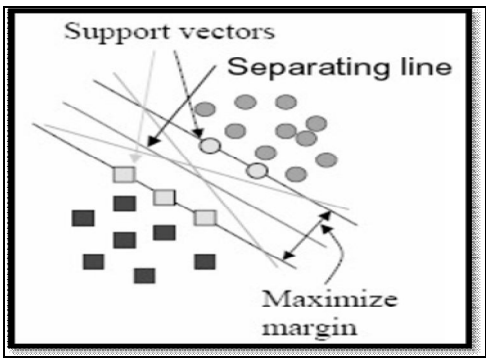

Figure.2: An optimal separating line for a two-dimensional space case

The support vectors are the points on the hyperplanes:

$$y_j(w \cdot x_j + b) = 1 \quad (3)$$

For another sample {xi, yi} that is not on the support vector hyperplanes, it has:

$$y_j(w \cdot x_j + b) = 1 \quad (4)$$

Mathematically, the margin M between two support vectors is finally obtained by:

$$M = \frac{2}{\|w\|} \quad (5)$$

Where $\|w\|$ is the norm of *w*.

Thus, maximizing the margin M is equivalent to minimizing $\|w\|$ with the constraint that there is no sample between the support vector hyperplanes. This constraint can be described as:

$$y_j(w \cdot x_j + b) \geq 1 \quad (6)$$

In the case that the original samples could not be separated by any hyperplane, SVM will transform the original samples into a higher dimensional space by using a nonlinear mapping. Here, $\Phi(x)$ denotes the mapping from R^N to a higher dimensional space Z. A hyperplane needs to be found in the higher dimensional space with maximum margin as:

$$w \cdot z + b = 0 \quad (7)$$

such that for each point (z_j, y_j), where $z_j = \Phi(x_j)$:

$$y_j(w \cdot z + b) \geq = j \quad 1,K,n \quad (8)$$

When the dataset is not linearly separable, the soft margin is allowed by introduction of n non-negative variables, denoted by 1, 2, (...) ξ=ξξ ξn , such that the constraint for each sample in Eq. (8) is rewritten as:

$$y_j(w \cdot z_j + b) \geq 1 - \xi = j \quad 1,K,n \quad (9)$$

The optimal hyperplane problem is the solution to the problem

$$minimize \ \frac{1}{2} w.w + C \sum_{j=1}^{k} \xi \quad (10)$$

$$\text{subject to } y_j(w \cdot z_j + b) \geq 1 - \xi = j \quad 1,K,n \quad (11)$$

where the first term in Eq. (3.10) measures the margin between support vectors, and the second term measures the amount of misclassifications. C is a constant parameter that tunes the balance between the maximum margin and the minimum classification error. Then, for a test point x which is mapped to z in the feature space, the classification result y is given as:

$$y = sign(w \cdot z\ b) \quad (12)$$

3.2. Fuzzy SVM

A membership s_j is assigned for each input sample (x_j, y_j), where $0 < s_j < 1$. Since the membership s_j is the attitude of the corresponding point xj toward one class, and the parameter ξ_j is a measure of error in the SVM, the term $s_j\xi_j$ is a measure of error with different weighting. The optimal hyperplane problem is then regarded as the solution to:

$$minimize \frac{1}{2} w.w + C \sum_{i=1}^{k} S_j \xi_j \quad (13)$$

$$subject\ to\ y_j(w.z_j + b) \geq 1 - \xi_j - j \quad 1,K,n \quad (14)$$

In order to use FSVM, a membership function needs to be defined for each input sample. Here, it used the membership function definition. From Eq. (3.10) one can see that if the ξ_i of a misclassified data x_i is increased, the newly learned hyperplane will have a tendency to correctly classify xi in order to eliminate the larger error that x_i introduced to the classifier and finally minimize Eq. (3.10). Correspondingly in Eq. (3.13), assigning a larger membership s_i for an input increases the probability of correctly classifying that sample while a smaller membership decreases the probability of correctly classifying the sample. Based on this observation, the membership function is defined as follows.

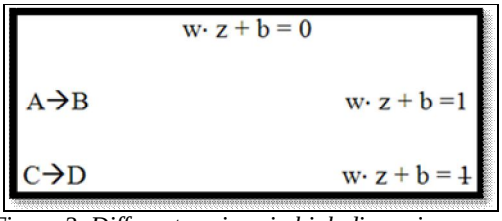

Figure.3. Different regions in high dimension space

1. First, a traditional SVM is trained using the original training set.
2. After step 1, the hyperplane w· z + b = 0 is found. Assuming that if w· z + b > 0 , the data is assigned to the positive class; otherwise, the data is assigned to the negative class. There also are two other hyperplanes w· z + b =1and w· z + b = −1. As indicated in Figure. 4.2, the high dimension space is divided into four regions by these three hyperplanes. For the positive samples, region A represents the input points that are correctly classified and the associated ξ s are 0. Region B represents the input points that are also correctly classified but the associated ξ s are non-zero. Region C and D represents the input points that are incorrectly classified.
3. The points in region A have no contribution to the optimization since their ξ s are 0. Thus, no matter what membership is assigned to them, it will not affect the resultant hyperplane. Here for simplicity, a constant value sA = s1 is assigned to them where 0 < s1 < 1.
4. The points in region B are correctly classified, but they have non-zero ξ s. Thus, they contribute to the optimization equation but should be treated as less important than the points in regions C and D, since they are correctly classified.

The more near to the hyperplane $w \cdot z + b > 0$, the more important in the next training procedure to achieve a better classification result. Given $d=w \cdot z + b$, where $z=\Phi(x)$ for input point x, the membership for region B is defined as:$S_B = S_1 + (1 – d) \times S_2$ (15) where $S_2 > 0$, $0 < s1 + s2 < 1$ and $0 \leq d \leq 1$ in region B.

5. The points in region C are incorrectly classified. It can be predicted that in the next training procedure, the hyperplane can move towards these points, thus allowing more of them can be classified correctly. The nearer the points to the hyperplane $w \cdot z + b > 0$, the less important they are in the next training procedure. As explained in step 4, however, they are more important than the points in region B. Using the same notation as step 4, the fuzzy membership for region C is defined as $S_c = (S_1 + S_2) + |d| \times S_3$ (16)
where $s_3 > 0$, $0 < s_1 + s_2 + s_3 < 1$, and $-1 \leq d \leq 0$ in region C.
6. The points in region D are incorrectly classified. The further away the points are from the hyperplane $w \cdot z + b > 0$, the more probably an outlier exists; thus, the smaller membership should be assigned. The membership for region D is defined as: $S_D = (S_1 + S_2 + S_3)/|d|^k$ (17)
where $k>0$ and $d \leq -1$ in region D. Here, k is a positive integer, and the larger k is, the faster the membership decreases with the increase of distance d. The value of k is chosen as 9 in the experiment. With the memberships defined in steps 3-6, an FSVM is trained and the obtained FSVM is used as a classification tool. Above, are the steps to design the proposed membership function.

3.3. Integrating Neutrosophic Set with Reformulated SVM

In order to use the reformulated SVM, a weighting function for input samples should be defined. Following the steps in Section 4.4, every sample has been associated with a triple $<t_j,i_j,f_j>$ as its neutrosophic components. A larger tj means the sample is nearer to the center of the labeled class and is less likely an outlier. So, t_j should be emphasized in the weighting function. A larger ij means the sample is harder to be discriminated between two classes. This factor should also be emphasized in the weighting function in order to classify the indeterminate samples more accurately. A larger f_j means the sample is more likely an outlier. This sample should be treated less importantly in the training procedure. Based on these analyses, the weighting function gj is defined as:

$$g_j = t_j + i_j – f_j \quad (18)$$

The proposed classifier, denoted as neutrosophic-support vector machine (N-SVM), reduces the effects of outliers in the training samples and improves the performance when compared to a standard SVM.

4. About the Dataset

With the enormous growth of computer networks usage and the huge increase in the number of applications running on top of it, network security is becoming increasingly more important. As it is shown in [Landwehr et al.1994], all the computer systems suffer from security vulnerabilities which are both technically difficult and economically costly to be solved by the manufacturers. Since 1999, KDD'99 [KDD Cup.1999] has been the most wildly used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. 2000 and is built based on the data captured in DARPA'98 IDS evaluation program [Lippmann et al.,2000].

5. Performance Metrics

The following are the performance metrics used to evaluate the performance of the NFSVM, HID (Reda Elbasiony et al.2013) and k-means (Muda et al.,2011). The classification accuracy, sensitivity and specificity can be calculated using the following metrics.

- True Positive: A legitimate attack which triggers an IDS to produce an alarm.
- True Negative: An event when no attack has taken place and no detection is made.
- False Positive: An event signaling an IDS to produce an alarm when no attack has taken place
- False Negative: When no alarm is raised when an attack has taken place.

6. Results and Discussions

Figure 4, Figure 5, Figure 6, Figure 7 depicts the True Positive, True Negative, False Positive, False Negative classification of the algorithms such as K-Means (Muda et al.,2011), HID (Reda Elbasiony et al.,2013) and NFSVM (proposed work i.e in chapter 3). It can be clearly understood that the proposed work NFSVM provides better results than the existing.

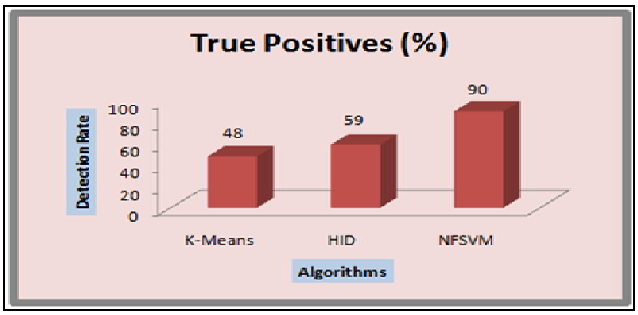

Figure 4: True Positive Analysis

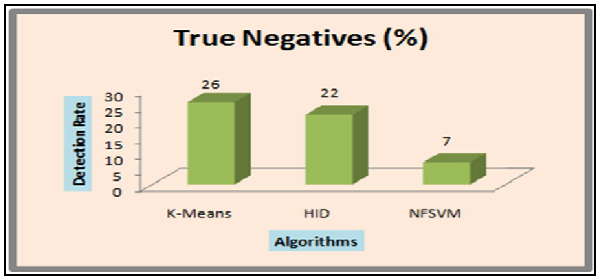

Figure 5: True Negative Analysis

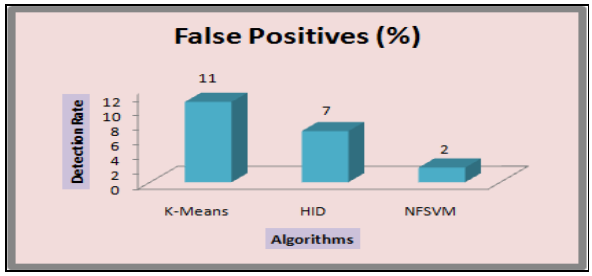

Figure 6: False Positive Analysis

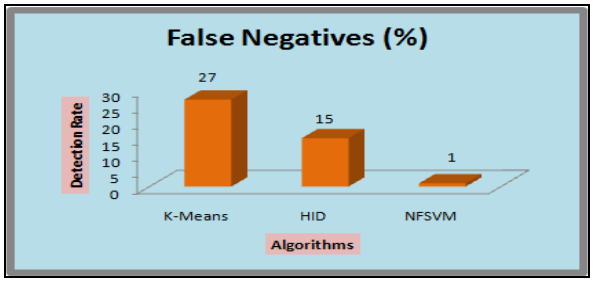

Figure 7: False Negative Analysis

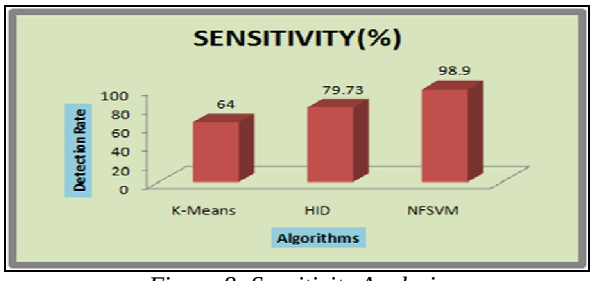

Figure 8: Sensitivity Analysis

Figure 8 depicts the sensitivity of the algorithms such as K-Means (Muda et al. 2011), HID (Reda Elbasiony et al.,2013) and NFSVM (proposed work i.e in chapter 3). It can be clearly understood that the proposed work NFSVM provides better sensitivity result 98.9 respectively.

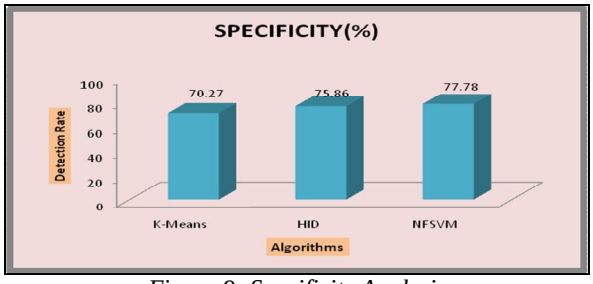

Figure 9: Specificity Analysis

Figure 9 depicts the specificity of the algorithms such as K-Means (Muda et al., 2011), HID (Reda Elbasiony et al.,2013) and NFSVM (proposed work i.e in chapter 3). It can be clearly understood that the proposed work NFSVM provides better specificity result 77.78 respectively.

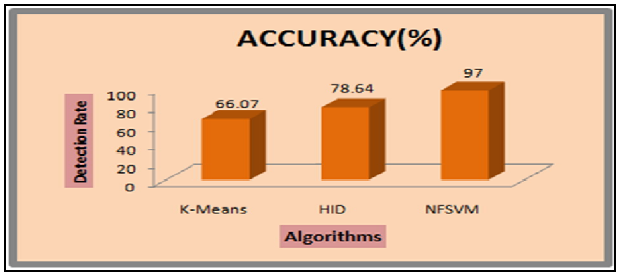

Figure 10: Accuracy Analysis

Figure 10 depicts the accuracy of the algorithms such as K-Means (Muda et al., 2011), HID (Reda Elbasiony et al., 2013) and NFSVM (proposed work i.e. in chapter 3). It can be clearly understood that the proposed work NFSVM provides better accuracy result 97 respectively.

7. Conclusion

In this research work fuzzy logic based neutrosophic classifier is applied in misuse, and anomaly detection. To address the problems of rule-based systems, the fuzzy logic based neutrosophic classifier is employed to build patterns of intrusions. By learning over training data, the proposed algorithm can build the patterns automatically instead of coding rules manually. The proposed approaches are implemented using MATLAB. The implementations are evaluated over KDD'99 dataset, and the experimental results show that the performances of our approaches are better than the best KDD'99 results.

8. References

1. Balakumar, Rangarajan, Ragavi, "Investigate the Use of Honeypots for Intrusion Detection Defense", International Journal of Advanced Research in Computer Science and Software Engineering 4(8), August - 2014, pp. 355-359
2. Jiong Zhang, Mohammad Zulkernine, and Anwar Haque, "Random-Forests-Based Network Intrusion Detection Systems", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 38, No. 5, September 2008\
3. Yuteng Guo, Beizhan Wang , Xinxing Zhao , Xiaobiao Xie , Lida Lin , Qingda Zhou, "Feature selection based on Rough set and modified genetic algorithm for intrusion detection", 2010 5th International Conference on Computer Science and Education (ICCSE), 2010 , Page(s): 1441 - 1446.
4. Umak, Raghuwanshi, Mishra, "Review on speedup and accurate intrusion detection system by using MSPSO and data mining technology",2014 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2014 , Page(s): 1 - 6.
5. Juvonen, Sipola,"Combining conjunctive rule extraction with diffusion maps for network intrusion detection", 2013 IEEE Symposium on Computers and Communications (ISCC), 2013 , Page(s): 000411 - 000416.
6. Aizhong Mi , Linpeng Hai "A clustering-based classifier selection method for network intrusion detection" 2010 5th International Conference on Computer Science and Education (ICCSE), 2010 , Page(s): 1001 - 1004.
7. Om, Kundu "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system" 2012 1st International Conference on Recent Advances in Information Technology (RAIT),2012 , Page(s): 131 - 136.
8. Katkar, Kulkarni, "Experiments on detection of Denial of Service attacks using Naive Bayesian classifier"2013 International Conference onGreen Computing, Communication and Conservation of Energy (ICGCE), 2013 , Page(s): 725 - 730.
9. Nadiammai, Hemalatha, "Perspective analysis of machine learning algorithms for detecting network intrusions" 2012 Third International Conference onComputing Communication & Networking Technologies (ICCCNT), 2012 , Page(s): 1 - 7.
10. Natesan,Rajesh, "Cascaded classifier approach based on Adaboost to increase detection rate of rare network attack categories" 2012 International Conference on Recent Trends In Information Technology (ICRTIT),2012 , Page(s): 417 - 422.
11. Landwehr, Bull, McDermott, and Choi, "A taxonomy of computer program security flaws," ACM Comput. Surv., vol. 26, no. 3, pp. 211–254, 1994.
12. KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Ocotber 2007.
13. Lippmann, Fried, Graf, Haines, Kendall, McClung, Weber, Webster, Wyschogrod, Cunningham, Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02,p. 1012, 2000.
14. Stolfo, Fan, Lee, Prodromidis, Chan, "Costbased modeling for fraud and intrusion detection: Results from the jam project," discex, vol. 02, p. 1130, 2000.
15. Reda Elbasiony, Elsayed Sallam, Tarek Eltobely, Mahmoud Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means", Ain Shams Engineering Journal, Elsevier, 2013 4, 753–762.