International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

Improving Intrusion Detection using Genetic Linear Discriminant Analysis

Azween Abdullah *[1], Cai Long Zheng [2]

Received 05th September 2014, Accepted 10th October 2014

Abstract: The objective of this research is to propose an efficient soft computing approach with high detection rates and low false alarms while maintaining low cost and shorter detection time for intrusion detection. Our results were promising as they showed the new proposed system, hybrid feature selection approach of Linear Discriminant Analysis and Genetic Algorithm (GA) called Genetic Linear Discriminant Analysis (GLDA) and Support Vector Machines (SVM) Kernels as classifiers with different combinations of NSL-KDD data sets is an improved and effective solution for intrusion detection system (IDS).

Keywords: *IDS, Features selection, Features transformation, NSL-KDD, GLDA, SVM Kernels*

1. Introduction

An intrusion is defined as anything which compromises confidentiality, availability or integrity [14]. User authentications, avoiding programming mistakes, firewalls and data encryptions are first-level defences against cyber-attacks and intrusions. Intrusion prevention is totally dependent on their detection, and detection is a key part of any security tool such as Adaptive Security Alliance, Intrusion Detection System, Intrusion Prevention System, firewalls and checkpoints.

The selection of a suitable data set is the backbone of any efficient intrusion detection approach. The performance of any intrusion detection system (IDS) also depends on the efficiency and accuracy of the data set. If the training data set is precise with optimal contents and rich features, the efficiency of the training as well as test system will be improved. There are many standard pre-built simulated data sets like Darpa's KDD Cup 98, 99, Six UCI db and NSL-KDD etc. KDD-Cup 99 is most widely used as a benchmark data set for training and testing of IDSs. KDD-CUP 99 is built based on the data captured in DARPA'98 which has been criticized by [5], mainly because of the characteristics of the synthetic data. One of the most important deficiencies in the KDD data set is the huge number of redundant records. On analyzing KDD training and test sets, the author found that about 78% and 75% of the records were duplicated in the training and test sets, respectively, which caused the learning algorithms to be biased towards the frequent records, thus preventing them from learning infrequent records which are usually more harmful to networks such as U2R and R2L attacks.

Due to the drawbacks of KDD-Cup 99 which highly affects the performance of evaluated systems and results in a very poor evaluation of intrusion detection approaches, an advanced form of KDD-Cup was proposed, namely NSL-KDD which consists of selected records of the complete KDD data set. Important drawbacks of KDD-Cup are fixed in NSL-KDD data set. Although there are many techniques for intrusion detection such as computational intelligence, soft computing, data mining, this research focuses on using an ensemble of soft computing approaches to improve detection rate and accuracy.

The rest of the paper is organized as follows. In Section 2, related work of IDS is discussed briefly. In Section 3, the proposed model with different phases is discussed and analyzed in detail. Conclusion and future work is mentioned briefly in Section 4.

2. Related Work

Reference [19] adopted the NSL-KDD data set in their research work on feature extraction for intrusion detection using the Linear Discriminant Analysis (LDA) approach. LDA is extensively used as feature dimension reduction approach to find out an optimal transformation that minimizes the within-class scatter and to maximize the between-class distance. Back Propagation Algorithm (BPA) was used to classify attacks into five classes. The Artifical Neural Network (ANN) approach was adopted to compare the performance of the proposed method. In their experiments, 41 features were reduced to only 4 features new space by reducing 97% of the input data and about 94% of the training time as well as same percentage of accuracy in new attack detection [19].

The hybrid approach for feature reduction was adopted by [6] as PCA was not suitable for nonlinear data set as well as for large data set. In their work, the authors preferred Generalized Discriminant Analysis (GDA) over PCA for feature selection. Besides reducing the number of input features, GDA also reduces the training time for classifiers by selecting the most discriminant features. It also increases the accuracy of classification. The anomaly detection approach was used to differentiate between normal data based on normal behavior and attack or intrusive data based on its attack behavior. The Self-Organizing Map (SOM) approach and C4.5 decision tree techniques were applied for classification of reduced feature space. The KDD-Cup 99 data set was applied in this research and 41 features were reduced to 12 features space by GDA. The experimental results showed that GDA outperformed

[1] *School of Computing and IT, Taylors University, Subang Jaya, Selangor, Malaysia*
[2] *Unitar International University, Petaling Jaya, Selangor, Malaysia*
** Corresponding Author: Email: azween.abdullah @taylors.edu.my*

Note: This paper has been presented at the International Conference on Advanced Technology&Sciences (ICAT'14) held in Antalya (Turkey), August 12-15, 2014.

PCA especially for large scale data set by providing a better detection rate as well as reduced training and testing time. Moreover, the C4.5 classifier outperformed SOM for all the attack classes.

An integrated intrusion detection system by [21] was proposed to model and implement an efficient system to reduce false alarms and to increase detection rate. The authors extracted the most important segments from the whole features of data set using Information Gain. To achieve a high detection rate of attacks, the authors introduced a high level of generality while deploying the subset of extracted or selected feature space. Genetic Algorithm (GA) and Radial Basis Functions (RBF) were used to classify known and unknown attacks. GA is based on the principles of genetics and natural selection and has a big potential in the domain of intrusion detection. Each individual in GA is called chromosome. Three basic genetic operations, Selection, Cross over and Mutation are applied sequentially to every individual during each generation. RBF networks are effectively used to prevent from overfitting. The proposed system was deployed using Java and KDD data sets. KDD consists of 41 features out of which 38 were numeric and 3 were symbolic. The performance of proposed system was compared with Hoffman GA rules for intrusion detection. The training time was reduced considerably as only nine features were considered. However due to the random usage of cross over and mutation operations, detection rate was not good for some runs [21].

An efficient intrusion detection system was proposed by [7] using feature subset selection based on MLP. The authors used Principal Component Analysis (PCA) and GA for preprocessing and MLP for feature classification using the KDD-cup data set. LDA outperformed PCA. PCA is not suitable for large data sets [4], hence their work was limited for small-sized data sets and results were not realistic against actual network traffic as there were obvious deficiencies in the KDD-Cup data set.

Reference [8] used PCA for feature reduction and Naive Bayes algorithm for classification to generate a smaller false positive alarms ratio and to increase the detection rate efficiently. The Naive Bayes classifiers used the probabilistic approach to determine attack probability while considering conditional dependency. The 41 features of the KDD 99 data set were reduced to 14 features and 12 major features that had greater Eigen values were identified by PCA. This new feature set contained the explanation for about 80% of the data variability while 98% of the inconsistency can be attributed to 24 features which can be considered as quite a sufficient value [22]. A brief comparison of the different approaches with their results is shown in Table 1.

Table 1. Comparative analysis of existing approaches

Author [Year]	Data set	Architecture	Accuracy
Osareh, and Bita[2008]	KDD	SVM	83%
S. M. Aqil [2010]	KDD	PCA,Naïve Bayes	N/A
Rupali datti [2010]	NSL-KDD	LDA1, ANN, BPA	96.5%
Lakhina et. al. [2010]	KDD-Cup	PCANNA	80.4%
Ahmad et al. [2011]	KDD-Cup	PCA, GA, MLP	99%
Shailendra Singh and Sanjay Silakari [2011]	KDD-Cup	GDA, SOM, C4.5	98%
Rita Ranjani Singh and Neetesh Gupta [2011]	NSL-DD	SOM	95%

3. Proposed Architecture

There were different interdependent phases in the proposed architecture for an efficient IDS. NSL-KDD was selected during the selection of a suitable data set phase. The LDA approach was used for feature transformation and GA for optimum feature subset selection. In the third phase, SVM Kernels was used as the classification approach in this research. After classification, the system was trained and tested according to the standard rules. Figure 1 shows the block diagram for the proposed system.

3.1. Selection of Suitable Data Set

KDD-Cup is the widely used data set for training and testing of IDSs. There are a total of 41 features which are classified into Basic, Content and Traffic features. As a result, some of inherited issues also exist in KDD-Cup like redundancy of similar records and complexity level of data behavior. NSL-KDD is an advanced version of KDD-Cup data set and does not suffer from the shortcomings found in KDD-Cup. The following presents the unique features that helped us pick NSL-KDD over KDD-Cup.

- No redundancy of records
- No duplication records in test data
- Less complexity level
- Affordable records in training and test sets

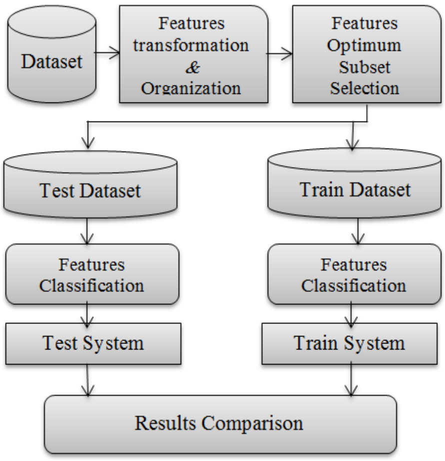


Figure 1. Block diagram of proposed system for IDS

The NSL-KDD features can be classified into the following three groups as shown in Figure 2.

1) Basic features: This category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features lead to an implicit delay in detection.

2) Traffic features: This category includes features that are computed with respect to a window interval and is divided into “Same host” and “Same service” features.

3) Content features: Unlike most of the DoS and probing attacks, the R2L and U2R attacks don't have any intrusion frequent sequential patterns. This is because the DoS and probing attacks involve many connections to some host(s) in a very short period of time; however, the R2L and U2R attacks are embedded in the data portions of the packets, and normally involves only a single connection.

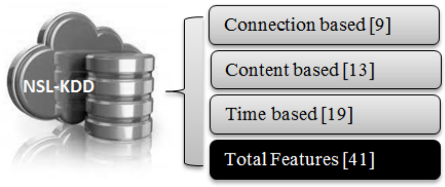


Figure 2. Categories of features in NSL-KDD data set

3.2. Preprocessing of Raw Data set

In most of the existing intrusion detection approaches, raw feature sets are given as input directly to classifiers which causes many problems. In some cases, features are transformed and subset of features is given as input to classifier. In this case, there are also some issues regarding the subset selection scenario. Some major issues in both the above mentioned approaches involve high false alarms, low detection rate and accuracy, loosing important information and many others. A detailed diagram that shows the related issues is shown in Figure 3.

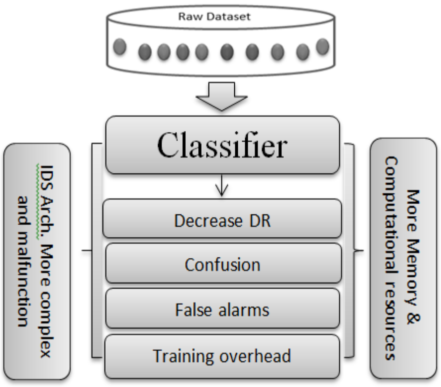

Raw Dataset
Classifier
Decrease DR
Confusion
False alarms
Training overhead
IDS Arch. More complex and malfunction
More Memory & Computational resources

Figure 3. Issues in existing approaches

Instead of directly inserting raw data set to selected classifiers, the raw data set is preprocessed in different ways to overcome different issues like training overhead, classifier confusion, false alarms and detection rate ratios. The preprocessing phase was divided into three sub phases as shown in Figure 4.

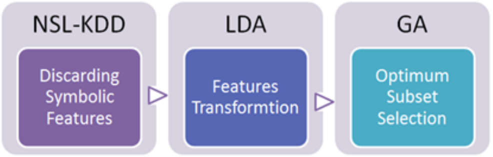


Figure 4. Flow chart of preprocessing steps

3.3. Discarding Symbolic Feature Vectors

There are three kinds of symbolic features (tcp, ftp_data and SF etc.) in the feature space of 41 features. As symbolic values are not of interest to our research, these three feature vectors were discarded to get the following new feature space.

$F(X_m) = X_1, X_2, X_3 \ldots\ldots\ldots X_m \qquad where\ m = 38$

3.4. Feature Transformation and Organization

In most of the existing intrusion detection approaches, raw feature sets are given as direct input to classifiers which cause some of the following issues.

- Using raw data set directly for classifiers guzzles more memory space as well as computational resources during the training and testing phases of the system.
- Detection rate decreases in this case.
- The classifier may become confused and generate false alarms.
- Training overhead is increased due to the processing over each input feature even it is not important for the classifier.
- The architecture of IDS becomes more complex.

To avoid the above mentioned issues, the LDA approach was adopted to transform original numeric feature spaces into new linear feature spaces. LDA is a high-dimensional data analysis method and suitable for feature transformation to facilitate classification [9]. Its steps are shown in Figure 5. There has been a tendency to use the PCA approach for the feature subset selection or reduction in many different domains like face recognition, image compression as well as intrusion detection [10] but LDA has more benefits and is preferred over PCA due to the following reasons.

- LDA outperforms PCA in case of large data sets [4].
- LDA directly deals with both discrimination within-classes as well as between-classes while PCA does not have any concept of the between-classes structure [1].
- LDA preserves class discriminatory information as much as possible while performing dimensionality reduction [11].

The following are steps involved in feature transformation and organization.

Suppose $x = (x_1, x_2, x_3, x_4, \ldots\ldots\ldots\ldots\ldots\ldots x_C)$ are Nx1 feature vectors where C=38 and each feature vector contains n feature samples. Following are steps adapted in LDA algorithm.

Step 1.

Compute the between class scatters using complete feature samples.

$$S_b = \sum_{i=1}^{C} (\alpha_i^j - \alpha_i)(\alpha_i^j - \alpha_i)^T$$

Step 2.

Calculate the Total class scatter matrix.

$$S_t = \sum_{i=1}^{C} \sum_{j=1}^{n} (\alpha_i^j - \bar{\alpha})(\alpha_i^j - \bar{\alpha})^T$$

Step 3.

Compute Eigenvalues and Eigenvectors using Eigen equation for LDA. $S_b X = \lambda S_i X$

Step 4.

Compute the Eigenvectors corresponding to Eigenvalues such that $Eigenvalues: \lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots\ldots \lambda_N$ and Eigenvectors: $X_1, X_2, X_3 \ldots X_N$ where N represents dimensionality of feature vectors and N = 38 in our case.

Step 5.

Evaluate the contribution of each feature vector.

$$C_j = \sum_{p=1}^{m} |V_{pj}|$$

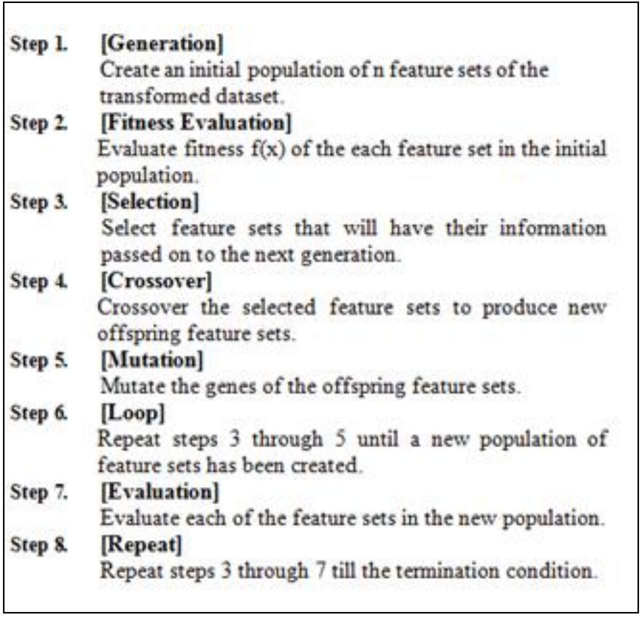
Step 6.

Sort the feature vectors in descending order corresponding to their impact or contribution.

Figure 5. LDA steps for feature transformation

3.5. Optimum Subset Selection

By using LDA for feature transformation, the data set was transformed into a new feature space called linear feature space. This new feature space may also be used as input to the classifier but the classifier becomes biased due to architecture complexity and training and testing efficiency decreases which in turn, increases memory consumption rate and computational cost. GA was applied to select optimal subset of linear features space (Figure 6).



Step 1. **[Generation]**
Create an initial population of n feature sets of the transformed dataset.

Step 2. **[Fitness Evaluation]**
Evaluate fitness f(x) of the each feature set in the initial population.

Step 3. **[Selection]**
Select feature sets that will have their information passed on to the next generation.

Step 4. **[Crossover]**
Crossover the selected feature sets to produce new offspring feature sets.

Step 5. **[Mutation]**
Mutate the genes of the offspring feature sets.

Step 6. **[Loop]**
Repeat steps 3 through 5 until a new population of feature sets has been created.

Step 7. **[Evaluation]**
Evaluate each of the feature sets in the new population.

Step 8. **[Repeat]**
Repeat steps 3 through 7 till the termination condition.

Figure 6. GA specific steps for features subset selection

3.6. Feature Classification

After the selection of the optimum feature subset, the classifier is designed to train and test the features using different Support Vector Machines (SVM) Kernels. The proposed approach was implemented with kernel functions by tuning different parameters including the cost parameter C and other kernel parameters. This was done by selecting parameters using 5x2 cross validation. An overview of the different SVM kernels is shown in Figure 7. The system was trained and tested with the given set of parameters to evaluate the best possible classifier performance on the selected data set.

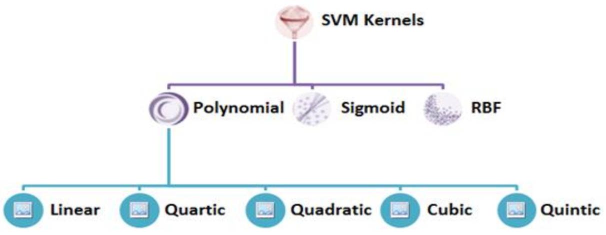
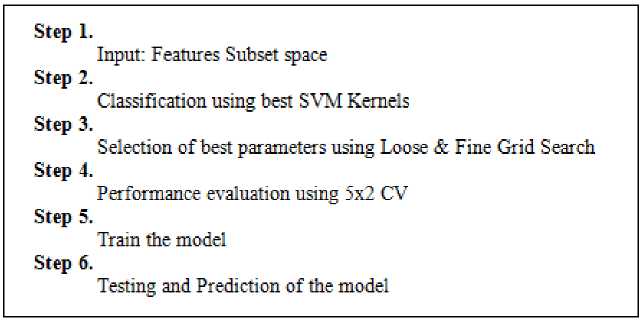


Figure 7. SVM Kernels categories

Figure 8 shows the different steps taken to classify the network traffic into normal or intrusive using SVM kernels.



Step 1.
Input: Features Subset space

Step 2.
Classification using best SVM Kernels

Step 3.
Selection of best parameters using Loose & Fine Grid Search

Step 4.
Performance evaluation using 5x2 CV

Step 5.
Train the model

Step 6.
Testing and Prediction of the model

Figure 8. Classification steps using SVM Kernels

4. Experimental Results

Data sets with 11, 15 and 21 feature vectors were selected for training as well as for testing experiments using the GA approach as the optimum subset from the complete data set of 41 features. Different tools including Net LDA, NeuroSolutions and Matlab were used for this purpose. Table 2 shows the 11 features selected using the GA approach.

Table 2. Optimum features subset from 41 features

No	*Feature Name*	*Type*
1	Duration	Continuous
2	Service	Discrete
3	Count	Continuous
4	dst_bytes	Continuous
5	logged_in	Discrete
6	srv_count	Continuous
7	rv_rerror_rate	Continuous
8	serror_rate	Continuous
9	srv_diff_host_rate	Continuous
10	dst_host_count	Continuous
11	Is_guest_login	Discrete

Network weights were adjusted during the training phase. Confusion matrices were used to verify the training process. The weights of the system were frozen after the training of the system was completed and the system performance was evaluated during the testing phase. The testing phase was divided into verification and generalization steps. The objective of verification was to calculate the learning efficiency of the trained system while the generalization step was used to measure the generalization ability of the trained system using another data set besides the training data set. We selected randomly 10,000 feature samples as the training data set from a total of 125,974 preprocessed linear feature samples while 20% of the training data was used as a cross validation data set. A separate data set of 5,000 was selected randomly from NSL-KDD preprocessed test data set of 22,545 connection records as shown in Figure 9.

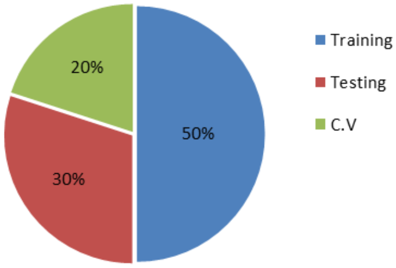


Figure 9. Data set distribution for training, testing & cross validation

We have used several parameters to evaluate the performance of the proposed system which include True Positive, True Negative, False Positive, False Negative, Accuracy rate, Detection rate, Sensitivity and Specificity.

1) Classification Accuracy = 100*(TP+TN) / (TP+FP+FN+TN)

2) Sensitivity: It is the measure of detecting normal patterns accurately.

Sensitivity = (100 * TP / TP + FN)

3) Specificity: It is the measure of detecting intrusive patterns accurately.

Specificity = (100 * TN / TN + FP)

Three different experiments were conducted using different SVM Kernels. Results in Table III reflect that when optimum subset of features is selected, time consumption rate is relatively reduced and accuracy ratio is increased. Since reduced feature space was given as input to the classifier, lesser resources were utilized due to minimum training and learning overheads, hence, computational cost was also minimized. Figure 10 depicts the performance using different subsets.

Table 3. Time & detection rate analysis

No.	Features	Not Selected	Time	Detection Rate
1	11	27	45 h	99.3 %
2	15	23	51 h	99 %
3	21	17	55 h	98.7 %

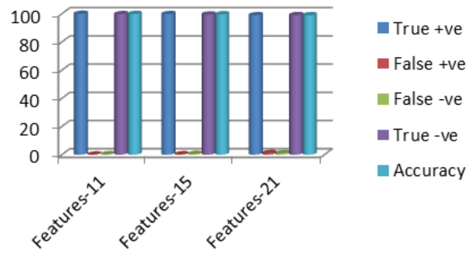


Figure 10. Performance measurements with different features space

The sensitivity and specificity results for different SVM kernels and data feature combinations are shown in Table IV. The graphical analysis for sensitivity and specificity are shown in Figures 11 and 12, respectively. Results in Table 4 clearly show that the RBF kernel performs best for all the recipes of features.

Table 4. Sensitivity & specificity analysis

Cases	*NSL-KDD 11 Features*		*NSL-KDD 15 Features*		*NSL-KDD 21 Features*	
	Sensitivity	*Specificity*	*Sensitivity*	*Specificity*	*Sensitivity*	*Specificity*
RBF Kernel	100	99.2	99.1	99.7	98	98.3
Linear Polynomial	99.7	99.3	99	99.3	99.1	98.7
Sigmoid Kernel	98.9	99	99.1	98.6	99.5	98
Quadratic Polynomial	97	97.4	100	95.7	99.1	97.7
Cubic Polynomial	98.4	95.5	90.1	97.1	100	98.1
Quartic Polynomial	99.9	98.7	100	97.5	99.1	98.1
Quintic Polynomial	99	98.5	93.9	98.1	98.3	96.4

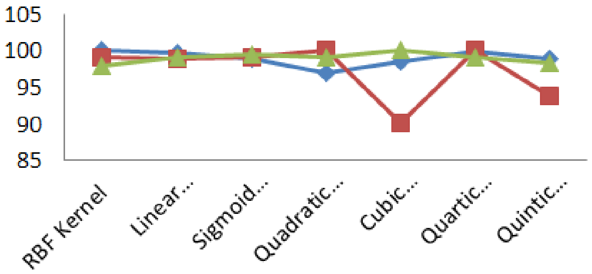


Figure 11. Sensitivity analysis of different feature spaces

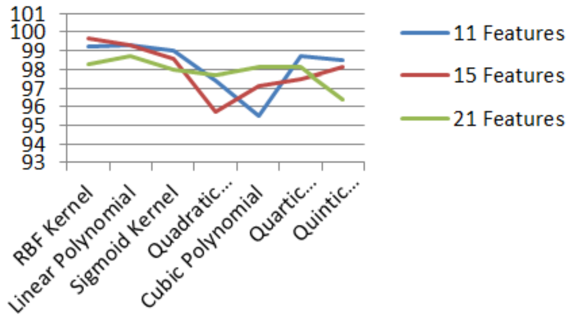


Figure 12. Specificity analysis of different feature spaces

The research results were compared with some existing approaches and are depicted in Figure 13.

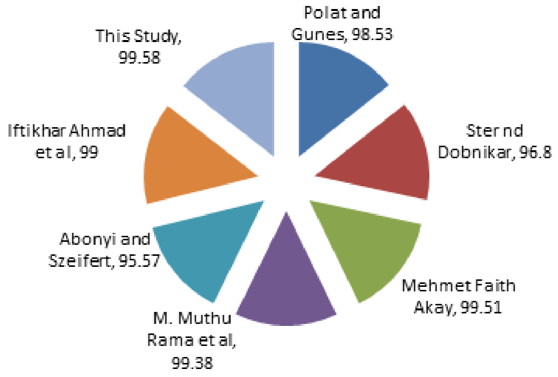


Figure 13. Comparison of new approach with existing approaches

5. Experimental Results

Feature transformation and selection is generally performed using single approach but in our work, the hybrid approach of LDA + GA named as GLDA was adopted to get better results. LDA is preferred over PCA as it outperforms PCA. The advanced form of KDD-Cup called NSL-KDD was used as standard data set. The prominent classification approach SVM with different kernels was used to classify network traffic into normal or intrusive. Our work shows that time consumption rate is relatively reduced whilst accuracy ratio as well as detection rate is increased due to optimum subsets. Since reduced feature space is used as classifier input, minimum resources are utilized and computational cost is minimized due to minimum training and learning overheads.

Our future plan is to design and develop an efficient intrusion detection system for multi class problems by selecting the optimal subset of features.

Acknowledgements

The project is funded by the Ministry of Education Malaysia under the Fundamental Research Grant Scheme 2013 titled: Predictive Analytic Theory Generation and Foundation for a Novel Bio-inspired Intrusion Prevention and Self-regeneration System for Cyber Defence (FRGS/2/2013/ICT02/TAYLOR /02/1).

References

[1] A. Martinez and A. Kak (2001). "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence," vol. 23, no. 2, pp. 228-233,.

[2] China Papers Online (2011). "Study on Application of Hybrid Soft-Computing Technique to Intrusion Detection".

[3] Adel Nadjaran Toosi and Mohsen Kahani (2007) "A new approach to intrusion detection based on an evolutionary soft computing model

using neuro-fuzzy classifiers," Department of Computer, Ferdowsi University of Mashhad, Iran.

[4] Kresimir Delac, Mislav Grgic and Sonja Grgic (2006). "Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set," University of Zagreb, FER, Unska 3/XII, Croatia.

[5] J. McHugh (2000) "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection," ACM Transactions on Information and System Security.

[6] Shailendra Singh, Sanjay Silakari and Ravindra Patel (2011). An Efficient Feature Reduction Technique for Intrusion Detection System, IPCSIT, Vol. 3.

[7] Ahmad I, Abdullah AB, and Alghamdi (2011). "Intrusion detection using feature subset selction based on MLP," Scientific Research and Essays, Vol 6(34).

[8] S. M. Aqil, M. Sadiq Ali Khan and Jawed Naeem (2010). Efficient Probabilistic Classification Methods for NIDS, IJCSIS, Vol. 8, No. 8, November.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman (1996). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, Proc Fourth Eur Conf Computer Vision, Vol. 1, 1418, pp. 45–58.

[10] M. Turk and A. Pentland (1991). "Eigenfaces for recognition," J Cogn Neurosci 3, 71–86.

[11] K. Baek, B. Draper, J.R. Beveridge and K. She (2002). "PCA vs. ICA: A Comparison on the FERET Data Set," Proc. of the Fourth International Conference on Computer Vision, Pattern Recognition and Image Processing, Durham, NC, USA, 8-14, pp. 824-827.

[12] Chittur A. (2006). "Model Generation for an Intrusion Detection System Using Genetic Algorithms," High school Honors Thesis.

[13] Acohido B. (2009). "Hackers breach heartland payment credit card system", 11 March.

[14] Abraham A. and Jain R. (2008). "Soft computing models for network intrusion detection systems, 15 May.

[15] Sandhya P., Ajith A., Crina G. and Thomas J. (2005). "Modeling intrusion detection system using hybrid intelligent systems. Journal of Network and Computer Applications,".

[16] Ilgun K, Kemmerer R.A. and Porras P.A. (1995). "State transition analysis: a rule-based intrusion detection approach," IEEE Trans Software Eng 21(3):181–199.

[17] Zadeh LA. (1994). "History; bisc during 90's,".

[18] Zadeh L.A. (1998). "Roles of soft computing and fuzzy logic in the conception," design and deployment of information/intelligent systems. In: Kaynak O, Zadeh LA, Turksen B, Rudas IJ (eds) Computational intelligence: soft computing and fuzzy-neuro integration with applications, vol 162. Springer, New York.

[19] Rupali D. (2010). "Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 1072-1078.

[20] Liao Y. and Vemuri V. R. (2002). "Use of k-nearest neighbor classifier for intrusion detection," Computer Security, vol. 21, no. 5, pp. 439-448.

[21] Selvakani Kandeeban S. and Rengan S. R. (2010). "Integrated Intrusion Detection System Using Soft Computing", I. J. Network Security 10(2): 87-92. 2008.

[22] M.Sadiq Ali Khan (2012). "Application of Statistical Process Control Methods for IDS," International Journal of Computer Science Issues, Vol. 9, Issue 6, No 1, November.

[23] Chittur A. (2006). "Model Generation for an Intrusion Detection System Using Genetic Algorithms," High school Honors Thesis, accessed in.