Enhanced Network Intrusion Detection System for Imbalance Dataset based on Feature Selection and Classification Algorithms

Yogadhar Pandey
p_yogadhar@yahoo.co.in

Shailendra Singh
SeniorMember, IEEE
NITTTR, Bhopal, India.

Abstract—: **With rapid development and extensive growth of information systems, security problems increasingly bring into prominence.IDS are trained for classes: normal and attacks and further for types of attacks. A data set is called imbalanced when its elements are not evenly divided among the classes and this problem is a difficult challenge faced by machine learning and data mining researchers. Sampling techniques are one of the solutions to this problem. This paper is an attempt to provide an efficient method for imbalanced dataset classification with effect of sampling and feature selection. Proposed class balancing approach improve the performance of the classifier, shows better accuracy and reduced false alarm rate when compared with existing approach. Our proposed approach has been tested using KDD CUP'99 dataset.**

Keywords—IDS;Sampling;Machine Learning ;Classification;Class Balancin;KDD'99Dataset

I. INTRODUCTION

The network security is becoming an essential need of modern society to protect the confidential information flowing over the networks. Detection of Intrusion over the network is most important task to prevent their unlawful use by the attackers [1]. Although static defense mechanisms such as firewalls and software updates can provide a reasonable level of security, more dynamic mechanisms such as intrusion detection systems (IDSs) should also be utilized.

The IDSs can also be classified into two categories depending on where they look for intrusions. A host-based IDS monitors activities associated with a particular host, and a network based IDS listens to network traffic. There are two general categories of intrusion detection systems: misuse detection and anomaly detection.

The first one is misuse detection, also called signature-based detection[2]. In this type of IDSs, the Search for evidence of attacks is based on knowledge accumulated from known attacks. This knowledge is represented by attacks' signatures which are patterns or sets of rules that can uniquely identify an attack. Being de- signed based on the knowledge of the past intrusions or known vulnerabilities, misuse-based IDSs are also called knowledge-base detection. The advantages of knowledge-based approaches are that they have a very good accuracy and very low false alarm rate.

The second type of IDSs is anomaly detection or behavior-based detection. In this approach models of legitimate activities are built based on the normal

Data, and then the deviation from the normal model will be considered as an attack or anomaly[3]. The main advantage of this approach over misuse detection is that it can detect attempts to exploit new and unforeseen vulnerabilities.

It also can help detect "abuse of privileges" types of attacks that do not actually involve exploiting any security vulnerability. However, this approach has its own shortcomings. The main reported problem is high false alarm rate which is caused by two kinds of problems. The first one is the lack of a training data set that covers all the legitimate areas, and the other one is that abnormal behavior is not always an indicator of intrusions. It can happen as a result of factors such as policy changes or offering of new services by a site.

In recent years, interest was given into machine learning techniques to overcome the constraint of traditional intrusion techniques by increasing accuracy and detection rates. Various well-known machine learning techniques can be used in our detection approach. One of the biggest challenges in network-

based intrusion detection is the extensive amount of data collected from the network. Therefore, before feeding the data to a machine learning algorithm, raw network traffic should be summarized into higher-level events such as connection records. Each higher-level event is described with a set of features. Selecting good features is a crucial activity and requires extensive domain knowledge.

II. Literature Survey:

In network based intrusion detection system many feature selection methods /approaches have been applied as per the need of the classification algorithms. Appropriate feature selection method is required to select proper features to reduce the dimensionality of the KDD'99 dataset. Class imbalance problem for minority classes like U2R and R2L affects the performance of the classifiers. Different author's contributions have summarized in their literature.

Zhang et al.[4] proposed statistical neural network classifier for anomaly which can be acquainted with UDP flood attacks. Comparing different neural network classifiers, the back propagation neural network (BPN) has shown to be more proficient in developing IDS. Author uses the back propagation method by Sample Query and Attribute Query for the Intrusion Detection, whereby analyzing and identifying the most important components of training data. It could reduce processing time, storage requirement.

Liu et al. [5] built an anomaly model which performed feature selection by PCA and classification by neural networks. Only 22 features were extracted from the 38 feature set. Principal components selected were based on the highest Eigen values. This technique minimized the total number of features and increased the detection rate. However, the approach of selecting the principal components is not globally optimal as certain subsets of principal components are only investigated. The increased generalization performance of PCA is obtained with the trade -off of large amount of computation time.

Mustapha Belouch et al.[6] evaluated the performance of four well-known classification algorithms; SVM, Naïve Bayes, Decision Tree and Random Forest using Apache Spark, a big data processing tool for intrusion detection in network traffic. The overall performance comparison is evaluated in terms of detection accuracy, building time and prediction time. Experimental results on UNSW-NB15, a recent public dataset for network intrusion detection, show an important advantage for Random Forest classifier among other well-known classifiers in terms of detection accuracy and prediction time, using the complete dataset with all 42 attributes.

Marwa F. Al-Roby[7] proposed new approach based on SOMTE (Synthetic Minority Over-sampling Technique) and clustering which is able to deal with imbalanced data problem involving multiple classes. We implemented our approach and experimental results show our approach is effective to deal with the multi class imbalanced data sets, and can improve the classification perfo rmance of minority class and its performance on the whole data set. In the best case, our F-measure improved from 66.91 to 95.1%.

Shadi Aljawarneh et al. [8] developed a new hybrid model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training. The experimental results revealed that the hybrid approach had a significant effect on the minimization of the computational and time complexity involved when determining the feature association impact scale. The accuracy of the proposed model was measured as 99.81% and 98.56% for the binary class and multiclass NSL-KDD data sets, respectively. However, there are issues with obtaining high false and low false negative rates. A hybrid approach with two main parts is proposed to address these issues. First, data needs to be filtered using the Vote algorithm with Information Gain that combines the probability distributions of these base learners in order to select the important features that positively affect the accuracy of the proposed model. Next, the hybrid algorithm consists

of following:classifiers:J48,MetaPagging,RandomTree,REPTree, AdaBoostM1, DecisionStump and NaiveBayes. Based on the results obtained using the proposed model, we observe improved accuracy, high false negative rate, and low false positive rule.

Ligang Zhou al.[9] investigates the effect of six different sampling methods on the performance of five quantitative bankruptcy prediction models. Each sampling method and quantitative model is tested on two datasets. The results shows that when there are hundreds of bankrupt observations in the dataset, under sampling method is better than oversampling method because there is no significant difference on performance but oversampling method consumes more computational time. When there are only dozens of bankrupt cases in the dataset, oversampling method SMOTE is a better choice and if the training sample size is too large to cause the failure of model construction, the combination of SMOTE and undersampling maybe an alternative. In the test on both datasets, it is interesting to observe that the difference of AUC performance of all models, trained by sample set obtained by random undersampling method, tested on random paired sample and real highly imbalanced sample is very slight or not significant, therefore, in the bankruptcy prediction model selection, the models can be evaluated and measured on their performance on random balanced sample set instead of the real highly imbalanced test sample. This work mainly focuses on the sampling method for bankruptcy prediction model construction with highly imbalanced dataset. All tested quantitative models just adopt the fundamental form and have no parameters and model optimization. The performance of models varies with the sampling method, but SVM can achieve good performance in most scenarios. There is a lot of bankruptcy prediction models, in practice, model selection process should be conducted since no model can always perform well.

W. Khreich et al.[10] Proposed feature extraction approach starts by segmenting the system call traces into multiple n-grams of variable length and mapping them to fixed-size sparse feature vectors, which are then used to train OC-SVM detectors. The results achieved on a real-world system call dataset show that our feature vectors with up to 6-grams outperform the term vector models (using the most common weighting schemes) proposed in related work. More importantly, anomaly detection system using OC-SVM with a Gaussian kernel, trained on our feature vectors, achieves a higher-level of detection accuracy (with a lower false alarm rate) than that achieved by Markovian and n-gram based models as well as by the state-of-the-art anomaly detection techniques. The proposed feature extraction approach from traces of events provides new and general data representations that are suitable for training standard one-class machine learning algorithms, while preserving the temporal dependencies among these events.

Ishfaq Manzoor et al. [11] proposed a feature reduction method using ANN-based classification. Features have been reduced using ranker attribute selection methods like: InfoGain, Correlation-based Feature selection. Proposed method for U2R and R2L achieve detection rate of 86.6% and 91.9% respectively. Precision for U2R and R2L classes are 42.88% and 87.5% respectively. Features still needs to be reduced to get an optimal featured set and precision should be improved which in turn reduce the model building time. Thus an efficient approach is needed for feature reduction.

In literature review different authors have applied different feature selection methods to reduce dimensionality of the KDD99 dataset. When features are reduced, time taken for model generation also gets reduced. But for minority classes like U2R and R2L, the precision is not much satisfactory. Therefore efficient sampling approach is needed to sample the highly imbalanced KDD99 dataset before feature selection method. Thus the detection rate and accuracy for minority classes U2R and R2L will improve and the false alarm rate (FAR) of the system will be reduced, which in turn improve the overall performance of the intrusion detection system.

III. SAMPLING:

Sampling is most widely used method to combat the problem of class imbalance. The objectives of sampling techniques are to create a dataset that has a comparatively balanced class distribution, so that conventional classifiers are capable to distinguish the decision boundary between the majority and the minority classes. Sampling has often been employed as an efficient means to decrease the size of selecting representatives which signifies the entire data set by looking at a portion[12].

The basic sampling methods include under-sampling and over-sampling. Under-sampling eliminates majority class examples while over-sampling increases the number of minority-class examples. Both of these sampling techniques decrease the overall level of class imbalance, thereby making the rare class less rare.

Under-sampling discards potentially useful majority-class examples and thus can degrade classifier performance. Therefore, they are normally used on very large data sets in which there are enough redundant data to be removed.

Over-sampling, however, can increase the time necessary to build a classifier because it introduces additional training cases. Even worse, because over-sampling often involves making exact copies of examples, it can lead to over fitting [13]. As an extreme case, final classification rules may be introduced to cover a single and replicated example. More importantly, Over-sampling introduces no new data, so it does not address the fundamental "lack of data" issue.

Some clever sampling methods are proposed which may use instance selection algorithms when removing or adding examples or combine under-sampling and over-sampling techniques. Some studies have recognized the fact that it is still unclear which sampling method performs best, what sampling rate should be used, and that the proper choice is probably domain specific. Common techniques are as follows:

Over-sampling

Random Over-sampling: Balance class distribution by replicating minority class examples randomly, but it increases the likelihood of overfitting, since it makes exact copies. Make decision regions of the learner more specific and closer to minority class.

Random oversampling with replication (ROWR)

Algorithm 1. ROWR (Smi,Sma)

Input: The original sample set of minority class Smi and

sample set of majority class Sma.

Output: sample set with balanced class S of size 2[Sma].

1. S = Sma;

2. for i = 1 to [Smi]

randomly selected an element ai from sample set Sma

S = S U{ai}.

endfor

Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE, proposed by Chawla et al. [13], is an improved oversampling approach in which the minority class is oversampled by creating ‘‘synthetic’’ examples rather than by oversampling with replacement. The main idea of SMOTE is to oversample the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbor.

Under-sampling

Random Under-sampling: Balance class distribution by removing majority class examples randomly. Main disadvantage is that it discards data that may contain useful information.

Random Undersampling (RU)

Random under-sampling method is to balance class distribution through the random elimination of majority class examples [14]. RU employed in this study is to randomly select part of the majority class to achieve the balance which obtains the equivalent result.

Algorithm: RU (Smi,Sma)

Input: The original sample set of minority class Smi and
sample set of majority class Sma.
Output: sample set with balanced class S of size 2[Smi]
1. S = Smi;
2. for i = 1 to [Smi]
Randomly selected an element ai from sample set Sma
S = S U{ai}.
Endfor

Algorithm: SMOTE(Oversampling)

Algorithm: SMOTE (Ck,N,k)
Input: Number of minority class samples Ck; Amount of minority class being oversampled N%; Number of nearest neighbor's k
Output: Union of (N/100) (Ck) synthetic minority class samples and the majority set (Ci)
1. if N < 100 then
Randomize the (Ck) minority class samples
T = (N/100) Ck
N = 100
End if
2. N= (int)(N/100)
3. k = number of neighbors
4. numattrs = Number of attributes
5. Sample[][]: array for original minority class samples
6. newindex = 0//keeps a count of number of synthetic samples generated
7. Synthetic[][]: array for synthetic samples
8. for i = 1 to Ck
Compute k nearest neighbors for i, and save the indices in the nnarray
Populate (N, i,nnarray,newindex, Synthetic)
Endfor
Populate (N, i,nnarray,newindex, Synthetic)
9. while N =/ 0
nn = random number between 1 and k
for attr = 1 to numattrs
dif = Sample[nnarray[nn]][attr] Sample[nnarray[i]][attr]
gap = random number between 0 and 1
Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
endfor
newindex++
N = N -1

IV. Feature Selection

Attribute selection also known as feature selection can be used to remove model redundancy thus reducing the time required for model generation. Feature selection is a "Technique of selecting a subset of relevant features by removing most irrelevant and redundant features from the data for building an effective and efficient learning model" [15].

Feature selection processes involve four basic steps in a typical feature selection method shown in Figure 1. First is subset generation procedure to generate the next candidate subset; second one is an evaluation function to evaluate the subset and third one is a stopping criterion to decide when to stop; and a validation procedure to check whether the subset is valid.

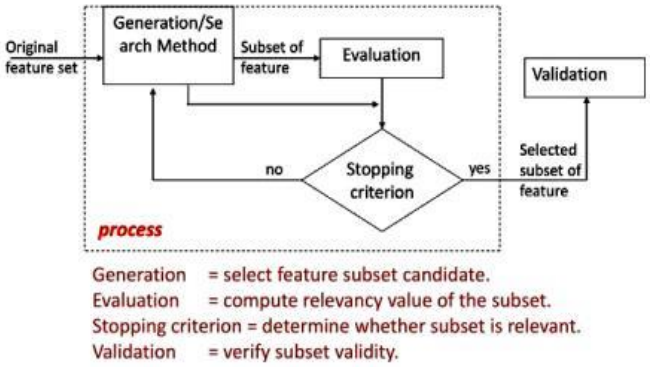


Figure 1: Feature Selection Process

Some Attribute evaluator is basically used for ranking all the features according to some metric. Feature selection flow on the KDD99 dataset has been carried out as follows[16].

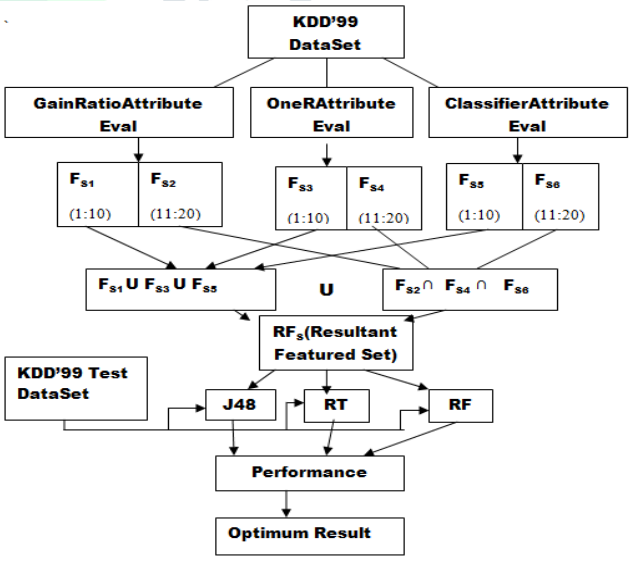


Figure2: Flow Chart of Feature Selection Approach

V. PROPOSED HYBRID SAMPLING TECHNIQUE

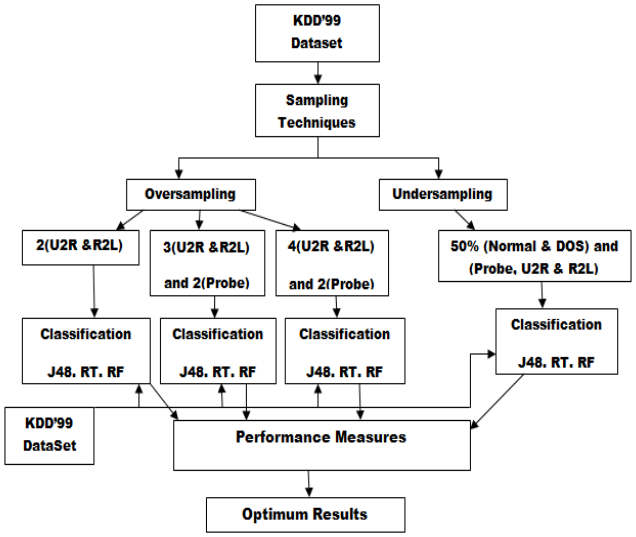


Figure3: Flow Chart of proposed classification Approach

Working Procedure of Proposed Sampling Algorithm

Proposed sampling algorithm summarized in following steps:

Step1→KDD99 training dataset fed as an input to sampling technique.

Step2→Oversample the minority class such that: 2(U2R& R2L).

Step3→Go to step1 and then oversample the featured set

Such that: {3(U2R & R2L) and 2(Probe)}

Step4→Go to step1 and then oversample such that: {4(U2R& R2L and 2(Probe)}.

Step5→Apply RFSA(feature selection algo.)

Step6→ Appply classification algorithms J48, RT & RF in step5.

Step7→Take KDD99 Test dataset as an input to the model build in step5.

Step8→Again Go to Step1

Step9→Undersample the majority classes DoS &Normal such as :{ 50 %(Normal & DoS) and same tuples of (Probe,U2R & R2L)}.

Step10: Apply RFSA(feature selection algo.)

Step11→Apply Classification Algorithms J48, RT & RF in step10.

Step12→ Take KDD99 Test dataset as input to the model build in step11

Step12→Compare performance of the classifiers

Step13→ After comparisons select the best performing one and get the optimum result.

The majority class is under-sampled by randomly removing samples from the majority class population until the minority class becomes some specified percentage of the majority class. This forces the learner to experience varying degrees of under-sampling and at higher degrees of under-sampling the minority class has a larger presence in the training set. In describing our experiments, our terminology will be such that if we under-sample the majority class at 200%, it would mean that the modified dataset will contain twice as many elements from the minority class as from the majority class. By applying a combination of under-sampling and over-sampling, the initial bias of the learner towards the negative (majority) class is reversed in the favor of the positive (minority) class. Classifiers are learned on the dataset perturbed by "SMOTING" the minority class and under-sampling the majority class[13].

As per the data set the training set used will contain training dataset having 494021 instances and test dataset includes 311029. The size of training and testing set may be varied. KDD Dataset[17] Obtained after applying combination of under and over sampling techniques:

After applying the technique, the sampled dataset will be obtained. The obtained data is balanced in terms of class distribution. Apply RFSA feature selection proposed in our previous research and then go for classification using

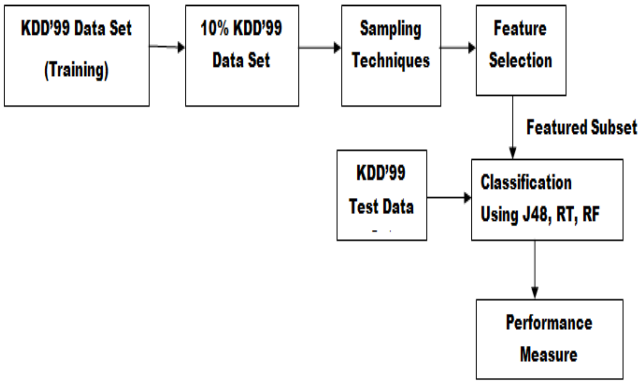


Figure4: Flow Chart for classification

J48, Random Tree (RT) and Random Forest (RF) classification algorithms [18] . As each feature of the data instances has different scale, the time taken to build a model is reduced.

V. PERFORMANCE MEASURE

Detection rate: Detection rate refers to the percentage of detected Attack among all attack data, and is defined as follows:

$$\text{Detection rate} = \frac{TP}{TP+TN} * 100$$

False alarm Rate: False alarm rate refers to the percentage of normal data which is wrongly recognized as attack. , and is defined as follows:

$$\text{False Alarm rate} = \frac{FP}{FP + TN} * 100$$

VI. RESULT AND DISCUSSION

Classification algorithms used for the experiments are J48, random tree, random forest.494021 connections are used for training set and 311029 connections are used for testing set This work uses weka 3.8[19] for performing the experiment. Performance of the different classifiers have been evaluated and summarized in tables.

Oversampling model:

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48{2(U2R&R2L)}	TPR(%)	99.4	87.43	94.2	94.15	91.2
	FPR	0.032	0.035	0.003	0.0012	0.0016
	Precision(%)	91.3	74.6	99.28	91.46	98.39
	Accuracy(%)	96.4	99.81	95.12	99.82	98.94
RF{2(U2R&R2L)}	TPR(%)	99.52	88.64	95.14	95.33	90.94
	FPR	0.03	0.032	0.0027	0.001	0.0012
	Precision(%)	92.89	78.23	99.43	92.21	97.92
	Accuracy(%)	97.3	99.86	95.94	99.87	98.1
RT{2(U2R&R2L)}	TPR(%)	98.74	84.92	91.23	89.21	93.64
	FPR	0.042	0.054	0.0045	0.029	0.0024
	Precision(%)	89.42	72.12	92.34	89.32	97.96
	Accuracy(%)	95.84	96.84	93.63	91.86	98.02

Table3: Results of oversampling

Results for oversampling model1 have been tabulated in table3.it is seen that RF outperforms for the Normal,R2L,U2R &DoS classes but accuracy and precision for probe class is slightly reduces (negligible). Compared the performance of the classifiers and found overall performance of RF is better among J48, RT and RF. **Undersampling model:**

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48{50% (Normal&DoS)and (probe,U2R & R2L)}	TPR(%)	99.86	87.52	97.14	94.08	90.9
	FPR	0.012	0.033	0.0012	0.0011	0.0018
	Precision(%	93.4	75.5	99.62	91.39	98.23
	Accuracy(%	98.24	99.82	96.2	99.79	98.78
RF{50% (Normal&DoS)and (probe,U2R & R2L)}	TPR(%)	99.7	88.58	96.27	94.98	90.24
	FPR	0.03	0.032	0.0022	0.001	0.0011
	Precision(%	93.1	78.49	99.58	92.03	97.85
	Accuracy(%	97.89	99.87	96.1	99.82	97.96
RT{50% (Normal&DoS)and (probe,U2R & R2L)}	TPR(%)	98.92	83.59	93.48	88.21	93.28
	FPR	0.039	0.049	0.003	0.024	0.0021
	Precision(%	91.37	72.02	94.59	88.97	97.79
	Accuracy(%	96.32	96.79	95.33	91.73	98.21

Table4: Results of Undersampling

Table4 summarize the results for undersampling.Results shows that for Normal & DoS performance of J48 outperforms than RT and RF. But there is not much significant changes observed in the performance of U2R, R2L and Probe classes. J48 shows better accuracy, less FAR and high precision. Hence J48 shows better results among RT and RF for Undersampling.

Comparison RF(oversampling)VsJ48(Undersampling)

As results shown on the table3and table4 ,oversampling with RF works better among J48,RF and RT so we have selected RF(oversampling)for comparison with J48(undersampling) which has also shown better performance among J48,RT and RF in undersampling. Comparison results have been summarized in table5.

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
RF(Oversampling)	TPR(%)	99.52	88.64	95.14	95.33	90.94
	FPR	0.03	0.032	0.0027	0.001	0.0012
	Precision(%	92.89	78.23	99.43	92.21	97.92
	Accuracy(%	97.3	99.86	95.94	99.87	98.1
J48(Undersampling)	TPR(%)	99.86	87.52	97.14	94.08	90.9
	FPR	0.012	0.033	0.0012	0.0011	0.0018
	Precision(%	93.4	75.5	99.62	91.39	98.23
	Accuracy(%	98.24	99.82	96.2	99.79	98.78

Table5: Comparison RF(oversampling)VsJ48(Undersampling)

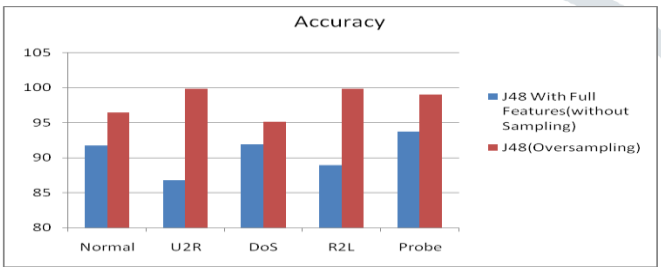
It is observed that oversampling performes (with RF) better than undersampling(J48} for majority as well as Minority classes.TPR of Normal class and R2L classes shows slight increment, but accuracy and precision of the oversampling (RF) shows better results and hence improve overall performance of the classifiers.

Comparison among J48 (without feature selection and Sampling), J48 (with Proposed Feature selection) and J48 (With Proposed Feature selection and proposed Sampling Approach):

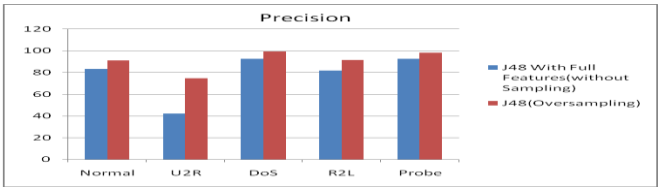
We have compared the J48 (without feature selection and without sampling), J48 (with proposed RFSA feature selection without sampling) and J48 (with RFSA feature selection and with proposed Sampling approach).results summarized in the table6:

Classifier	Metric	Normal	U2R	DoS	R2L	Probe
J48 With Full Features(without Sampling)	TPR(%)	96.4	76.58	93.52	78.2	83.7
	FPR	0.0679	0.038	0.0042	0.0036	0.0067
	Precision(%	83.5	42.3	92.67	81.7	92.64
	Accuracy(%	91.68	86.8	91.89	88.9	93.7
J48 with RFSA(without sampling)	TPR(%)	99.2	83.1	94.6	92.13	90.17
	FPR	0.048	0.062	0.0028	0.0018	0.002
	Precision(%	89.2	46.7	99.78	90.2	98.18
	Accuracy(%	95.6	99.72	96.82	99.67	98.88
J48(Oversampling)	TPR(%)	99.4	87.43	94.2	94.15	91.2
	FPR	0.032	0.035	0.003	0.0012	0.0016
	Precision(%	91.3	74.6	99.28	91.46	98.39
	Accuracy(%	96.4	99.81	95.12	99.82	98.94

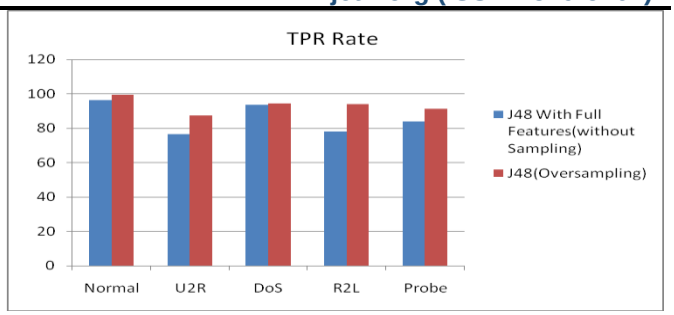
Table6: comparisons results of J48 with sampling, without sampling and with sampling and with feature selection



Accuracy:J48 with full featured set(without sampling) Vs J48 Oversampling



Precision of J48 with full featured set(without sampling) Vs J48 Oversampling



TPR Rate of J48 with full featured set(without sampling) Vs J48 Oversampling

Following conclusions have been observed on the analysis of the results summarized in table6 as follows:
TPR of the minority classes like U2R and R2L for J48 without sampling and j48 with feature selection and Oversampling) improved from 76.58% to 87.43 %(U2R) and from 78.2% to 94.15 %(R2L).
Similarly precision, accuracy and FAR also shows drastic improvement in their performance evaluation.

VII. Conclusion & Future Scope

The experimental results endorse that the proposed sampling technique with RFSA outperform for minority classes as well as majority classes. It is also observed that the performance metric (TPR) is improved drastically for U2R (76.58% to 87.43%) and R2L (78.2% to 94.15 %).

Proposed sampling algorithm is efficient and extensible in terms of various performance parameters. The performance of sampled and balanced dataset is better than un-sampled imbalanced dataset. The results also show that the sampling and feature selection and improve the classifier performance and hence improve the performance of intrusion detection system.

The future work includes more refinement in the feature selection algorithms which are leading for reduction in the number of features. The future work also includes for the development of efiicient sampling techniques using Extreme machine learning to improve the performance of the minority classes of the imbalanced dataset.

References

[1] Pillai, M. M., Jan HP Eloff, and H. S. Venter. "An approach to implement a network intrusion detection system using genetic algorithms." In Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries, pp. 221-221. South African Institute for Computer Scientists and Information Technologists, 2004.

[2] Kruegel, Christopher, and Thomas Toth. "Using decision trees to improve signature-based intrusion

detection." In International Workshop on Recent Advances in Intrusion Detection, pp. 173-191. Springer, Berlin, Heidelberg, 2003.

[3] Binkley, James R., and Suresh Singh. "An Algorithm for Anomaly-based Botnet Detection." SRUTI 6 (2006): 7-7.

[4] Zhang, Zheng, Jun Li, C. N. Manikopoulos, Jay Jorgenson, and Jose Ucles. "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification." In Proc. IEEE Workshop on Information Assurance and Security, pp. 85-90. 2001.

[5] Liu, Guisong, Zhang Yi, and Shangming Yang. "A hierarchical intrusion detection model based on the PCA neural networks." Neurocomputing 70, no. 7-9 (2007): 1561-1568.

[6] Mustapha Bekkouche, Salah El Hadaj , Mohamed Hammad "Performance evaluation of intrusion detection based on machine learning using Apache Spark Procedia Computer Science (ScienceDirect)127 (2018) 1–6

[7] Al-Roby, Marwa F., and Alaa M. El-Halees. "Classifying Muti-Class Imbalance Data" Egyptian Computer Science Journal 37, no. 5 (2013): 74-81.

[8] Aljawarneh, Shadi, Monther Aldwairi, and Muneer Bani Yassein. "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model" Journal of Computational Science 25 (2018): 152-160

[9] Zhou, Ligang. "Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods." Knowledge-Based Systems 41 (2013): 16-25.

[10] Khreich, Wael, Babak Khosravifar, Abdelwahab Hamou-Lhadj, and Chamseddine Talhi. "An anomaly detection system based on variable N-gram features and one-class SVM" Information and Software Technology 91 (2017(online): 186-197.
Available;http://www.sciencedirect.com/science/article/pii

[11] Manzoor, Ishfaq, and Neeraj Kumar. "A feature reduced intrusion detection system using ANN classifier." Expert Systems with Applications 88 (2017): 249-257.

[12] Dennis J. Drown, Taghi M. Khoshgoftaar, and Ramaswamy Narayanan. Using evolutionary sampling to mine imbalanced data. In Sixth International Conference on Machine Learning and Applications, 2007. ICMLA 2007., pages 363-368, 2007.

[13] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer "SMOTE: Synthetic Minority Over-sampling Technique" Journal of Artificial Intelligence Research 16 (2002) 321–357.

[14] Yueai, Zhao, and Chen Junjie. "FHNN: A Resampling Method for Intrusion Detection." In Information Engineering (ICIE), 2010 WASE International Conference on, vol. 2, pp. 168-171. IEEE, 2010.

[15] Eesa, Adel Sabry, Zeynep Orman, and Adnan Mohsin Abdulazeez Brifcani. "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems." Expert Systems with Applications 42, no. 5 (2015): 2670-2679.

[16] Yogadhar Pandey,Shailendra Singh "An Efficient Selection Approach for Network – based Intrusion Detection System Using Machine Learning Algorithm" published in proceeding of IJTRM (ISSN 2348-9006). Vol.5, Issue3.pp:1-8,2018.

[17] Olusola, Adetunmbi A., Adeola S. Oladele, and Daramola O. Abosede. "Analysis of KDD'99 intrusion detection dataset for selection of relevance features." In Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp. 20-22. 2010.

[18] Suh-Lee, Candace, Ju-Yeon Jo, and Yoohwan Kim. "Text mining for security threat detection discovering hidden information in unstructured log messages." In Communications and Network Security (CNS), 2016 IEEE Conference on, pp. 252-260. IEEE, 2016.

[19] Koc, Levent, Thomas A. Mazzuchi, and Shahram Sarkani. "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier." Expert Systems with Applications 39, no. 18 (2012): 13492-13500.