A Novel Method of Preprocessing and Evaluating the KDD CUP 99 Data Set

Esraa Saad Abdulmajed, Faculty of Information Science and Technology, National University of Malaysia, Bangi, Selangor, Malaysia. E-mail:esraa1972@yahoo.com

Ravie Chandren Muniyandi, Faculty of Information Science and Technology, National University of Malaysia, Bangi, Selangor, Malaysia. E-mail:ravie@ukm.edu.my

Mustafa Abdullah Azzawi, Faculty of Information Science and Technology, National University of Malaysia, Bangi, Selangor, Malaysia. E-mail:mustafaa@siswa.ukm.edu.my

Abstract--- Detection of cyber-based attacks on computer networks continues to be a very important and challenging area of research in the field of Network Intrusion Detection. The KDD Cup 99 research dataset which was developed specifically for Network Intrusion Detection (NID) research is preferred by most researchers. Despite the popularity of the KDD Cup 99 in the field of Network Intrusion Detection research, it suffers from some drawbacks which entail appropriate preprocessing before it can be used. In this article a novel method of preprocessing and evaluating the KDD Cup 99 research dataset has been proposed. The simulation results showed that the KDD Cup 99 research dataset thus preprocessed was able to test and train the Basic ELM and the Online Sequential ELM to a high degree of accuracy.

Keywords--- KDD Cup 99, Network Intrusion Detection, Cyber-Based Attacks, KDD Cup 99 Preprocessing.

I. Introduction

Development of efficient mechanisms for detection of cyber-based attacks on computer networks continues to be a very important and challenging area of research. An important area of research in this field focuses on improving the efficiency of detecting network intrusion by cyber-attacks which consists of multiple classes. The KDD Cup 99 research dataset which was developed specifically for Network Intrusion Detection (NID) research is preferred by most researchers.

Despite the criticism of the KDD Cup 99 by some NID researchers, it remains the standard research dataset best suited to benchmark performance and compare the effectiveness of various approaches to Network Intrusion Detection. This is primarily because it is the most trusted and credible public benchmark dataset available.

The original KDD Cup 99 dataset consists of nearly 5,000,000 labeled records, each having 41 features. The labeled records belong to either the ‘‘Normal Class’’ or the ‘‘Attack’’ class. The 4 attack classes are:

- DDOS – Distributed Denial of Service Attack which consumes computing resources such as CPU cycles and/or memory preventing the system from handling normal tasks.
- Probing Attack – Collecting information about a network with the intent of exploiting discovered vulnerabilities.
- U2R – via a normal user account for which the attacker has obtained access, a system vulnerability is exploited in order to gain root access and therefore control of the system.
- R2L – Remote to local attack leverages system vulnerability to gain access to a network resource from a distance.

Since the KDD Cup 99 is the most popular research dataset of choice among researchers for performing Network Intrusion Detection research, developing and evaluating approaches for preprocessing the Data Set to eliminate duplicate records, to ensure that the number of records in the different classes are balanced and that the records are scaled in the same range to facilitate evaluation and analysis is of paramount importance.

II. Literature Review

With a significant shift in research interest in employing anomaly detection as opposed to signature-based IDSs for Network Intrusion Detection (NID) research [1] to overcome its inherent drawbacks and weaknesses, significant research has been done in analyzing the KDD Cup 99 as it is the most popular and widely used research data set for evaluation of these systems [1].

According to [1], the current usage of the KDD Cup 99 data set in Network Intrusion Detection (NID) research suffers from some inherent weaknesses [1] which are the existence of a high number of redundant records and the proclivity of many papers to use random parts of the KDD train set as training sets. The high number of redundant records causes the learning algorithms to be biased towards the records which occur with a higher frequency thus preventing them from learning infrequent records which are usually more harmful to networks such as U2R and R2L attacks. Moreover, the existence of these repeated records in the test set will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records. In the field of machine learning, many researchers have tried to construct complex learners to optimize accuracy and detection rate over the KDD Cup 99 data set. In [1], the J48 decision tree learning [2], Naive Bayes [3], NBTree[4], Random Forest [5], Random Tree [6], Multi-layer Perceptron [7], and Support Vector Machine (SVM) [8] are selected from the Weka collection to learn the overall behavior of the KDD Cup 99 dataset.

In [1], a solution to solve the above-mentioned issues is proposed. In this approach, the new train and test sets consist of selected records of the complete KDD Cup 99 data set. The selected data set does not suffer from the above-mentioned drawbacks. Moreover, the number of records in the modified train and test sets created in [1] is reasonable. This makes it feasible to perform experiments on the complete set without the necessity of randomly selecting a small portion of the whole dataset. This allows evaluation results of different research work to be consistent and comparable. Moreover, since the KDD Cup 99 is built in based on the data captured in DARPA'98 [2], some of the existing problems in DARPA'98 remain in the KDD Cup 99.

Despite this data set having some of the drawbacks and problems discussed in [2] primarily because of the characteristics of the synthetic data and failing to be a perfect representative of existing real networks, the lack of public data sets for network-based IDSs permits it to be employed as an effective benchmark data set to aid researchers compare varying intrusion detection methods.

In [3], Portnoy et al. partitioned the KDD data set into ten subsets, each with approximately 490,000 instances or 10% of the data. Despite this, they observed that the distribution of the attacks in the KDD data was very uneven which rendered cross-validation very difficult. Moreover, a lot of these subsets had instances of only a single type. This drawback of the KDD Cup 99 data set has also been reported in [4]. The authors in [4]mentioned two problems caused by including the smurf and neptune attacks in the data set. The first problem was that these two types of DoS attacks constituted over 71% of the testing data set which completely effected the evaluation. Secondly, since they generated large volumes of traffic, they were easily detectable by other methods and there was no dire necessity of employing anomaly detection systems to find them. In [9], a novel machine learning algorithm for improving the classification ability of the KDD Cup 99 is proposed which consists of a combination of a discretizator, a filter method and a very simple classical classifier. The results obtained demonstrated the feasibility of this approach which achieved comparable or even better performances than many complex algorithms but with a significant reduction in the number of input features. The proposed method was tested over another two large datasets which maintained the same behavior as the KDD Cup 99 dataset.

In [10], a new approach which merges feature selection and classification for multiple class NSL-KDD Cup 99 intrusion detection dataset employing a Support Vector Machine (SVM) is proposed. Experimental results show that the proposed method is able to achieve 91% classification accuracy using only three features, 99% classification accuracy using 36 features and all 41 training features achieving 99% classification accuracy.

In [11], a fuzzy approach to feature reduction is presented and the evolved features are analyzed using classification algorithms in Tanagra. Experimental results have shown that the proposed algorithm yields a very low misclassification rate when compared to other algorithms. In [12], an efficient algorithm for intrusion attack classification is proposed by analyzing the KDD Cup 99. In [13], an intrusion detection model is proposed by blending Fuzzy-C-means clustering, Artificial Neural Network (ANN) and support vector machine (SVM). This model is proven to significantly improve the prediction of network intrusion. The simulation experiments and evaluations of the model were performed with the corrected KDD Cup 99 dataset.The sensitivity, specificity and accuracy of the model were chosen as the evaluation metrics.

III. Problem Statement

Using the KDD Cup 99 for Network Intrusion Detection research suffers from 3 main drawbacks:

1. Duplicated records exist in the research dataset.
2. There is an imbalance in the number of records in the different classes.
3. There is no standardized scale to relatively compare and analyze the different records.

IV. Methodology

In order to overcome the drawbacks of the KDD Cup 99 research dataset, constructing and evaluating a novel approach of preprocessing the KDD Cup 99 is proposed in this work. The steps of preprocessing and evaluating the approach are:

A. Downloading and Reading the Dataset

The complete dataset can be downloaded from this link:

http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. After the dataset is downloaded, MATLAB is employed to read it. The dataset contains 494021 records. Each record contains 42 fields 4 of which have string values. The fields which have string values are: Field 2 - protocol type, Field 3 – service, Field 4 – flag and Field 42 - attack type.

The dataset is unbalanced which is illustrated in the distribution of the records in the different classes as shown in Table 1 below:

Table 1: Distribution of the Records in the KDD Cup 99 Before Preprocessing

Normal class	**97278**
DDOS Attack	**391458**
Smurf	280790
Teardrop	979
Pod	264
back	2203
Land	21
Neptune	107201
Probing Attack	**4107**
Ipsweep	1247
Portsweep	1040
Nmap	231
Satan	1589
U2R Attack	**52**
Perl	3
Loadmodule	9
buffer_overflow	30
Rootkit	10
R2L Attack	**1126**
ftp_write	8
guess_passwd	53
Imap	12
Multihop	7
Phf	4
Warezmaster	20
Warezclient	1020

B. Preprocessing the Dataset

Before the dataset can be used, string values need to be replaced with integer values and duplicated records need to be eliminated. Each field of string values will have a number of integer values equal to the number of string values; for example, the field “protocol type” will have integer values of 1, 2 or 3. Since the dataset contains approximately half a million records, comparing each record with all the other records to remove the duplicated ones is a very time-consuming and cumbersome process.

To reduce the complexity and speed up the process, the dataset is split up into subsets that do not have any shared records. To do this, the data is split up into records that have the same values of “attack type”,” protocol type”, “service”, “flag”, and “logged in” fields. The “logged in” field is a binary field with a field number value of 12. Even after the data is split, it is noticed that some subsets have a large number of records, which increases the possibility of the existence of duplicated records in that subset.

In order to resolve this problem, the sum of the integer values of the fields of all the records is computed. If the summation of the integer values of the fields of two records is different, the two records are considered to be different.Else, the two records which have identical values of summation are examined more closely to determine

whether or not they are duplicates.After all the duplicated records are removed, the total number of records reduces to 145,586.

The number of records remaining in each class after the duplicated records have been removed is illustrated in Table 2 below:

Table 2: Distribution of the Records in the KDD Cup 99 After Preprocessing

Normal class	**87832**
DDOS Attack	**54572**
Smurf	641
Teardrop	918
Pod	206
back	968
Land	19
Neptune	51820
Probing Attack	**2131**
Ipsweep	651
Portsweep	416
Nmap	158
Satan	906
U2R Attack	**52**
Perl	3
Loadmodule	9
buffer_overflow	30
Rootkit	10
R2L Attack	**999**
ftp_write	8
guess_passwd	53
Imap	12
Multihop	7
Phf	4
Warezmaster	20
Warezclient	893
Spy	2

In addition to replacing string values with integer values and removing duplicated records, the value of each field has to be scaled to the range of [0 1]. It is important to keep in mind that regardless of the record, all values of feature 20 “num outbound cmds” and all values of feature 21 “is host login” are 0 before scaling.

C. Saving the Results

Before saving the results, field 43 is added which has the value of mapping from attack type (1 to 23) to attack class (1 to 5). The resultant data after reading them and all the preprocessing processes along with any helpful data is saved in a mat file named “TheKDDCup99.mat”.

The most important variables in the mat file are listed in Table 3 below:

Table 3: Important Variables in the Mat File Containing the Saved Dataset

trainData	A matrix contains the read data
reducedTrainData	A matrix contains the data after removing the duplicated records
scaledTrainData	A matrix contains the reduced data after scaling to the range [0 1]
Classes	A cell matrix contains the name of all attack types
Protocols	A cell matrix contains the name of all used protocols
Services	A cell matrix contains the name of all services found in the dataset
flags	A cell matrix contains the name of all flags found in the dataset

A flowchart depicting the procedure of downloading and reading the dataset, preprocessing the dataset and saving the results is illustrated in Figure 1.

V. Simulation Model

In order to validate the proposed approach for preprocessing the KDD Cup 99, a simulation model using the MATLAB environment was created to preprocess the KDD Cup 99 research dataset which is then used to train a basic ELM and an online sequential ELM. These simulations are performed to analyze the training and testing accuracy for each learning algorithm.Figure 1 below shows the flowchart of the basic tasks of the model.

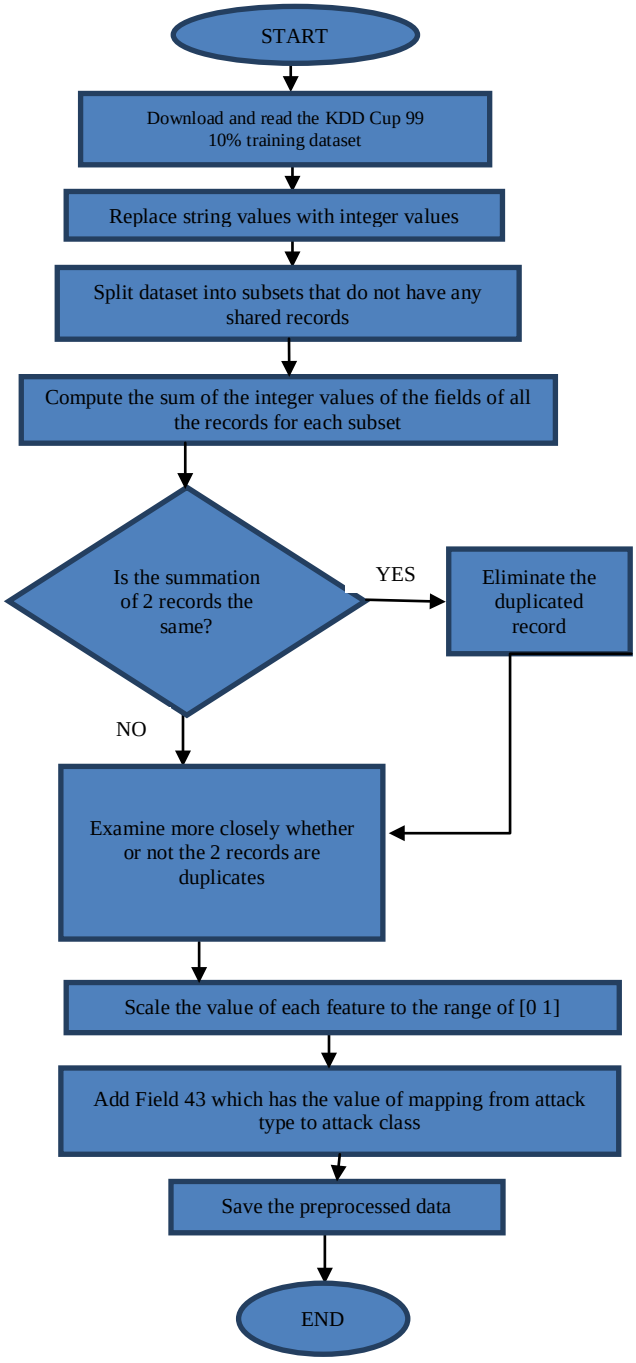


Figure 1: Flowchart of the Proposed Approach for Downloading and Reading, Preprocessing and Saving the KDD Cup 99 Dataset

VI. Results and Discussion

Training Accuracy

The results of the training accuracy of the learning algorithms for different number of neurons in the hidden layer areshown in Table 4. The results provide a comparison between the performance of basic ELM and online sequential ELM, the accuracy of the basic ELM shows better outcomes than the online ELM.

Table 4: Training Accuracy of the Learning Algorithms

Number of neurons in the hidden layer	Basic ELM	Online Sequential ELM
150	0.9912	0.9883
180	0.9914	0.9894
200	0.9920	0.9890

Testing Accuracy

Alternatively, the results of the testing accuracy of the learning algorithms for different number of neurons in the hidden layer are summarized in Table 5. The results show thatthe basic ELM shows a slightly better outcome than the onlinesequential ELM and the number of the neurons in the hidden layer does not have a radical impact on the accuracy.

Table 5: Testing accuracy of the learning algorithms

Number of neurons in the hidden layer	Basic ELM	Online Sequential ELM
150	0.9258	0.9224
180	0.9255	0.9231
200	0.9283	0.9224

Confusion Matrices

Taking into account the fact that there are 5 attack classes (Normal, DDOS, Probing, U2R, and R2L), the confusion matrix for each case is illustrated in Figures 2 (*a*) to (*l*) below. A comparison is made between the basic ELM and the online sequential ELM, including 150,180, 200 neurons in the hidden layer with both cases of training and testing, hence six confusion matrices here are used for each approach.

$$\begin{bmatrix} 0.9974 & 0.0008 & 0.00030.0000 & 0.0015 \\ 0.0121 & 0.9879 & 0.00010.0000 & 0.0000 \\ 0.1065 & 0.0066 & 0.88690.0000 & 0.0000 \\ 0.3269 & 0.0577 & 0.01920.4038 & 0.1923 \\ 0.1111 & 0.0030 & 0.00000.0010 & 0.8849 \end{bmatrix}$$

(a) Basic ELM with 150 neurons in the hidden layer for Training data

$$\begin{bmatrix} 0.9956 & 0.0015 & 0.00220.0001 & 0.0006 \\ 0.0557 & 0.9408 & 0.00350.0000 & 0.0000 \\ 0.2241 & 0.1566 & 0.61370.0000 & 0.0056 \\ 0.7857 & 0.0000 & 0.04290.1429 & 0.0286 \\ 0.9388 & 0.0330 & 0.01990.0007 & 0.0075 \end{bmatrix}$$

(b) Basic ELM with 150 neurons in the hidden layer for Testing data

$$\begin{bmatrix} 0.9972 & 0.0010 & 0.00030.0000 & 0.0014 \\ 0.0121 & 0.9878 & 0.00000.0000 & 0.0001 \\ 0.0924 & 0.0009 & 0.90660.0000 & 0.0000 \\ 0.3269 & 0.0000 & 0.00000.4423 & 0.2308 \\ 0.1111 & 0.0080 & 0.00000.0000 & 0.8809 \end{bmatrix}$$

(c) Basic ELM with 180 neurons in the hidden layer for Training data

$$\begin{bmatrix} 0.9929 & 0.0038 & 0.00260.0000 & 0.0007 \\ 0.0523 & 0.9369 & 0.01070.0000 & 0.0001 \\ 0.2054 & 0.1089 & 0.68530.0000 & 0.0004 \\ 0.8000 & 0.0143 & 0.00000.1714 & 0.0143 \\ 0.9722 & 0.0043 & 0.01340.0003 & 0.0098 \end{bmatrix}$$

(d) Basic ELM with 180 neurons in the hidden layerfor Testing data

$$\begin{bmatrix} 0.9976 & 0.0009 & 0.00020.0000 & 0.0012 \\ 0.0108 & 0.9891 & 0.00000.0000 & 0.0000 \\ 0.0953 & 0.0070 & 0.89770.0000 & 0.0000 \\ 0.2885 & 0.0385 & 0.00000.4423 & 0.2308 \\ 0.1041 & 0.0040 & 0.00100.0020 & 0.8880 \end{bmatrix}$$

(e) Basic ELM with 200 neurons in the hidden layerfor Training data

$$\begin{bmatrix} 0.9952 & 0.0016 & 0.00270.0001 & 0.0004 \\ 0.0554 & 0.9391 & 0.00550.0000 & 0.0000 \\ 0.2409 & 0.0884 & 0.66410.0000 & 0.0067 \\ 0.5857 & 0.0571 & 0.02860.2571 & 0.0714 \\ 0.9258 & 0.0036 & 0.02680.0007 & 0.0432 \end{bmatrix}$$

(f) Basic ELM with 200 neurons in the hidden layerfor Testing data

$$\begin{bmatrix} 0.9971 & 0.0013 & 0.00030.0000 & 0.0013 \\ 0.0170 & 0.9822 & 0.00080.0000 & 0.0001 \\ 0.1122 & 0.0136 & 0.87420.0000 & 0.0000 \\ 0.3462 & 0.0000 & 0.05770.4231 & 0.1731 \\ 0.1732 & 0.0040 & 0.00100.0020 & 0.8198 \end{bmatrix}$$

(g) Online Sequential ELM with 150 neurons in thehidden layer for Training data

$$\begin{bmatrix} 0.9925 & 0.0046 & 0.00230.0000 & 0.0004 \\ 0.0539 & 0.9403 & 0.00570.0000 & 0.0000 \\ 0.2539 & 0.1641 & 0.58170.0000 & 0.0004 \\ 0.6429 & 0.0000 & 0.18570.1571 & 0.0143 \\ 0.9454 & 0.0072 & 0.04550.0007 & 0.0013 \end{bmatrix}$$

(h) Online Sequential ELM with 150 neurons in thehidden layer for testing data

$$\begin{bmatrix} 0.9971 & 0.0012 & 0.00040.0001 & 0.0013 \\ 0.0157 & 0.9841 & 0.00000.0000 & 0.0001 \\ 0.1178 & 0.0023 & 0.87850.0000 & 0.0014 \\ 0.5000 & 0.0000 & 0.00000.3077 & 0.1923 \\ 0.1211 & 0.0050 & 0.00300.0020 & 0.8689 \end{bmatrix}$$

(i) Online Sequential ELM with 180 neurons in thehidden layer for Training data

$$\begin{bmatrix} 0.9866 & 0.0113 & 0.00180.0000 & 0.0004 \\ 0.0536 & 0.9432 & 0.00300.0000 & 0.0002 \\ 0.2278 & 0.0940 & 0.67820.0000 & 0.0000 \\ 0.8571 & 0.0143 & 0.00000.0429 & 0.0857 \\ 0.9716 & 0.0059 & 0.01440.0000 & 0.0082 \end{bmatrix}$$

(j) Online Sequential ELM with 180 neurons in thehidden layer for Testing data

$$\begin{bmatrix} 0.9972 & 0.0010 & 0.00030.0001 & 0.0014 \\ 0.0161 & 0.9832 & 0.00060.0000 & 0.0001 \\ 0.1164 & 0.0174 & 0.86630.0000 & 0.0000 \\ 0.4423 & 0.0000 & 0.00000.3269 & 0.2308 \\ 0.1091 & 0.0040 & 0.00000.0010 & 0.8859 \end{bmatrix}$$

(k) Online Sequential ELM with 200 neurons in thehidden layer for Training data

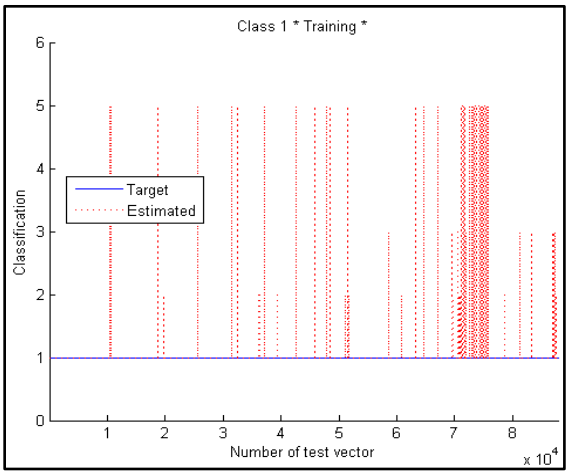
$$\begin{bmatrix} 0.9864 & 0.0107 & 0.00210.0001 & 0.0008 \\ 0.0470 & 0.9439 & 0.00620.0000 & 0.0029 \\ 0.2539 & 0.0831 & 0.66260.0000 & 0.0004 \\ 0.9429 & 0.0143 & 0.00000.0286 & 0.0143 \\ 0.9863 & 0.0023 & 0.00880.0000 & 0.0026 \end{bmatrix}$$

(l) Online Sequential ELM with 200 neurons in the hidden layer for Testing data

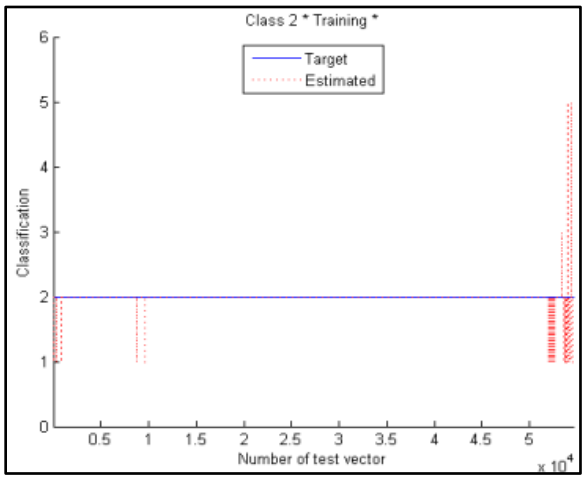
Figure 2: Confusion Matrices for all12 Cases

Graphical Representations

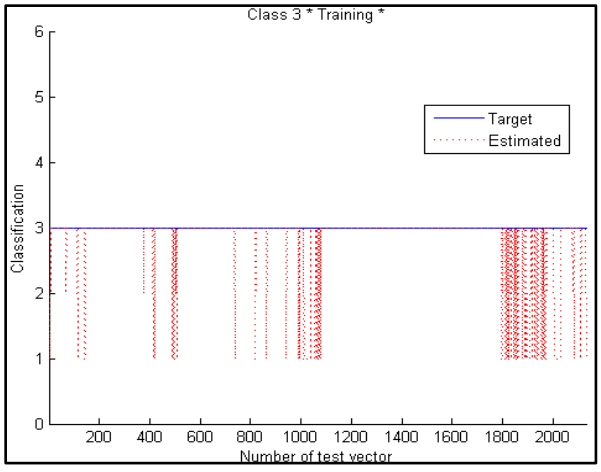
To better understand these values, the confusion matrices are visualized as graphs which indicate the estimated and targeted values, Figure 3 below showsthe graphical representation of the basic ELM with 200 neurons in the hidden layer for the training dataset.



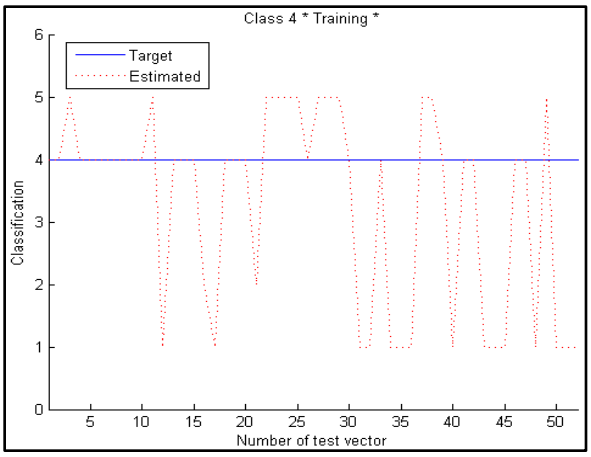
(a)



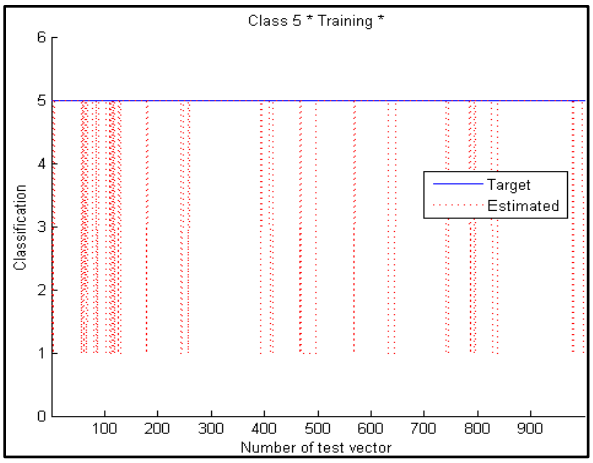
(b)



(c)



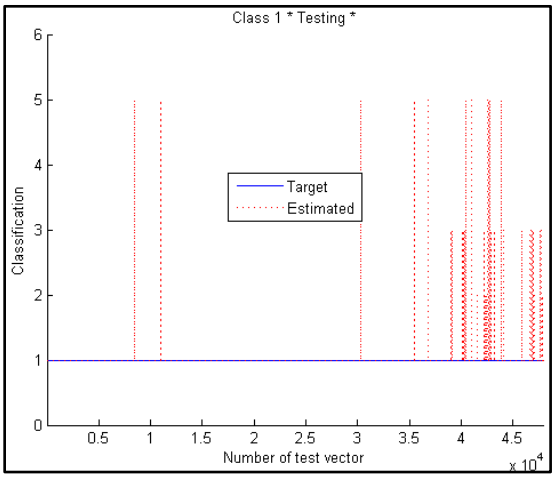
(d)



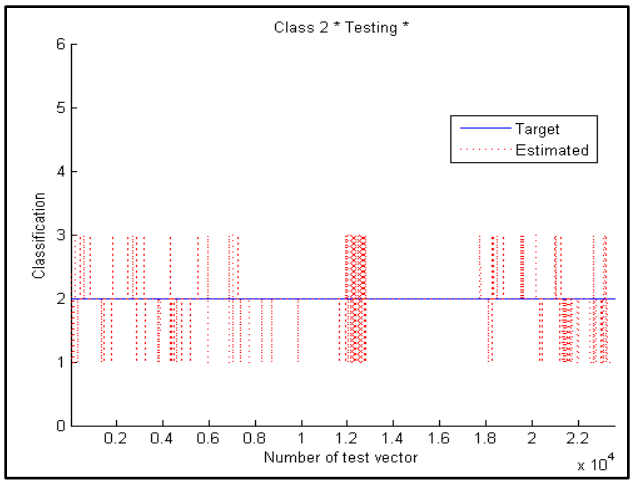
(e)

Figure 3: Graphical Representation of the Basic ELM with 200 Neurons in the Hidden Layer for the Training Dataset

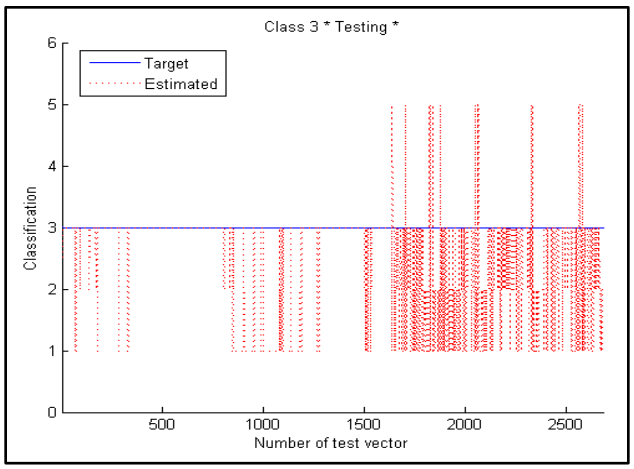
On the other hand, figure 4 below shows the graphical representation of the basic ELM with 200 neurons in the hidden layer for the testing dataset. The estimated values for each class is accurate enough to reach the target of this model.



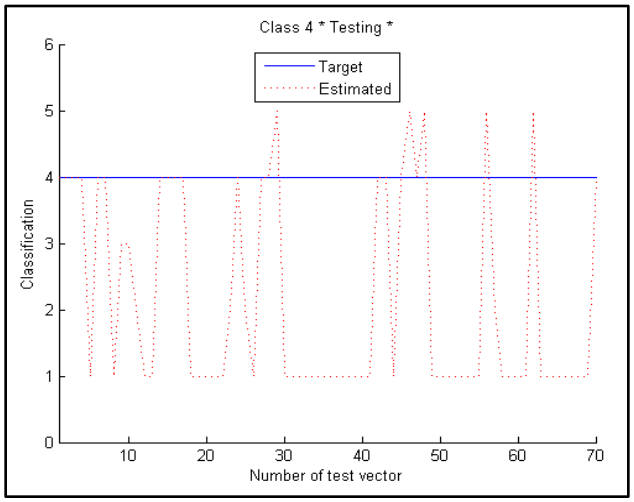
(a)


Class 2 * Testing *
Target
Estimated
Classification
Number of test vector
x 10^4

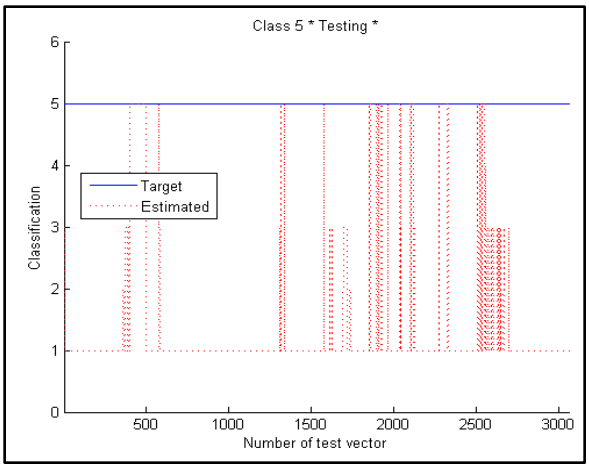
(b)


Class 3 * Testing *
Target
Estimated
Classification
Number of test vector

(c)


Class 4 * Testing *
Target
Estimated
Classification
Number of test vector

(d)



(e)

Figure 4: Graphical Representation of the Basic ELM with 200 Neurons in the Hidden Layer for the Testing Dataset

As can be seen from the confusion matrices and the graphical representations in the figures above, the KDD Cup 99 research dataset thus preprocessed was able to test and train the Basic ELM and the Online Sequential ELM to a high degree of accuracy. Moreover, it is evident that the accuracy of training and testing the Basic ELM is higher than the Online Sequential ELM. Furthermore, using the preprocessed KDD Cup 99 training dataset to test the learning algorithms achieved a higher degree of accuracy than when it was used to train them. Additionally, it was observed that an increase in the number of neurons in the hidden layer generally increased the training and testing accuracy of the learning algorithms when the preprocessed KDD Cup 99 research dataset was used.

VII. Conclusion and Future Work

In this article a novel method of preprocessing and evaluating the KDD Cup 99 research dataset has been proposed. In order to validate the proposed approach, the KDD Cup 99 research dataset thus preprocessed was used to train a basic ELM and an online sequential ELM and simulations were performed to analyze the training and testing accuracy for each learning algorithm. The simulation results showed that the KDD Cup 99 research dataset thus preprocessed was able to test and train the Basic ELM0 and the Online Sequential ELM to a high degree of accuracy. To resolve the issue of the imbalance of the records in the KDD Cup 99, future work should focus on creating additional records to counter the imbalance while retaining useful information. Moreover, to further validate the dataset thus preprocessed, it should be tested using a more diverse range of learning algorithms.

References

[1] Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. *IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, 2009.

[2] Quinlan, J.R. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 1993, 1-38.

[3] John, G.H. and Langley, P. Estimating continuous distributions in Bayesian classifiers. *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995.

[4] Kohavi, R. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[5] Breiman, L. Random Forests. *Machine Learning* **45** (1) (2001) 5–32.

[6] Aldous, D. The continuum random tree. *The Annals of Probability*, 1991, 1–28.

[7] Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E. and Suter, B.W. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks* **1** (4) (1990) 296-298.

[8] Chang, C. and Lin, C. LIBSVM: a library for support vector machines, 2001.

[9] Bolon-Canedo, Verónica, N.S. and A. Alonso-Betanzos. A combination of discretization and filter methods for improving classification performance in KDD Cup 99 dataset. *International Joint Conference on Neural Networks*, 2009.

[10] Pervez, M.S. and Dewan, M.F. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. *8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2014.

[11] Das, A. and Sathya, S.S. A fuzzy approach to feature reduction in KDD intrusion detection dataset. *Third International Conference on Computing Communication & Networking Technologies (ICCCNT)*, 2012.

[12] Chandolikar, N.S. and Nandavadekar, V.D. Efficient algorithm for intrusion attack classification by analyzing KDD Cup 99. *Ninth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2012.

[13] Chandrasekhar, A.M. and Raghuveer, K. Confederation of FCM clustering, ANN and SVM techniques to implement hybrid NIDS using corrected KDD cup 99 dataset. *International Conference on Communications and Signal Processing (ICCSP)*, 2014.

[14] Portnoy, L., Eskin, E. and Stolfo, S. Intrusion detection with unlabeled data using clustering. *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, 2001.

[15] Leung, K. and Leckie, C. Unsupervised anomaly detection in network intrusion detection using clusters. *Proceedings of the Twenty-eighth Australasian conference on Computer Science* **38** (2005) 333– 342,

[16] Axelsson, S. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)* **3** (3) (2000) 186–205.

[17] Gaffney Jr, J. and Ulvila, J. Evaluation of intrusion detectors: A decision theory approach. *Proceedings of IEEE Symposium on Security and Privacy, (S&P)*, 2001, 50–61.

[18] Di Crescenzo, G., Ghosh, A. and Talpade, R. Towards a theory of intrusion detection. *Lecture notes in computer science* **3679** (2005) 1-267.

[19] Cardenas, A., Baras, J. and Seamon, K. A framework for the evaluation of intrusion detection systems. *Proceedings of IEEE Symposium on Security and Privacy, (S&P)*, 2006, 1-15.

[20] Gu, G., Fogla, P., Dagon, D., Lee, W. and Skori´c, B. Measuring intrusion detection capability: An information-theoretic approach. *Proceedings of ACM Symposium on Information, computer and communications security (ASIACCS06)*, 2006, 90–101.

[21] Mahoney, M. and Chan, P. An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection. *Lecture Notes in Computer Science*, 2003, 220–238.