Feature Selection and Intrusion classification in NSL-KDD Cup 99 Dataset Employing SVMs

Muhammad Shakil Pervez, and Dewan Md. Farid
Department of Computer Science and Engineering,
United International University, Bangladesh
mshakilpervez@gmail.com, dewanfarid@cse.uiu.ac.bd

***Abstract*—Intrusion is the violation of information security policy by malicious activities. Intrusion detection (ID) is a series of actions for detecting and recognising suspicious actions that make the expedient acceptance of standards of confidentiality, quality, consistency, and availability of a computer based network system. In this paper, we present a new approach consists with merging of feature selection and classification for multiple class NSL-KDD cup 99 intrusion detection dataset employing support vector machine (SVM). The objective is to improve the competence of intrusion classification with a significantly reduced set of input features from the training data. In supervised learning, feature selection is the process of selecting the important input training features and removing the irrelevant input training features, with the objective of obtaining a feature subset that produces higher classification accuracy. In the experiment, we have applied SVM classifier on several input feature subsets of training dataset of NSL-KDD cup 99 dataset. The experimental results obtained showed the proposed method successfully bring 91% classification accuracy using only three features and 99% classification accuracy using 36 features, while all 41 training features achieved 99% classification accuracy.**

***Index Terms*—classification, feature selection, intrusion detection, NSL KDD Cup 99 dataset, support vector machines.**

I. Introduction

In machine learning and data mining, multi-class classification is the problem of classifying instances into more than two classes [1], [2]. Since many classification methods have been developed specifically for supervised learning, multi-class classification often requires the combined use of multiple classifiers [3], [4]. The main objective of the classification task is to classify the training or test instances after having seen a set of training data. In real life, learning from multi-class data for the purpose of machine learning and data mining is very active research areas for two main reasons: (a) information retrieval from existing multi-label class data is closely compacted in real-life application domains such as intrusion detection, medical diagnosis, radar signal classification, astronomical and medical diagnosis, weather predictions, economical changes, business intelligence and marketing, and (b) multi-label learning includes challenging research issues. Multi-label class data deals with N number of class values. Each group of class value contains a set of data instances with N number of input features [6], [7], [8]. Some time input features having strong relationship among each other with the association with class value. Intrusion detection system (IDS) is a security management tool for detecting threats in computer networks. An IDS collects data from various sources within a computer network and then analysis the collected data to find the attacks that include both intrusions (attacks from outside) and misuse (attacks from within) [5], [9].

Feature selection is a form of search in the training data. It selects a subset of input features d from a total of D original input features in the training data by using an optimisation of scientific theorem to improve the classification accuracy of a learning classifier [3]. In general terms feature selection is the process of searching through the subsets of features in training data, and tries to find the best one. In complex classification area like IDS, feature selection is really necessary as irrelevant and redundant input features in training data make a complex classification model and also reduce the classification rate. In most cases, usually two methods are used to deal with input feature selection from training datasets: (a) filter methods, and (b) wrapper methods. Filter methods select the sub-sets of input features based on the basic characteristics of the training data. It is totally free from the learning classifier which has been used. Filter methods have the ability to handle big training data with large number of input features. Generally, It is less computationally expensive than the wrapper methods. On the other side, wrapper methods involve optimising a learning classifier as part of the feature selection process. It some time gives better results at comparing to filter methods. But wrapper models are very complex and time consuming than the filter methods. Also wrapper methods are not applicable at some of the application domains with some datasets.

In this paper, we proposed a feature selection algorithm by employing the filter method with support vector machine (SVM) classifier on NSL-KDD Cup 99 intrusion detection dataset for multi-class intrusion classification tasks. SVM is a supervised learning model in machine learning and data mining. It analyses data and recognises patterns in data that is used for classification and regression analysis. SVM is commonly used for classification, regression, transduction, novelty detection, and semi-supervised learning. We identified the important input feature set and remove the irrelevant input feature set from the NSL-KDD Cup 99 intrusion detection dataset by applying filter method. And then applied SVM classifier on selected input feature sets to find the best feature with high classification rate. In the experiment, we have applied SVM classifier on several feature subsets of training dataset of NSL-KDD cup 99 dataset. The results obtained showed the proposed method achieved 91% classification accuracy using only three features and 99% classification accuracy using 36 features, while all 41 training features achieved 99% classification accuracy.

The rest of the paper is organised as follows. Section II presents the related works on feature selection and classification in intrusion detection. Section III discusses the support vector machine classifier. Section IV presents the proposed feature selection algorithm in detail. Section V describes the

978-1-4799-6399-7/14/$31.00 2014 IEEE

NSL-KDD Cup 99 dataset and experimental results. Finally, section VI concludes with directions for future work.

II. Related Works

In supervised learning, we use feature selection where we get a subset of features in order to get higher classification accuracy. From original total features from the training set, we remove the irrelevant input features. When we work with the application domains with large number of input dimensions like streaming data environment, such as network traffics monitoring, fraud detection banking sector, weather forecasting, analysis the trend of stock markets and intrusion detection.

In 2010, Farid et al. [11], [12], [13] proposed a learning algorithm based on decision tree (DT) to find the important input feature set from training data for intrusion detection systems. This method finds the important features in training data with a decision tree constructed by employing ID3 and C4.5 decision tree learning algorithms. The weight of the input features are assigned by a value (the minimum depth at which the feature is tested in the decision tree). The depth of root node of the decision tree is one. The weight for a feature is set to, where d is the minimum depth of the decision tree at which the feature is tested in the tree. The weights of the features that do not exist in the decision tree are assigned to zero.

In 2003, Mukkamala & Sung [15] employed feature selection for intrusion detection applying neural networks (NNs) and support vector machines (SMVs) to rank the input features according with each specific class label. In 2009, Yang et al. [14] presented a wrapper-based input feature selection method to discover the most important input features from the training data applying random mutation hill climbing technique, and then employs linear support vector machine (SVM) to assist the performance of selected subset of input features.

III. Support Vector Machines

Support Vector Machines (SVMs) is a powerful supervised learning algorithm that already applied in many real-world applications such as pattern recognition, text and image classification, hand-writing recognition, and bioinformatics analysis. The SVM classification is based on the concept of decision boundaries. A decision boundary separates a set of instances having different class values between two groups. It supports both binary and multi-class classifications. In order to get result for recognising patterns from training data, each instance belongs to one of two classes. SVM classification builds a learning model that assigns new instances into one class, based on a non-probabilistic binary linear classifier. SVM represents the training instances in space, as the instances of the separate classes are divided by a clear gap (wide as possible). Then the test instances are assigned into that same space and classified to the class based on which side of the gap the test instance placed on. Now a day, SVMs are used for linear and non-linear classification. It efficiently perform a non-linear classification employing the kernel trick for mapping high-dimensional input features. Figure 1 shows an illustration of a linearly separable two-class classification with the two possible linear classifiers.

A. Vector Inner Product (VIP)

Let us consider, u and v are the two dimensional vector, $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ and $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$.

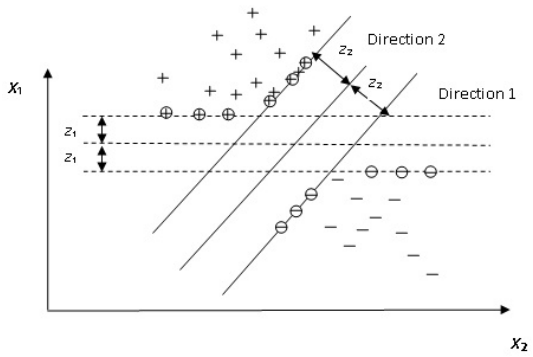

Fig. 1. A linearly separable two-class classification with the two possible linear classifiers.

$$u^T v = [u_1 u_2] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = v^T u = p. \parallel u \parallel \tag{1}$$

In equation 1, $\parallel u \parallel$ is the length of the vector u, and p is the signed length of projection of the vector v onto the vector u.

$$\parallel u \parallel = \sqrt{{u_1}^2 + {u_2}^2} \tag{2}$$

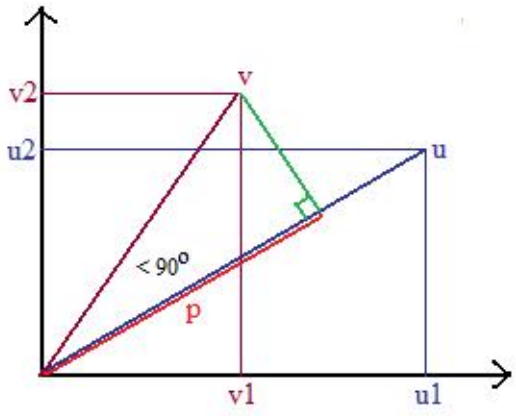

Fig. 2. Computing the VIP between u and v, angle is less than 90^o.

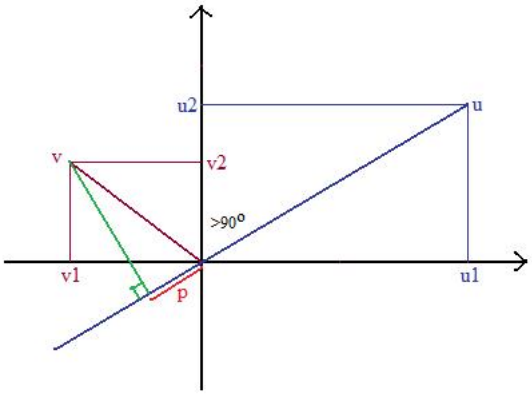

Fig. 3. Computing the VIP between u and v, angle is greater than 90^o.

$$u^T v = p. \parallel u \parallel \tag{3}$$

$$u^T v = u_1 v_1 + u_2 v_2 \tag{4}$$

So, for computing the vector inner product employing equation 3 and 4.

$$u^T v = v^T u \tag{5}$$

$$u^T v = p. \parallel u \parallel = v^T u = u_1 v_1 + u_2 v_2 \leftarrow p \tag{6}$$

In equation 6, u_1 is a row number and p is also row number. The angle between u and v is greater than 90^o, where $p < 0$. Inner product of two vector can be negative, if the angle is greater than 90^o. Now, we are going to apply this properties (*vector inner product*) to try to understand *SVM* optimisation.

B. SVM Decision Boundary

SVM try to minimise the θ in the following equation.

$$min_\theta \frac{1}{2} \sum_{j=1}^{n} (\theta_j)^2 \tag{7}$$

From equation 1, we can write $\theta^T x^i \geq 1$ if $y^i = 1$ and $\theta^T x^i \leq -1$ if $y^i = 0$. Now simplification is $\theta_0 = 0$, and $n = 2$, number of feature. So, we have only two feature x_1 and x_2.

$$min_\theta \frac{1}{2} \sum_{j=1}^{n} (\theta_j)^2 = \frac{1}{2}({\theta_1}^2 + {\theta_2}^2) = \frac{1}{2}(\sqrt{{\theta_1}^2 + {\theta_2}^2})^2 = \frac{1}{2} \parallel \theta \parallel^2 \tag{8}$$

So, all SVM doing in the optimisation objective is to minimising the $\parallel \theta \parallel$ of the length of the parameter vector θ. Therefore,

$$\theta^T x^i = p^i. \parallel \theta \parallel \tag{9}$$

$$\theta^T x^i = \theta_1 {x_1}^i + \theta_2 {x_2}^i \tag{10}$$

Therefore, $min_\theta \frac{1}{2} \sum_{j=1}^{n} (\theta_j)^2 = \frac{1}{2} \parallel \theta \parallel^2$, where $p^i. \parallel \theta \parallel \geq 1$, if $y^i = 1$, and $p^i. \parallel \theta \parallel \leq -1$, if $y^i = 0$. And p^i is the projection of x^i onto the vector θ.

IV. Proposed Feature Selection Algorithm

Given a training dataset, $D = \{A_1, A_2, \cdots, A_n\}$, of n number of input features the proposed feature selection algorithm build a classifier, M_* using all input features in the training dataset, D. Then the algorithm remove one input feature from the training dataset and build another classifier, M_i. If the classification accuracy of $M_i \geq M_*$ then the algorithm considers the new set of features. Otherwise, the new set of features is stored in a database. By this way, we consider all the input features from A_n to A_1 for selecting the sub-set of features in the training dataset. Algorithm 1 outlines the proposed feature selection algorithm used in this paper.

V. Experiments

This section describes the NSL-KDD Cup 99 dataset, experimental environments, and presents the evaluation results for the proposed method.

Algorithm 1 Feature Selection Algorithm

Input: $D = \{A_1, A_2, \cdots, A_n\}$ // A set of input features.
Output: A sets of features, S.
Method:

```
S_1 = {A_1, A_2, ..., A_n};
S = {S_1};
build a classifier, M_* using S, and find the classification accuracy;
s = 1;
for i = n − 1 to 1 do
    create a set of features, S_i with A_1, ..., A_i;
    build a classifier, M_i using S_i, and find the classification accuracy;
    if classification accuracy, M_i ≥ M_* of S then
        S = {S_i};
    else
        if S_i ≥ threshold_value then
            s = s + 1;
            S = S ∪ S_i;
        end if
    end if
    i = i − 1;
end for
```

A. The NSL-KDD Cup 99 dataset

The performance of the proposed algorithm is tested on NSL KDD Cup 99 benchmark dataset from UCI machine learning repository [16]. NSL-KDD Cup 99 dataset is the new version of the KDD Cup 99 dataset. NSL-KDD Cup 99 dataset solves some of the limitations of the KDD Cup 99 dataset. The KDD 1999 cup benchmark intrusion detection dataset was applied in the 3^{rd} International Knowledge Discovery and Data Mining Tools Competition. It was a model capable of identifying characteristic between intrusions and normal connections for making a network intrusion detector. In NSL-KDD Cup 99 dataset, each instance entitled with features of a class of network data. Each class is labeled with either normal or attack. The classes in NSL-KDD Cup 99 dataset are grouped into five main classes: (a) Normal, (b) Denial of Service (DoS), (c) Remote to User (R2L), (d) User to Root (U2R), and (e) Probing (Probe). Each instance in NSL-KDD Cup 99 dataset is a network connection, which have total 41 input features (either discrete or continuous values). The features in NSL-KDD Cup 99 dataset are divided into three groups: (a) the basic input features of network connection including some flags in TCP connections, duration, prototype, number of bytes from source IP addresses or from destination IP addresses, and service, (b) the content input features of network connections, and (c) the statistical input features that are computed either by a time window or a window of certain kind of connections. Table I, II, and III describes the datasets in details.

TABLE I
NSL KDD Cup 99 dataset descriptions

Dataset	No of Attribute	Types of Attribute	Total Instances	Class values
Training 100%	41	Real & Nominal	1,25,973	23
Training 20%	41	Real & Nominal	25,192	23
Testing	41	Real & Nominal	22,544	40

TABLE II
NSL KDD CUP 99 DATASET DESCRIPTIONS

Main Attacks	22 Attacks Classes
DoS	back, land, neptune, pod, smurf, teardrop
R2L	ftp_write, guess_passwd, imap, multhop, phf, spy, warezclient, warezmaster
U2R	buffer_overflow, perl, loadmodule, rootkit
Probing	ipsweep, nmap, portsweep, satan

TABLE III
INSTANCES IN EACH CLASS IN TRAINING DATASETS

Class Value	Training 100%	Training 20%
normal	67343	13449
back	956	196
land	18	1
neptune	41214	8282
pod	201	34
smurf	2646	529
teardrop	892	188
fp_write	8	1
guess_passwd	53	10
imap	11	5
multhop	7	2
phf	4	2
spy	2	1
warezclient	890	181
warezmaster	20	7
buffer_overflow	30	6
loadmodule	9	1
perl	3	3
rootkit	10	4
ipsweep	3599	710
nmap	1493	301
portsweep	2931	587
satan	3633	691

The strong advantage of NSL-KDD Cup 99 dataset is that the training and testing instances are reasonable, so it becomes affordable to execute the experiments on the total set of training and testing dataset without the need to randomly select a small portion of dataset. The NSL-KDD Cup 99 dataset has the following superior positions over the original KDD 99 dataset:

- NSL-KDD Cup 99 dataset does not include redundant training instances that confuse the learning classifiers.
- There is no duplicate instances in the testing data of NSL-KDD Cup 99 dataset. So, the mining models generated by learning classifiers are biased free.
- The training instances from each attack group is opposite proportional to the percentage of instances in the original KDD dataset. So, the classification accuracy of learning algorithms vary in a wider range that makes it more efficient to have an accurate judgement of different learning methods.
- The number of training and testing instances are reasonable to run the experiments without randomly selected small portion of training instances.

KDD 99 dataset is a perfect dataset to test intrusion detection because it has the large number of redundant instances, which causes the learning classifiers to be biased towards the frequent instances, and thus prevent them from learning unfrequent instances that are usually more harmful to computer networks. As User to Root (U2R) and Remote to User (R2L) attacks are made in these approaches. The existence of these repeated instances in the testing dataset will cause the evaluation results to be biased by the classifiers that have better classification rates on the frequent instances.

B. Experimental setup

The experiments were carried out by a MacBook Pro with Retina display with 2.7 GHz quad-core Intel Core i7 Processor and 16 GB of RAM. The proposed algorithm was implemented in Java using NetBeans IDE 7.3.1, which is the first IDE providing support for JDK 7 and Java EE 6 (http://netbeans.org/index.html). The basic code for the SVM classifier is adopted from Weka3. Weka3 is an open source data mining software for machine learning applications [17]. The proposed algorithm was tested by applying the classification accuracy, error rate, true positive, false positive, precision, recall, and f-measure on NSL-KDD Cup 99 dataset.

The performance measures of a classifier that how accurate the classifier predicting the class label of instances (both training and testing instances). To compute the performance of learning classifiers we need to know the four terms: (1) True positives, TP (the positive instances are correctly classified by the learning algorithm), (2) True negatives, TN (the negative instances are correctly classified by the learning algorithm), (3) False positives, FP (the negative instances are misclassified as positive by the learning algorithm), and (4) False negatives, FN (the positive instances are misclassified as negative by the learning algorithm) The classification rate of a learning algorithm on a given test data is the percentage of test instances that are correctly classified by the learning algorithm, which is measured by Equation 11.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

The *error rate* of a learning algorithm on a given test data is the percentage of test set instances that are misclassified by the learning algorithm, which is shown in Equation 12.

$$errorRate = \frac{FP + FN}{P + N} \tag{12}$$

The *sensitivity* is referred to as the *true positive* rate (i.e., the proportion of positive instances that are correctly identified by learning algorithm), shown in Equation 13.

$$sensitivity = \frac{TP}{P} \tag{13}$$

The *specificity* is referred to as the *true negative* rate (i.e., the proportion of negative instances that are correctly identified by learning algorithm), shown in Equation 14. A good learning algorithm would be described as 100% sensitivity and 100% specificity rate.

$$specificity = \frac{TN}{N} \tag{14}$$

The *precision* and *recall* measures are also frequently used for classification. *Precision* can be thought of as a measure of exactness (i.e., what percentage of instances labeled as positive are actually such), whereas recall is a measure of completeness (what percentage of positive instances are labeled as such). If recall seems familiar, thats because it is the same as sensitivity (or the true positive rate).

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P} \quad (16)$$

TABLE IV
EVALUATION MEASURES - CLASSIFIER PERFORMANCE

Measure	Formula
classification Accuracy	$\frac{TP+TN}{P+N}$
Error rate/ Misclassification rate	$\frac{FP+FN}{P+N}$
Sensitivity/ True positive rate/ Recall	$\frac{TP}{P}$
Specificity/ True negative rate	$\frac{TN}{N}$
Precision	$\frac{TP}{TP+FP}$
F-Measure	$\frac{2*precision*recall}{precision+recall}$

The 10-fold cross validation divides a given training dataset into 10 sub-datasets of size N/10. Then used nine datasets for training and used the remaining one dataset for testing. The process repeats 10 times and take a mean of classification accuracy. For classification, the classification rate calculate is the overall number of correct classifications from the k iterations, divided by the total number of instances in the dataset.

C. Results and discussion

Tables V and VI tabulate the performances of the SVM on NSL KDD Cup 99 datasets. Figures 4 to 6 show the classification accuracy of the SVM classifier on NSL KDD Cup 99 datasets with selected features on 100% training data, 10 fold cross-validation on 100% training data and 100% testing data respectively. Table VI shows that the proposed method achieved 91% classification accuracy using only three features and 99% classification accuracy using 36 features, while all 41 training features achieved 99% classification accuracy on full training NSL KDD 99 Cup dataset.

TABLE V
PERFORMANCE OF THE SVM CLASSIFIER ON NSL KDD CUP 99 DATASETS

Datasets	Training (100%)	Training (20%)	Testing
classification rate	99.01 %	98.475 %	82.37 %
Error rate	0.98 %	1.524 %	17.62 %
TP Rate (Weighted Avg.)	0.99	0.985	0.82
FP Rate (Weighted Avg.)	0.007	0.013	0.15
Precision (Weighted Avg.)	0.99	0.984	0.74
Recall (Weighted Avg.)	0.99	0.985	0.82
F-Measure (Weighted Avg.)	0.99	0.982	0.77

TABLE VI
CLASSIFICATION ACCURACY (%) OF THE SVM CLASSIFIER ON NSL KDD CUP 99 DATASETS USING FEATURES SETS

SVM with feature subsets	Training (100%)	10 fold CV on Training (100%)	Testing Data
41 Features	99.01	98.96	82.37
36 Features	99.01	98.95	82.38
29 Features	98.34	98.29	82.24
17 Features	97.92	97.92	82.45
14 Features	97.73	97.68	82.68
9 Features	95.48	95.44	81.38
6 Features	95.07	95.01	81.03
5 Features	93.77	93.73	80.53
4 Features	93.33	93.31	80.03
3 Features	91.01	90.99	78.85

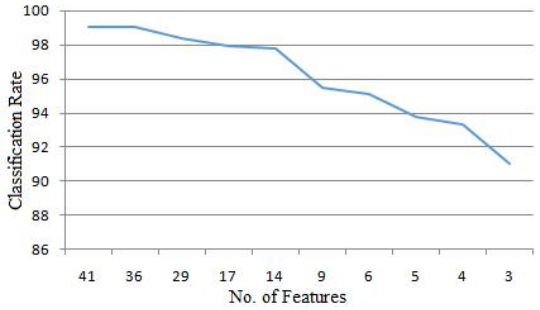

Fig. 4. Classification accuracy employing SVM with selected features on 100% training data.

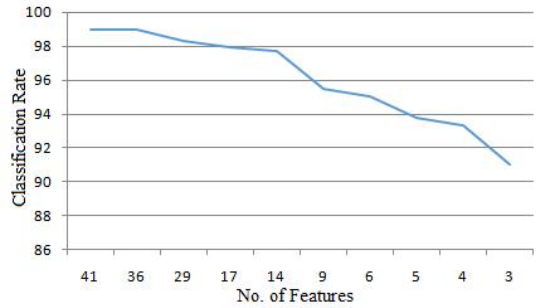

Fig. 5. Classification accuracy employing SVM with selected features using 10 fold cross-validation on 100% training data.

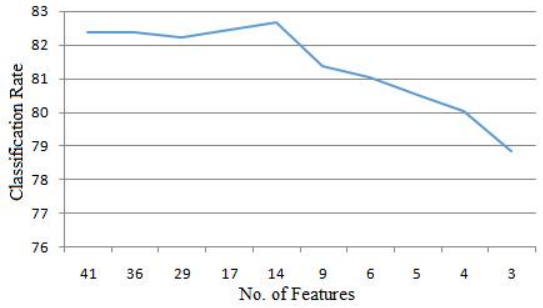

Fig. 6. Classification accuracy employing SVM with selected features on test data.

VI. CONCLUSION

This paper presented an intrusion classification approach based on the combination of feature selection and SVM classifier using NSL-KDD Cup 99 intrusion detection benchmark dataset. The proposed method maintains the classification accuracy of the SVM classifier but it uses a reduced set of input features from training data. The NSL-KDD Cup 99 dataset is a multiple class classification problem that contains normal and attack network connections. In this research, we have applied SVM classifier on several feature subsets of NSL-KDD Cup 99 dataset and the experimental analysis shown that the proposed method achieved 91% classification accuracy using only three input features and 99% classification accuracy using 36 input features, while all 41 input features achieved 99% classification accuracy. In future work, we are planning to ensemble other mining classifiers with SVM to achieve the good classification accuracy of the minority class instances.

REFERENCES

[1] Dewan Md. Farid, Li Zhang, Chowdhury Mofizur Rahman, M.A. Hossain, and Rebecca Strachan, "Hybrid Decision Tree and naïve Bayes classifiers for Multi-class classification Tasks," *Expert Systems with Applications*, Vol. 41, Issue 4, Part 2, March 2014, pp. 1937-1946.

[2] Dewan Md. Farid, Li Zhang, Alamgir Hossain, Chowdhury Mofizur Rahman, Rebecca Strachan, Graham Sexton, and Keshav Dahal, "An Adaptive Ensemble classifier for Mining Concept Drifting Data Streams," *Expert Systems with Applications*, Vol. 40, Issue 15, November 2013, pp. 5895-5906.

[3] Dewan Md. Farid, and Chowdhury Mofizur Rahman, "Mining Complex Data Streams: Discretization, Attribute Selection and classification," *Journal of Advances in Information Technology*, Vol. 4, No. 3, August 2013, pp. 129-135.

[4] Dewan Md. Farid, and Chowdhury Mofizur Rahman, "Assigning Weights to Training Instances Increases classification Accuracy," *International Journal of Data Mining & Knowledge Management Process*, Vol. 3, No. 1, January 2013, pp. 13-25.

[5] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, "Mining Complex Network Data for Adaptive Intrusion Detection," *Advances in Data Mining Knowledge Discovery and Applications*, ISBN 978-953-51-0748-4, INTECH, September 2012, pp. 327-348.

[6] Amit Biswas, Dewan Md. Farid, and Chowdhury Mofizur Rahman, "A New Decision Tree Learning Approach for Novel Class Detection in Concept Drifting Data Stream classification, *Journal of Computer Science and Engineering*, Vol. 14, Issues 1, July 2012, pp. 1-8.

[7] Fauzia Yasmeen Tani, Dewan Md. Farid, and Mohammad Zahidur Rahman, "Ensemble of Decision Tree classifiers for Mining Web Data Streams, *International Journal of Applied Information Systems*, Vol. 1, No. 2, January 2012, pp. 30-36.

[8] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, "An Ensemble Approach to classifier Construction based on Bootstrap Aggregation, *International Journal of Computer Applications*, Vol. 25, No. 5, July 2011, pp. 30-34.

[9] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, "Adaptive Intrusion Detection based on Boosting and Nave Bayesian classifier, *International Journal of Computer Applications*, Vol. 24, No. 3, June 2011, pp. 12-19.

[10] A. J. M. Abu Afza, Dewan Md. Farid, and Chowdhury Mofizur Rahman, "A Hybrid classifier using Boosting, Clustering, and nave Bayesian classifier, *World of Computer Science and Information Technology Journal*, Vol. 1, No. 3, April 2011, pp. 105-109.

[11] Dewan Md. Farid, Nouria Harbi, and Mohammad Zahidur Rahman, "Combining Nave Bayes and Decision Tree for Adaptive Intrusion Detection, *International Journal of Network Security & Its Applications*, Vol. 2, No. 2, April 2010, pp. 12-25.

[12] Dewan Md. Farid, Jerome Darmont, and Mohammad Zahidur Rahman, "Attribute Weighting with Adaptive NBTree for Reducing False Positives in Intrusion Detection, *International Journal of Computer Science and Information Security*, Vol. 8, No. 1, April 2010, pp. 19-26.

[13] Dewan Md. Farid, and Mohammad Zahidur Rahman, "Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm, *Journal of Computers*, Vol. 5, No. 1, January 2010, pp. 23-31.

[14] Y. Li, J. L. Wang, Z. H. Tian, T. B. Lu, and C. Young, "Building lightweight intrusion detection system using wrapper-based feature selection mechanisms, *Computer Security*, vol. 28, No. 6, September 2009, pp. 466-475.

[15] S. Mukkamala and A. H. Sung, "Feature selection for intrusion detection with neural networks and support vector machines, *Journal of the Transportation Research Board*, Vol. 1822, 2003, pp. 33-39.

[16] A. Frank, and A. Asuncion, "UCI machine learning repository," *University of California, Irvine*, 2010, http://archive.ics.uci.edu/ml, Accessed 26.08.2014.

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, Vol. 11, No. 1, 2009, http://www.cs.waikato.ac.nz/ml/weka/, Accessed 26.08.2014.