Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 6 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

# A Cost Sensitive Machine Learning Approach for Intrusion Detection

By Adamu Teshome & Dr. Vuda Sreenivasa Rao

*Bahir Dar University, Ethiopia*

*Abstract-* The problems with the current researches on intrusion detection using data mining approach are that they try to minimize the error rate (make the classification decision to minimize the probability of error) by totally ignoring the cost that could be incurred. However, for many problem domains, the requirement is not merely to predict the most probable class label, since different types of errors carry different costs. Instances of such problems include authentication, where the cost of allowing unauthorized access can be much greater than that of wrongly denying access to authorized individuals, and intrusion detection, where raising false alarms has a substantially lower cost than allowing an undetected intrusion. In such cases, it is preferable to make the classification decision that has minimum cost, rather than that with the lowest error rate.For this reason, we examine how cost-sensitive machine learning methods can be used in Intrusion Detection systems. The performance of the approach is evaluated under different experimental conditions and different models in comparison with the KDD Cup 99 winner resultsin terms of average misclassification cost, as well as detection accuracy and false positive ratesthough the winner used original KDD dataset whereas for this research NSL-KDD dataset which is new version of the original KDD cup data and it is better than the original dataset in that it has no redundant data is used.

*Keywords: intrusion detection, data mining, cost sens-itive learning.*

*GJCST-C Classification : H.1.2*

# A Cost Sensitive Machine Learning Approach for Intrusion Detection

Adamu Teshome [α] & Dr. Vuda Sreenivasa Rao [σ]

*Abstract-* The problems with the current researches on intrusion detection using data mining approach are that they try to minimize the error rate (make the classification decision to minimize the probability of error) by totally ignoring the cost that could be incurred. However, for many problem domains, the requirement is not merely to predict the most probable class label, since different types of errors carry different costs. Instances of such problems include authentication, where the cost of allowing unauthorized access can be much greater than that of wrongly denying access to authorized individuals, and intrusion detection, where raising false alarms has a substantially lower cost than allowing an undetected intrusion. In such cases, it is preferable to make the classification decision that has minimum cost, rather than that with the lowest error rate.For this reason, we examine how cost-sensitive machine learning methods can be used in Intrusion Detection systems. The performance of the approach is evaluated under different experimental conditions and different models in comparison with the KDD Cup 99 winner resultsin terms of average misclassification cost, as well as detection accuracy and false positive ratesthough the winner used original KDD dataset whereas for this research NSL-KDD dataset which is new version of the original KDD cup data and it is better than the original dataset in that it has no redundant data is used. For comparison of results of CS-MC4, CS-CRT and KDD winner result, it was found that CS-MC4 is superior to CS-CRT in terms of accuracy, false positives rate and average misclassification costs. CS-CRT is superior to KDD winner result in accuracy and average misclassification costs but in false positives rate KDD winner result is better than both CS-MC4 and CS-CRT classifiers.

*Keywords: intrusion detection, data mining, cost sensitive learning.*

## I. Introduction

The field of intrusion detection has received increasing attention in recent years. One reason is the explosive growth of the Internet and the large number of networked systems that exist in all types of organizations. The increased number of networked machines has led to a rise in unauthorized activity, not only from external attackers but also from internal sources such as disgruntled employees and people abusing their privileges for personal gain [26].Since intrusions take advantage of vulnerabilities in computer systems or use socially engineered penetration techniques,intrusion detection (ID) is often used as another wall of protection. Intrusion detection includes identifying a set of malicious actions that compromise the integrity, confidentiality, and availability of information resources.

Enough data exists or could be collected to allow network administrators to detect any violations. Unfortunately, the data is so volumes, and the analysis process so time consuming that the administrators don't have the resources to go through it all and find the relevant knowledge, save for the most exceptional situations [30].Given the nature of this problem, the possible solution is data mining approach[3], [30].

Data mining approach for intrusion detection techniques generally fall into one of two categories; misuse detection and anomaly detection. In misuse detection, each instance in a data set is labeled as 'normal' or 'intrusion' and a learning algorithm is trained over the labeled data. These techniques are able to automatically retrain intrusion detection models on different input data that include new types of attacks, as long as they have been labeled appropriately. Unlike manual intrusion detection systems, models of misuse are created automatically, and can be more sophisticated and precise than manually created signatures. A key advantage of misuse detection techniques is their high degree of accuracy in detecting known attacks and their variations. Their obvious drawback is the inability to detect attacks whose instances have not yet been observed.

Anomaly detection, on the other hand, builds models of normal behavior, and automatically detects any deviation from it, flagging the latter as suspect. Anomaly detection techniques thus identify new types of intrusions as deviations from normal usage [5]. While an extremely powerful and novel tool, a potential draw-back of these techniques is the rate of false alarms. This can happen primarily because previously unseen (yet legitimate) system behaviors may also be recognized as anomalies, and hence flagged as potential intrusions. Hybrid IDS combine bothmethods and it is better one [22].

The problem with the current researches is that they try to minimize the error rate (make the classification decision to minimize the probability of error) by totally ignoring the cost that could be incurred. However, for many problem domains, the requirement is not merely to predict the most probable class label, since different types of errors carry different costs [10].

*Author α: Lecturer, School of Computing and Electrical Engineering, BIT, Bahir Dar University, Ethiopia.*
*e-mail: adamu_teshome@yahoo.com*
*Author σ : Professor, School of Computing and Electrical Engineering, BIT, Bahir Dar University, Ethiopia.*

For example of such problem in the case of computer network security include authentication, where the cost of allowing unauthorized access can be much greater than that of wrongly denying access to authorized individuals, and intrusion detection, where raising false alarms has a substantially lower cost than allowing an undetected intrusion. In such cases, it is preferable to make the classification decision that has minimum expected cost, rather than that with the lowest error probability [23].

Another very important case is, if class imbalanced datasets occurs but this happens in many real-world applications where the class distributions of data are highly imbalanced [23]. Again, it is assumed that the minority or rare class is the positive class, and the majority class is the negative class. Often the minority class is very small, such as 1% of the dataset. If most traditional (cost insensitive) classifiers are applied on the dataset, they will likely to predict everything as negative (the majority class) [33].

The intrusion data used for this research, KDD data set, which is publicly available and most widely used data set, has class distributions of training and test datasets with different distribution and each attack types has different costs. Statistically, the attacks of U2R and R2L are of the rarest, which makes them very hard to predict. On the other hand, they are the most dangerous types. Once an attacker gains the super user right or successfully remote login, disasters of the whole system are nothing but unavoidable [19].

Comparably, attacks of Probe are not that much dangerous. Although attacks of DOS (denial of service) are massive in the whole original dataset, they impose less danger. This is because the nature of denial of service attacks lies in that they are trying to initiate as many as possible connections to consume the network traffics and server CPU time [19].

Because of the above mentioned reasons cost sensitive learning which considers cost in decision making with acceptable accuracy is better solution for computer network intrusion detection. We used cost sensitive learning algorithms, cost sensitive classification tree CS-CRT and cost sensitive decision tree CS-MC4 algorithms. These algorithms use misclassification cost matrix to minimize the expected cost and for the detection of best prediction. Yet despite its importance, the topic is seldom addressed and researched.

## II. Intrusion Detection, Cost Sensitive Machine Learning and Performance Measure

An intrusion detection system attempts to detect intrusions. In this paper, we focus on network-based systems, i.e., network intrusion detection systems (NIDS) whose primary source of data is network traffic. In contrast, there is host intrusion detection systems (HIDS) which rely on information gathered on individual hosts. Hybrid systems are both network-based and host-based [5], [22].

### *a) Classification of intrusion detection based on different detection method*

#### i. *Misuse Detection*

It is also named signature-based detection, which can transform the information of attack symptom or policy disobeying into state transition-based signature or rule, and such information is stored in signature database. To judge whether or not it is attack, pre-treated case data should be first compared with the signature of signature database, and those conforming to attack signature data can be judged as attack [22]. Its advantage is high detection rate and low false alarm rate for known attacks; however, its detection capacity is low for unknown detection methods, and attack database should be renewed on a regular basis.

#### ii. *Anomaly Detection*

It may establish a profiles for normal behavior of users, which comes from statistics data of users in the former period; when detection is performed, the profiles is compared with actual users' data, if the offset is below threshold value, user's behavior can be considered normal, and it has no intention of attacks; if the offset is above threshold value, user's behavior can be considered abnormal [22]. Anomaly detection is based on an assumption that intruder's behavior is different from normal users' behavior. Detection rate of the method is high, and it is more likely to detect unknown attacks, but misjudgment (false positive) rate is also high.

#### iii. *Hybrid*

It is also possible to include both normal and attack data in the model. Such systems are referred to be hybrid detectors.

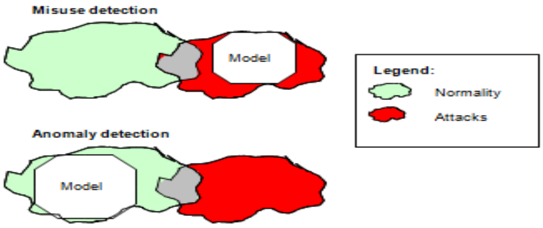

*Figure 1 :* The relation between misuse detection and anomaly detection.

### *b) Architecture*

In general, a NIDS consists of different components.

1. *Collector:* Provides an interface for accessing data that is used by the detection process. For a NIDS, the primary kind of data collector is a network tap. A tap provides access to all raw network packets which cross a particular position of a network.
2. *Detector:* Conducts the actual detection process. The detector is the “brain" of the NIDS. It accesses data provided by collector and storage (see below), and it decides what should trigger an alert.
3. *User Interface:* Reports results to the user, and enables the user to control the NIDS.
4. *Storage:* Stores persistent data required by the detector or the user interface. Such data is either derived by the detector itself or provided externally.
5. *Responder:* Reacts to detected intrusions in order to prevent future damage. Active responses may include dropping the connectivity to the potential attacker or even counter-attacks. A response may be triggered automatically or manually via the user interface.

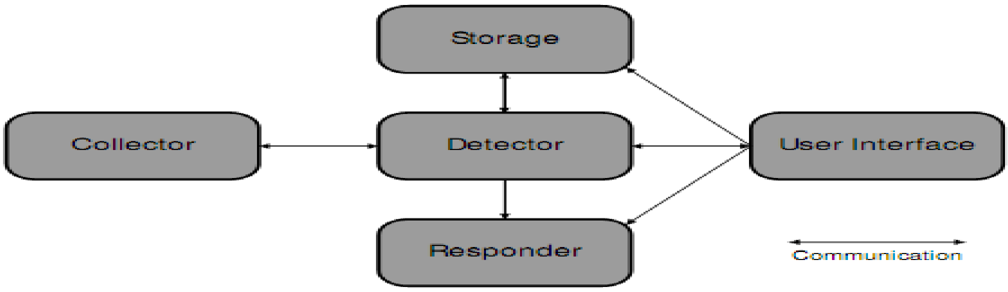

*Figure 2 :* NIDS architecture

## III. Cost Sensitive Machine Learning

Costs are central to statistical decision theory but cost sensitive learning received only modest attention before [19]. Cost sensitive learning is a type of learning in data mining that takes the misclassification costs (and possibly other types of cost) into consideration. The goal of this type of learning is to minimize the total cost. The key difference between cost sensitive learning and cost insensitive learning is that cost sensitive learning treats the different misclassifications differently. Cost insensitive learning does not take the misclassification costs into consideration. The goal of this type of learning is to practice a high accuracy of classifying examples into a set of known classes.

Cost sensitive learning is an extensively used practice in data mining, which assigns different levels of misclassification penalty to each class. Cost sensitive technique has been incorporated into classification algorithms by taking into account the cost information and trying to optimize the overall cost during the learning process.

### *a) Cost Models*

The cost model of IDS devises the total expected cost of the IDS. The cost model depends on the detection performance of the IDS. Misclassification costs false positive (FP, the cost of misclassifying a negative instance into positive) and false negative (FN, the cost of misclassifying a positive instance into negative), and the cost of correct classification is zero. They simply assign FP as the weight to each negative instance, and assign FN as the weight to each positive instance. That is, the weight ratio of a positive instance to a negative instance is proportional to FN/FP [9], [19].

The cost of misclassifying an instance of class i as class j is C(i, j ) and is assumed to be equal to 1 unless specified otherwise(i, i ) = 0 for all i . Any classification tree can have a total cost computed for its terminal node assignments by summing costs over all misclassifications. The issue in cost sensitive learning is to induce a tree that takes the costs into account during its growing and pruning phases [19]. These misclassification cost values can be given by domain experts, or learned via other approaches. In cost sensitive learning, it is usually assume that such a cost matrix is given and known. For multiple classes, the cost matrix can be easily extended by adding more rows and more columns.

*Table 1:* An example of cost matrix for binary classification.

| | Actual negative | Actual positive |
|---|---|---|
| Predict negative | C(0,0), or TN | C(0,1), or FN |
| Predict positive | C(1,0), or FP | C(1,1), or TP |

When optimizing sampling levels to improve overall cost, a logical evaluation criterion is the cost itself. Cost is calculated as follows by taking the assumption $C\ (+|+) = C\ (-|-) = 0$.Cost $= FN_{rate} x\ C\ (-|+) + FP_{rate} x\ C\ (+|-)$. Therefore, once classification occurs in the cross-validation phase, the wrapper or filter calculates the validation cost and uses this information to select optimal sampling levels. This approach is dependent on a priori knowledge of the cost relationship between classes [10]. Cost sensitive learning algorithm which directly introduce and make use of misclassification costs into the learning algorithms are ICET, EG2, CS-ID3, CS-MC4 and CS-CRT. The cost sensitive learning algorithms that are used for this research are cost sensitive learning algorithms i.e. cost sensitive decision tree CS-MC4 which is the cost sensitive version of C4.5and CS-CRT the cost sensitive version of C-RT classification tree that use misclassification cost matrix to minimize the expected cost and for the detection of best prediction.

### b) *Performance Measures*

In general, intrusion detection systems require high detection rate, low false alarm rate and lower average misclassification cost. The followings are commonly used performance measures

i. *Error Rate*

The error rate, which is only an estimate of the true error rate and is expressed to be a good estimate, if the number of test data is large and representative of the population and is defined as [68]:

$$\text{Error Rate} = \frac{TotalTestsamples - TotalCorrectlyclassifiedsamples}{TotalTestsamples} \times 100\%$$

ii. *Accuracy*

Overall Classification accuracy (OCA) is the most essential measure of the performance of a classifier. It determines the proportion of correctly classified examples in relation to the total number of examples of the test set i.e. the ratio of true positives and true negatives to the total number of examples.

From the confusion matrix, we can say that [9] accuracy is the percentage of correctly classified instances over the total number of instances in total data, namely the situation TP and TN, thus accuracy can be defined as follows:

$$\text{Accuracy} = \frac{Tota\ln umberofcorrectlyclassifiedsamples}{Tota\ln umberoftestsamples} \times 100\%$$

Or

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

iii. *Detection Rate*

Detection rate refers to the proportion of attack detected among all attack data, namely, the situation of TP, thus detection rate is defined as follows [20]:

$$\text{DetectionRate} = \frac{no.ofsamplesclassifiedcorrectly}{no.ofsamplesusedfortesting} \times 100\%$$

Or

$$\text{Detection Rate} = \frac{TP}{TP + FN} \times 100\%$$

iv. *False Positive Rate*

Another name is False Alarm Rate (FAR) measures the number of misclassified positive instances in relative to the total number of misclassified instances. Can be expressed also the proportion that normal data is falsely detected as attack behavior, namely, the situation of FP, thus false alarm rate is defined as follows [20]:

$$\text{False Positive Rate} = \frac{tota\ln o.ofmissclassifiedsamples}{tota\ln o.oftestsamples} \times 100\%$$

Or

$$\text{False alarm rate} = \frac{FP}{FP + TN} \times 100\%$$

Recall and Precision are two widely used metrics employed in applications where successful detection of one of the classes is considered more significant than detection of the other classes [].

v. *Average Misclassification Cost (AMC)*

$$\text{AMC} = 1/N \sum_{i=1}^{5} \sum_{j=1}^{5} \text{CM (I, J)*C (I, J)}$$

Where CM corresponds to confusion matrix, C corresponds to the cost matrix, and N represents the number of patterns tested [10].

## IV. Data Collection and Preparation

NSL-KDD data set which is the new version of KDD Cup 99 dataset and the only publicly available [21] and the widely used data set for intrusion detection have been used for the experimental purpose. The process of data cleaning and preparation is highly dependent on the specific machine learning algorithm and software chosen and algorithms used for the machine learning task. The researcher attempted to prepare the data according to the requirements of the Tanagra which is powerful, user friendly and freely available for noncommercial purpose machine learning software [11] and according to the requirements of CS-MC4 and CS-CRT algorithms by consulting different literatures.

## V. Training and Testing Models

The Intrusion detection models were developed using cost sensitive classification tree CS-CRT and cost sensitive decision tree CS-MC4 algorithms on full training NSL-KDD dataset using a powerful machine learning and data mining Tanagra tool and then Full testing dataset of NSL-KDD passed through the developed models to detect the intrusions and find the detection error rates, precision, false positives rate, average misclassification cost and accuracy of the detection models but for comparison of models we used mainly average misclassification cost, false positives rate and accuracy of detection.

## VI. Experimentation

We used data mining software tool known as Tanagra version 1.4.34 available freely at [11]. The software is a GUI based software and easy to use. Tanagra is capable of classifying large volumes of data within a second depending on the speed and specification of computer processor. All experiments were performed using an Intel Core 2 Duo Processor 2.16 GHz processor with 4 GB of RAM and implemented on a Vista Windows operating system.

### a) *NSL-KDD Intrusion Detection Dataset:*

We have used NSL-KDD intrusion dataset which is available at [21] and it is a dataset suggested to solve some of the inherent problems of the original KDD Cup 99 dataset [13,20]. The NSL-KDD dataset has the following advantages over the original KDD Cup 99 dataset.

1. It does not include redundant records in the training set, so the classifiers will not be biased towards more frequent records.
2. There are no duplicate records in the proposed test sets, therefore the performance of the learners are not biased by the methods which have better detection rates on the frequent records.
3. The number of selected records from each difficulty level group is inversely proportional to the percentage of the records in the original KDD dataset. As a result the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
4. The number of records in the training and testing sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion.
5. Statistical observations, one of the most important deficiencies in the KDD dataset is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records and thus prevent them from learning unfrequent records which are usually more harmful to networks such as U2R and R2L attacks.

Table2 and Table3 shows the statistics of the redundant records in the KDD Cup 99 training and testing datasets.

*Table 2 :* Statistics of redundant records in the KDD training dataset.

| Original | Records | Distinct Records | Reduction Rate |
|---|---|---|---|
| Attacks | 3,925,650 | 262,178 | 93.32% |
| Normal | 972,781 | 812,814 | 16.44% |
| Total | 4,898,431 | 1,074,992 | 78.05% |

*Table 3 :* Statistics of redundant records in the KDD testing dataset

| Original | Records | Distinct Records | Reduction Rate |
|---|---|---|---|
| Attacks | 250,436 | 29,378 | 88.26% |
| Normal | 60,591 | 47,911 | 20.92% |
| Total | 311,027 | 77,289 | 75.15% |

### a) *Evaluation Metrics*

We have used the cost matrix defined for the KDD Cup 99 Dataset [9] which is shown in Table 4.

| Category | Normal | Probe | DOS | U2R | R2L |
|---|---|---|---|---|---|
| Normal | 0 | 1 | 2 | 2 | 2 |
| Probe | 1 | 0 | 2 | 2 | 2 |
| DOS | 2 | 1 | 0 | 2 | 2 |
| U2R | 3 | 2 | 2 | 0 | 2 |
| R2L | 4 | 2 | 2 | 2 | 0 |

## VII. Experimentation and Result

The experimentations have two major parts; the first one is experimentation on CS-MC4 and CS-CRT using all the 41 attributes and the second is experimentation on CS-MC4 and CS-CRT using information gain (IG) feature (attribute) selection method. Comparative discussions of each approaches used with the KDD 99 winner results are given.

Table 5 summarizes comparison results of detection accuracy (which refers to the proportion that a type of data is correctly classified) of normal and 4 different attacks (i.e. Probe, Dos, U2R, R2L) based on CS-MC4 and CS-CRT classifiers using 24, 30, 37, and all the attributes in comparison with KDD winner results.

*Table 5 :* Performance comparison of testing result for five-class classification.

| | Normal | | DOS | | R2L | | Probe | | U2R | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Cs-mc4 | Cs-crt | Cs-mc4 | Cs-crt | Cs-mc4 | Cs-crt | Cs-mc4 | Cs-crt | Cs-mc4 | Cs-crt |
| All attributes | 98.71 | 92.91 | 99.85 | 97.44 | 91.69 | 61.46 | 98.80 | 99.66 | 72.50 | 17.50 |
| 24 attributes | 98.65 | 93.56 | 99.81 | 98.01 | 89.53 | 45.68 | 98.85 | 97.94 | 52.50 | 10.00 |
| 30 attributes | 98.61 | 98.25 | 99.81 | 99.64 | 91.20 | 88.87 | 98.85 | 99.04 | 77.50 | 20.00 |
| 37 attributes | 98.70 | 92.83 | 99.81 | 97.44 | 91.69 | 61.46 | 98.80 | 99.66 | 65.00 | 17.50 |
| KDD winner | 99.5 | | 97.1 | | 8.4 | | 83.3 | | 13.2 | |

It is evident from this Table that:

1. *For Normal:* CS-MC4 classifiers outperform their CS-CRT counterparts in all cases but when only 30 attributes are used, the difference between the accuracy of CS-MC4 and CS-CRT became minimal i.e. 0.36%. For this type of class, in all the cases the accuracy of the KDD winner result is better than both CS-MC4 and CS-CRT classifiers.
2. *For Probe attack:* the accuracy of CS-MC4 is better than that of CS-CRT when only 24 attributes are used, but in other circumstances CS-CRT outperforms CS-MC4. For this type of attack, in all the cases the accuracy of both CS-MC4 and CS-CRT is better than the KDD winner result.
3. *For Dos attack:* CS-MC4 classifiers outperform their CS-CRT counterparts in all cases but when only 30 attributes are used, the difference between the accuracy of CS-MC4 and CS-CRT became minimal, i.e. 0.17%. For this type of attack, in all the cases the accuracy of both CS-MC4 and CS-CRT is better than the KDD winner result.
4. *For U2R attack:* the accuracy of CS-MC4 classifiers is better than their CS-CRT counterparts in all cases. For this type of attack, in all the cases the accuracy of both CS-MC4 and CS-CRT is better than the KDD winner result except when only 24 attributes are used for CS-CRT classifier.
5. *For R2L attack:* accuracy of CS-MC4 classifiers is better than their CS-CRT counterparts in all cases. For this type of attack, in all the cases the accuracy of both CS-MC4 and CS-CRT are by far better than the KDD Cup 99 winner result.

In general, it is evident from the above table that in terms of accuracy, CS-MC4 outperformed the CS-CRT. As it can be seen from the result, the feature selection method used (i.e. information gain value) did not increase the accuracy of CS-MC4 classifier. Even though CS-MC4 classifier achieved the same result using all the attributes (41 attributes) and 37 attributes (at information gain value greater than 0), the reduction of the attributes from 41 to 37 could increase the speed of the classifier and reduce the space required. CS-CRT achieved a better result when only 30 attributes (information gain value greater than and equal to 0.011) are used relative to the other cases.

So, feature selection using information again value achieved better result for CS-CRT classifier from 41 attributes to 30 attributes but for CS-MC4 classifier from 41 attributes to 37 attributes. Results which are achieved in this work are compared with the results obtained by KDD winner results as shown in table 4. As it can be seen, the accuracy of KDD Winner is only better in normal, but it is far worse than CS-MC4 and CS-CRT in U2R and R2L attacks; this might be because of the data used for this research is NSL-KDD intrusion dataset (which is new version of the original KDD cup data and it is better than the original dataset in that it has no redundant data) but for the KDD winner the original KDD cup dataset is used.

Table 6 summarizes the efficiency of information gain value for feature selection and the performance comparison of CS-MC4, CS-CRT classifier and KDD winner result based on overall accuracy, false positives rate and average misclassification cost using 24, 30, 37 and all the attributes for 5 attack classes on NSL-KDD dataset.

*Table 6 :* Performance comparison of different methods.

| | Overall accuracy (%) | | False positives rate (%) | | Average Misclassification Cost | |
|---|---|---|---|---|---|---|
| | CS-MC4 | CS-CRT | CS-MC4 | CS-CRT | CS-MC4 | CS-CRT |
| All attributes | 98.9 | 94.2 | 1.3 | 7.1 | 0.0199 | 0.0895 |
| 24 attributes | 98.8 | 94.1 | 1.4 | 6.4 | 0.0232 | 0.0982 |
| 30 attributes | 98.8 | 98.4 | 1.4 | 1.7 | 0.0201 | 0.0335 |
| 37 attributes | 98.9 | 94.1 | 1.3 | 7.2 | 0.0199 | 0.0887 |
| KDD winner | 92.7 | | 0.55 | | 0.2331 | |

It is evident from above table that:

1. Overall accuracy of CS-MC4 classifiers is better than their CS-CRT counterparts in all cases. And in all the cases, the overall accuracy of both CS-MC4 and CS-CRT is better than the KDD winner result.
2. False positives rate of CS-MC4 classifiers are better than their CS-CRT counterparts in all cases. And in all the cases, the false positive rate of the KDD winner result is better than both CS-MC4 and CS-CRT classifiers.
3. Average misclassification cost of CS-MC4 classifiers is better than their CS-CRT counterparts in all cases. And in all the cases, the Average misclassification cost of both CS-MC4 and CS-CRT is better than the KDD winner result.
4. Attribute reduction using information again value achieved better result for CS-CRT classifier from 41 attributes to 30 attributes, from 94.2% to 98.4% accuracy, from 7.1% to 1.7% false positive rate, and average misclassification cost from 0.0895 to 0.0335; but for CS-MC4 classifier it only reduce the attribute from 41 attributes to 37 attributes.

## VIII. Conclusions

This paper focuses on using cost sensitive learning techniques to existing data mining algorithms to deal with cost of different types of attacks in intrusion detection while at the same time reducing the false positives rate. The cost issue is widely ignored in the intrusion detection research. As a result, most of the research projects tried to minimize the false positives rate which may not reflect real-world scenario of dealing with ever increasing different types of attacks with different costs; and an important consideration is the fact that raising false alarms carries a significantly lower cost than not detecting attacks.

For comparison results of CS-MC4, CS-CRT and KDD 99 winner result, it was found that CS-MC4 is superior to CS-CRT in terms of accuracy, false positives rate and average misclassification costs. CS-CRT is superior to KDD winner result in accuracy and average misclassification costs but in false positives rate KDD winner result is better than both CS-MC4 and CS-CRT classifiers.

This paper proposed learning approach for network intrusion detection that performs feature selection method using information gain by selecting important subset of attributes. The performance of the proposed approach on the NSL-KDD intrusion detection dataset achieved a better accuracy from 94.2% to 98.4% for the CS-CRT classifier. It also reduced the false positives for the CS-CRT algorithm from 7.1% to 1.7%, and the average misclassification costs from 0.0895 to 0.0335; but for the CS-CM4 algorithm, it only reduced the attribute from 41 to 37 attributes.Even though feature selection method could not increase accuracy or reduce false positive rate and average misclassification cost for CS-MC4, it could increase the speed of the classifier and reduce the computation space required. The experimental results have manifested that feature (attribute) selection improves the performance of IDS.

## References Références Referencias

1. Abraham, R. Jain and J. Thomas, D-SCIDS: Distributed soft computing intrusion detection system, Journal of Network and Computer Applications, vol. 30, p p. 81-98, 2007.
2. Ajith, G. Crina, C. Yuehui. Cyber Security and the Evolution of Intrusion Detection Systems, Information Management and Computer Security 9(4): 175-182, 2001.
3. AlirezaOsareh and BitaShadgar (2008). Intrusion Detection in Computer Networks based on Machine Learning Algorithms. IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.11
4. Chai, X., Deng, L., Yang, Q., and Ling,C.X. , Test-Cost Sensitive Naïve Bayesian Classification, In Proceedings of the Fourth IEEE International Conference on Data Mining. Brighton, UK: IEEE Computer Society Press, 2004.
5. D.E. Denning, An Intrusion Detection Model, IEEE Trans-actions on Software Engineering, SE-13:222-232, 1987.
6. Domingos, P, MetaCost: A general method for making classifiers cost-sensitive, In Proceedings of the Fifth International Conference on Knowledge

Discovery and Data Mining, 155-164, ACM Press, 1999.

7. Drummond, C., and Holte, R., Exploiting the cost (in) sensitivity of decision tree splitting the criteria. In Proceedings of the 17 International Conference on Machine Learning, 239- 246, 2000.
8. E. Bloedorn, Data Mining for Network Intrusion Detection: How to Get Started, MITRE Technical Report, August 2001.
9. ElkanC Results of the KDD'99 Classifier Learning Contest, 1999. Available from <http:// www- cse. ucsd.edu/users/elkan/clresults.html>
10. Elkan, C. The Foundations of Cost-Sensitive Learning, In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence, 973-978. Seattle, Washington, 2001.
11. http://eric.univlyon2.fr/~ricco/tnagra.html.
12. [12]J. Luo, Integrating Fuzzy Logic With Data Mining Methods for Intrusion Detection, Master's thesis, Department of Computer Science, Missis - sippi State University, 1999.
13. J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory, ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.
14. J. P. Anderson, Computer Security Threat Monitoring and Surveillance, Technicalreport, Jam - esP Anderson Co., Fort Washington, Pennsylvania, April 1980.
15. Joseph, S and Rod, A, Intrusion detection: methods and systems. Information Management and Computer Security 11(5):222-229, 2003.
16. KDD CUP 1999 Data. The UCI KDD Archive Information and Computer Science, University of California,Irvine.http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
17. L. Ertoz, E. Eilerson and A. Lazareviv, The MINDS – Minnesota intrusion detection system, next generation data mining, MIT Press, 2004.
18. L. T. Heberlein, K. N. Levitt, and B. Mukherjee. A Method To Detect Intrusive Activity in a Networked Environment, In Proceedings of the 14th National Computer Security Conference, pages 362{371, october 1991.
19. Lee W, Stolfo SJ. A Framework for constructing features and models for intrusion detection systems, ACM TransInfSystSecurity, 2000.
20. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.
21. Nsl-kdd data set for network-based intrusion detection Systems. Available on: http://nsl.cs.unb. ca/ NSL- KDD/, March, 2010.
22. Paul Dokas, LeventErtoz, Vipin Kumar, AleksandarLazarevic, Jaideep Srivastava, Pang-Ning Tan, Data Mining for Network Intrusion Detection ,University of Minnesota, 2001.
23. Provost, F., Machine learning from imbalanced data sets 101, In Proceedings of the AAAI'2000 Workshop on Imbalanced Data, 2000.
24. R. Heady, G. Luger, A. Maccabe, and M. Servilla. The Architecture of a Network Level Intrusion Detection System, Technical report, Department of Computer Science, University of New Mexico, August 1990.
25. R. Heady, G. Luger, A. Maccabe, and M. Servilla. The Architecture of a Network Level Intrusion Detection System, Technical report, Department of Computer Science, University of New Mexico, August 1990.
26. S. Wafa, Al-Sharafat , ReyadhSh.Naoum, Adaptive Framework for Network Intrusion Detection by Using Genetic-Based Machine Learning Algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.
27. S.Selvakani, R.S. Rajesh, Genetic Algorithm for framing rules for intrusion Detection, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.11, November 2007.
28. Sheng, V.S. and Ling, C.X. , Thresholding for Making Classifiers Cost-sensitive, In Proceedings of the 21National Conference on Artificial Intelligence, 476-481, Boston, Massachusetts, 2006.
29. SimsonGarfinkel and Gene Spafford. Practical Security, O'Reilly and Associates, Sebastopol, California, 1991.
30. Sterrybrugger, Data Mining Methods for Network Intrusion Detection, University of California, June 2004.
31. Turney, P.D. Types of cost in inductive concept learning, In Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning, Stanford University, California, 2000.
32. Turney, P.D., Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm, Journal of Artificial Intelligence Research 2:369- 409, 1995.
33. Zadrozny, B. and Elkan, C., Learning and Making Decisions When Costs and Probabilities are Both Unknown, In Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining, 204-213, 2001.
34. Zadrozny, B., Langford, J., and Abe, N., Cost-sensitive learning by Cost-Proportionate instance Weighting, In Proceedings of the 3th International Conference on Data Mining, 2003.