ISSN:0975-9646

# Enhancement and Evaluation of Pattern Classifiers in NIDS: A Survey

Nikhil A. Gaikwad[1], Prof. Sunil M. Sangve[2]

*[12] Computer Department, SP Pune University, Pune Maharashtra, India*

*Abstract*— **As the use of Internet is growing, the need of strong computer security and network is increasing. Intrusion detection is an evolving as a research area to fulfil the demands of IT business. Though intrusion prevention is the best option from security viewpoint, practically it may not be possible as hackers are forming new techniques for breaking the security. Hence, detecting an intrusion at the earliest becomes the prudent option. The objective of intrusion detection is to examine the network data continuously in order to monitor the network security and detect the malicious activity in the form of an attack or intrusion. It should have a high detection rate and low false positive. Due to the vastness of data to be examined, the data mining techniques have been focused in the research of network intrusion detection where pattern classifiers play an important role. A generalized framework has been proposed by Biggio et. al. (2013) which is useful for evaluation of classifier security at design stage and explains Support vector machine-based intrusion detection. In this paper, we study the classical and empirical evaluation model from NIDS perspective along with taxonomy of attacks. Also we study the methods available for feature selection so that we could enhance the empirical model for betterment in performance of classifiers.**

*Keywords*— **Network Intrusion Detection, Pattern Classifier, Adversary Applications, Feature Selection**

## I. Introduction

Internet is a global public network and is growing day-by-day. With this growth, there are potential changes in the user demands and business requirements. Technology has certainly served to this E-business market to expand their horizon but this has also brought in certain risks to the business. As there are both, the harmless and harmful users on the Internet, the information being exchanged between two parties, be it between people or people to system or system to system, needs to be secure and authentic. Hence, it is essential to every business organization to ensure the security of their private resources from malicious activity triggered by the hackers. Firewalls do a fair job of protecting a private network by filtering incoming traffic from the Internet. It either allows or disallows the connections based on the organization's security policy and business needs. So this looks like a locking system. The Network Intrusion Detection System (NIDS) complements to this by behaving like a burglar alarm. If someone breaks the lock and tries to steal, then it will raise an alarm.

Network Intrusion Detection Systems (NIDSs) monitor network traffic in order to detect any kind intrusion. If any distrustful activity is observed during monitoring, then they raise an alert to the network or system administrator. As mentioned by Biggio (2010) in [1], the primary goal of NIDS is to differentiate between legitimate and intrusive network traffic. There are two approaches for intrusion detection- Misuse detection and anomaly detection.

In Misuse detection approach, the system tries to classify the normal and abnormal actions from already known attacks. It works by comparing network traffic, system call sequences, or other features of known attack patterns [2]. The main disadvantage of this approach is that they are unable to detect the abnormal or malicious activities which are previously unknown.

In Anomaly detection approach, the system studies the patterns of normal behaviour and identifies an abnormal or malicious behaviour if it is sufficiently different from known normal behaviour. In this approach, a training set is constructed, and updated at regular time periods to observe the changes of normal traffic, by collecting unsupervised network traffic during operation [2]. So here, the system does not look only for exact match but looks for abnormality among other normal patterns. Thus even if a new, unseen pattern is appeared in the network, anomaly detection technique is able to classify it into either legitimate or malicious category. This approach resolves the problem in Misuse detection, but it could fail if the hacker plans an attack to send packets in such manner that enforces wrong learning patterns for intrusion detection.

As network intrusion detection systems have to work upon huge volume of data, the pattern classification techniques have gained much focus these days. Pattern classification is also referred to as Pattern recognition. As mentioned by Duda [3], the primary goal and approach in pattern classification is to postulate the class of given models, process the identified data to eliminate noise (not due to the models), and for any identified pattern choose the model that corresponds best. [3] In simple terms, it is the process of classifying the given set of patterns into a set of labels by analysing the attributes (features) associated with the patterns.

Pattern classifiers are supposed to analyse each record that is each packet coming into the network intrusion detection system. A network packet contains number of attributes which carry certain information. However, not all these attributes are significant to classify the packet into legitimate or malicious category. So, if one concentrates only one significant attributes which have impact on deciding category without affecting end result, then it will certainly save processing time. Feature subset selection (also referred as Feature Selection) can be seen as the process of identifying and removing as many irrelevant and redundant features as possible. In [4], the reasons are mentioned why feature reduction is helpful. This is because (i) irrelevant features do not have any significant impact on the accuracy of prediction, and (ii) redundant features do not play any role in improvement of the result of prediction as they mostly provide same (redundant) information which is already present in other feature(s).

In this paper we have discussed on the study of classification model of attacks, classifier evaluation models, pattern classifiers for intrusion detection and feature selection methods in section II. At last, it is ended with the conclusion and future work in section III.

## II. LITERATURE SURVEY

In this section, we have studied previous research papers related to pattern classifiers and feature selection. The brief review of existing related work is as follows:

As NIDS has to work on huge amount of data and keep looking for malicious activity, the pattern classification techniques are very useful. This area is trending in current research interests and many new ideas are coming up for more accurate, efficient and evolved NIDS.

### *A. Taxonomy of attacks*

In this section we study about the classification of attacks and defences modelled against those attacks. This is mentioned in [5] and explained in [6].

The attack model can be classified into different types based on following three properties-

*1) Influence*: Describes the capability of attacker. Further classified into (i) Causative and (ii) Exploratory. A Causative attack means the attacker is able to influence the training data points which are used for preparing the classifier. An Exploratory attack means the attacker does not influence the learned classifier, but uses other techniques such as probing the detector, to observe information about training data or the detector.

*2) Specificity*: Refers to specificity of the intention of the attacker, i.e. where is the attacker focusing. Further classified into (i) Targeted and (ii) Indiscriminate. In Targeted attack, the focus of attack is on a specific point or a small group/set of points. In an Indiscriminate attack, the attacker has a more flexible goal that involves a general class of points, thus trying to make as much damage as it can, within available time spell.

*3) Security Violation:* Refers to the kind of security breach caused by the attacker. Further classified into (i) Integrity and (ii) Availability. An integrity attack affects in such way that intrusion points (malicious) are classified as normal/ legitimate points, thus increasing false negatives. An availability attack results in any kind of classification error, either false negative or false positive, but effectively making the system unusable.

The defence techniques used against these attacks are-

*1) Regularization*: As per mathematical definition for Regularization in [7], Regularization a process of adding more information (data points) into training dataset, so that in order to tackle an ill-posed problem (An ill-posed problem is one which is not well-posed problem. And, a well-posed problem is such problem for which unique solution exists and the solution is continuously depend on input data) or over-fitting problem (that is too much of noise than the normal data). It can also be interpreted as encoding a prior distribution on the parameters, penalizing parameter choices that are less likely a priory.

*2) Randomization*: Particularly to be used against Targeted attacks. Targeted attacks focus on a particular point or a small set of point, so they are more sensitive to the variation in the decision boundary. If randomization technique is placed on the boundary, then adversary will get faulty feedback from the learner.

3) *Information Hiding*: Particularly to be used against Exploratory attacks. Exploratory attacks try to affect general class of points, so information hiding technique can ensure the safety of significant attributes in the given training dataset.

### *B. Classical and Empirical Performance Evaluation Models*

The classical performance evaluation methods such as k-fold or bootstrapping [5] are based on the assumption, that the data distribution mentioned in training dataset D will appear during the operation as well. This can be useful in case of causative attacks where generally the data distribution during attack will be same as the one in training dataset. But this significantly fails in other kinds of attack. Practically the data distribution during the attack is mostly different than training dataset. Hence the classifier is unable to predict how the data distribution will change. Thus, the only way to train the classifier in classical model is by developing countermeasures against the attack after its occurrence. This is referred to as “reactive” arms race, as shown in Fig. 1.

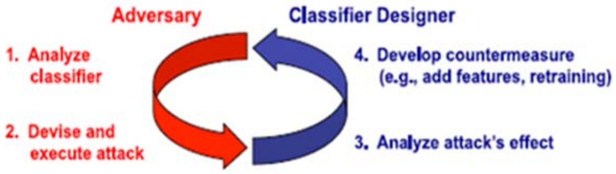
Adversary
Classifier Designer
1. Analyze classifier
2. Devise and execute attack
3. Analyze attack's effect
4. Develop countermeasure (e.g., add features, retraining)

Fig. 1 Classical “reactive” arms race [5]

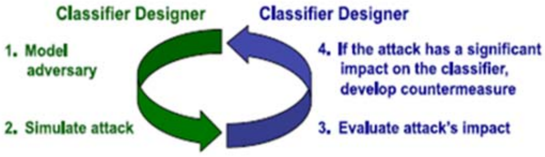
Classifier Designer
Classifier Designer
1. Model adversary
2. Simulate attack
3. Evaluate attack's impact
4. If the attack has a significant impact on the classifier, develop countermeasure

Fig. 2 Empirical “proactive” arms race [5]

On the other hand, in order to make the classifier ready for an attack before its occurrence, the empirical performance evaluation model [5] is devised. This model contains- (i) simulation of the attack, (ii) evaluation of impact of attack and (iii) development of countermeasures, if the attack has significant impact on the classifier. This is referred to as “proactive” arms race, as the system is being ready to detect a probable attack and take appropriate actions.

In [5], three main open issues were identified in classical performance evaluation model: (i) analysing the vulnerabilities of classification algorithms and the corresponding attacks, (ii) developing novel methods to assess classifier security against these attacks, which is not possible using classical performance evaluation methods, and (iii) developing novel design methods to guarantee classifier security in adversarial environments.

The issues (i) and (ii) above are addressed by Biggio et. al. (2013) [5] and they developed a framework for the empirical evaluation of classifier security at design phase. This empirical evaluation model overcomes the limitation of classical methods and extends performance evaluation steps of the classical design cycle of pattern classifier. However, it is a generalized model and only one pattern classifier (one-class SVM) was tested for NIDS application.

### C. *Choosing a pattern classifier*

As there are number of other pattern classification techniques available, only selected ones had to be used for testing of our proposed model. The one-class SVM pattern classifier is already mentioned in empirical evaluation model, so it is selected. To compare one-class SVM, it is good to have a variant of SVM, so Multi-class SVM [8] is selected. Then many recent papers were referred to understand the trend of selecting pattern classifiers by the researchers, specifically in network intrusion detection area. It is observed that, apart from SVM, other classifiers in focus are k-NN and Naïve Bayes ([9], [10]). So they are selected.

*1) One-class SVM*: The Support Vector Machine (SVM) algorithm is generally used for supervised learning. As mentioned in [11], the idea behind SVM is that the data is assumed to be linearly separable. Therefore, a linear hyperplane (or decision boundary) exists there which separates the data points into two different classes. In case of two-dimensional, the hyperplane is a simple straight line. However, in principle, there would be infinite hyperplanes that can separate the training data. The hyperplanes can divide the training records (or points) into their respective classes without making any misclassification errors.

*2) Multi-class SVM*: As mentioned in [8], the Multi-class SVM classifiers can be applied to intrusion detection systems because of multi-types existing in the network. ‘One-against-all’, ‘One-against-one’ and ‘Binary Tree’ are popular methods among them. Binary Tree SVM requires only (k-1) two-class SVM classifiers for a case of k classes.

3) *k-Nearest Neighbour*: As mentioned in [11], In k-NN, the training tuples are described by n attributes. Each tuple represents a point in n-dimensional space. When given an unknown tuple, a k-nearest neighbour (k-NN) classifier searches the pattern space for the k training tuples which are closest to the unknown tuple. These k training tuples are the k-nearest neighbours of the unknown tuple. Closeness is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples X1=(x11, x12,.., x1n) and X2=(x21, x22,.., x2n) obtained from below equation-

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

4) *Naïve Bayes*: Naïve Bayes is a probabilistic classifier. It assigns class labels to problem instances. Problem instances are represented as vectors of feature (attribute) values and class labels are picked from finite set. As mentioned in [12], Naive Bayes classifier assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, the children can be considered playing cricket match if it is Sunday and not raining and cricket kit is available. Thus irrespective of the correlations among the features, the probability of each feature is calculated and contributed to final score.

### D. *Choosing a dataset*

KDD-99 is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition. It was held in association with KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The task for competition was to build a network intrusion detector. It should contain a predictive system which can categorize the connections into “good” (i.e normal or legitimate connections) or “bad” (i.e. intrusions or attacks). KDD-99 database contains a standard set of data to be audited. It includes a variety of intrusions which were simulated in a military network environment [13]. It is considered as a standard dataset for testing of classifiers for NIDS.

### E. *Feature selection methods*

Feature selection is the process of choosing "interesting" features from the dataset for further processing. As the input dataset (KDDCup99) contains more than 40 features, the idea of identifying key features and perform classification on that subset is really worth. While looking for the feature selection algorithms, came across a fast clustering-based feature selection algorithm (FAST) [14]. This algorithm is tested on 35 publicly available image, microarray, and text datasets and has shown promising results in comparison with other techniques listed below-

*1) CFS*: In [15], Correlation-based Feature Selection is achieved by the hypothesis that a good feature subset is one that contains features highly correlated with the target, yet uncorrelated with each other.

*2) FCBF*: In [16], Fast Correlation-Based Filter is explained. It is a fast filter method which can identify relevant features as well as redundancy among relevant features without pairwise correlation analysis.

*3) Relief*: In [17], Relief weighs each feature according to its ability to discriminate instances under different targets based on distance-based criteria function. However, Relief is ineffective at removing redundant features. Relief-F [18] extends Relief method. It enables Relief method to work with noisy and incomplete data sets and to deal with multiclass problems, but still cannot identify redundant features.

*4) Consist*: In [19], the Consist method searches for the minimal subset that separates classes as consistently as the full set can under best first search strategy.

*5) FOCUS-SF*: It is a variation of FOCUS [20]. FOCUS has the same evaluation strategy as Consist, but it examines all subsets of features. Considering the time efficiency, FOCUS-SF replaces exhaustive search in FOCUS with sequential forward selection.

As mentioned in [14], the Fast clusteringbAsed feature Selection algoriThm (FAST) is based on MST (minimum spanning tree) technique. The FAST algorithm works in two steps. It filters out a mass of irrelevant features in the first step. This reduces the possibility of improperly bringing the irrelevant features into the subsequent analysis. Then, in the second step, FAST removes a large number of redundant features by choosing a single representative feature from each cluster of redundant features. As a result, only a very small number of discriminative features are selected.

## III. Conclusions

We observed that empirical evaluation model for pattern classifier has overcome the limitations of classical evaluation model, by modelling the adversary (attacks) in training dataset and later testing it for unseen attacks. We also studied different pattern classifiers which could be helpful for NIDS. We also studied the feature selection methods and noticed that, FAST method has shown better results as compared to others.

Considering the reviewed literature, we propose an enhancement to the empirical model to mould it for network intrusion detection systems. We will continue with the dataset construction steps as mentioned in empirical model and also the one-class SVM. The extension to the model will be done in 2 stages- first by introducing feature selection method on the training dataset, and then by introducing multiple classifiers in the model. We will test their performance and generate comparative output to make sure that the accuracy of classifiers is improved.

## Acknowledgment

We are glad to express our sentiments of gratitude to all who provided their valuable suggestions and recommendations to us. We would like to express sincere thanks to the Principal of our college. We are also thankful to the College Library and staff members and friends.

## References

[1] B. Biggio, G. Fumera, and F. Roli, "Multiple Classifier Systems for Robust Classifier Design in Adversarial Environments," International Journal of Machine Learning and Cybernetics, vol. 1, no. 1, pp. 27-41, August 2010

[2] D. Kang, D. Fuller, and V. Honavar, "Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation", Proceedings of the 2005 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, USA, 2005

[3] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification", Wiley-Interscience Publication, 2000

[4] G.H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," Proc. 11th Intl Conf. Machine Learning, pp. 121-129, 1994.

[5] B. Biggio, G. Fumera, and F. Roli, "Security Evaluation of Pattern Classifiers under Attack", IEEE Transactions on Knowledge and Data Engineering, Vol.26, No.4, April 2014

[6] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, and J.D. Tygar, "Can Machine Learning be Secure?", Proceedings of the ACM Symposium on Information, Computer and Communication Security (ASIACCS), March 2006

[7] Pattern Recognition, https://en.wikipedia.org/wiki/Pattern_recognition

[8] R. Ravishankar Reddy, B. Kavya, and Y. Ramadevi, "A Survey on SVM Classifiers for Intrusion Detection", International Journal of Computer Applications, Vol.98, No.19, July 2014

[9] S. Taruna and S. Hiranwal, "Enhanced Naive Bayes Algorithm for Intrusion Detection in Data Mining", International Journal of Computer Science and Information Technologies, Vol. 4(6), 2013

[10] Hesham Altwaijry and Saeed Algarny , "Bayesian based intrusion detection system", Journal of King Saud University - Computer and Information Sciences, Elsevier, Vol.24, 2012

[11] J. Han, M. Kambler, "Data Minig: Concepts and Techniques" Elsevier, Second Edition, 2006

[12] Naive Bayes Classifier, https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[13] KDDCup99 dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[14] Q. Song, J. Ni, and G. Wang, "A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data", IEEE Transactions on Knowledge and Data Engineering, Vol.25, No.1, January 2013

[15] M.A. Hall, "Correlation-Based Feature Subset Selection for Machine Learning", PhD dissertation, Univ. of Waikato, 1999.

[16] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", Proceedings of the 20th International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

[17] K. Kira and L.A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm", Proc. 10th National Conference Artificial Intelligence, pp. 129-134, 1992.

[18] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," Proc. European Conf. Machine Learning, pp. 171-182, 1994.

[19] M. Dash, H. Liu, and H. Motoda, "Consistency Based Feature Selection", Proc. Fourth Pacific Asia Conference Knowledge Discovery and Data Mining, pp. 98-109, 2000

[20] H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features", Artificial Intelligence, vol. 69, nos. 1/2, pp. 279-305, 1994.