ISSN 2277-3061

Enhanced Tree Based Real Time Intrusion Detection System in Big Data

1. S.J.SATHISH AARON JOSEPH
RESEARCH SCHOLAR / HEAD , DEPT OF COMPUTER APPLICATIONS
J.J.COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS),PUDUKKOTTAI,TAMIL NADU
satjoe7@gmail.com
2. R.BALASUBRAMANIAN
PROFESSOR , P.G AND RESEARCH DEPARTMENT OF COMPUTER SCIENCE
J.J.COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)
PUDUKKOTTAI , TAMIL NADU
drrb_1951@yahoo.com

ABSTRACT

Intrusion detection is one of the major necessities of the current networked environment, where every information is available in its corresponding digital form. This paper presents an enhanced tree based approach that can be used to perform intrusion detection faster and with better accuracy. The training data is subject to the random forest algorithm. This algorithm is a combination of tree predictors, and each tree depends upon the random vector generated. Spark based implementations of the Random Forest algorithm is used in a Hadoop cluster on datasets with varied imbalance to obtain the results. It has been observed that the classifier provided results in real time with an accuracy >90%, hence is more appropriate for online intrusion detection.

Indexing terms/Keywords

Classification; Data Imbalance; Decision Trees; Intrusion Detection; Random Forest

INTRODUCTION

Online transactions has become one of the day to day activities due to the advent of ecommerce. Most of the ecommerce organizations prefer to store sensitive information, so that it becomes easier for the user to perform transactions [5]. It becomes mandatory for the organizations to store this information in a secure manner. Any information available in digital form is prone to hacks. Due to the importance associated with the contained information, the process of detection and prevention should be synonymous to the attack. Existing intrusion detection and prevention systems tend to provide a buffer of a few transactions before identifying the intrusion, which tends to be costly. Further, network data tends to be huge, hence analyzing the entire data is both time and processor consuming [12]. Hence it becomes mandatory for the systems to either use Big Data techniques or reduce the available data to chunks that can be handled by the existing processing architecture. This approach presents an architecture to identify intrusions in real time, combined with the flexibility of using even very huge amounts of data.

RELATED WORKS

Intrusion Detection, being a mature technique has several contributions in literature. This section discusses some of the most recent techniques in the area of intrusion detection.

A profiling based intrusion detection system that uses network traffic to identify intrusions is presented in [1]. Network packets tends to occur in large numbers. Analyzing each packet for intrusion is not feasible. This method is based on a reduction strategy that eliminates probably legitimate packets and passes only a few packets for processing. Alpha and beta profiling are used to reduce the number of data for comparison. Feature based reduction is also performed to reduce the number of comparisons further. A statistical rule based intrusion detection mechanism is presented in [6]. This is a genetic algorithm based technique that is designed to evolve a set of simple interval based rules. Genetic algorithm is modified such that the rule set is maintained small. A similar method concentrating on DDoS attacks is presented in [7]. A similar technique using GA that speeds up the detection mechanism is presented in [16]. Algorithms to speedup pattern matching have also been analyzed and documented in [8], which provides efficient techniques to improve the process of pattern matching. These techniques were effectively utilized in the process of intrusion detection. Similar rule based techniques were in prevalence. An Apriori algorithm based intrusion detection is presented in [9].

A collaborative method for intrusion detection in mobile networks is proposed in [2]. This method mostly concentrated on stealth attacks which cannot be detected by any existing intrusion detection mechanisms. A multi-level intrusion detection mechanism is presented in [3]. This method uses a coarse grained and a fine grained mechanism to identify intrusions. Examining each packet for intrusion is very tedious and hence not feasible. The coarse grained detection mechanism is activated initially and checks for intrusions. A packet, if identified as probable intrusion, is taken to the fine grained control for extended evaluation. A priority based intrusion detection mechanism is presented in [10]. This method also presents post correlation techniques to analyze the results obtained. Neighbor based intrusion detection methods [11] that operate by ranking the neighbors have been widely used in clustered environments. Similar to statistical models, machine learning models are also on the raise, due to the ever changing nature of the intrusions. An analysis of such mechanisms is presented in [13]. A combined SVM and PSO based intrusion detection method that also uses dimensionality reduction is presented in [17]. Increase in the number of processor cores and reduction in cost of processors has lead to an increased use of parallelization techniques. A survey of intrusion detection techniques on GPUs is presented in [15]. A parallel intrusion detection method that uses GPGPUs to perform faster and energy efficient intrusion detection is presented in [4].

Though several fast intrusion detection techniques exists in literature, a major problem in this area is the absence of real time detection. The approach presented here tends to reduce this gap by providing real time detection of intrusions in Big Data.

ENHANCED TREE BASED REAL TIME INTRUSION DETECTION SYSTEM IN BIG DATA

Intrusion Detection in real time is a tedious task requiring faster and more accurate strategies in order to provide effective results. The proposed architecture uses an ensemble of decision trees to perform the process of classification. Each decision tree operates independently on a subset of the data to provide the decision rules, which are then integrated to form the final classifier model. This process works on the basic principle that results of several weak classifiers can be combined to form a strong classifier.

The input data is preprocessed to convert it to a format accepted by the classifier. The operations carried out in this stage is minimal, because random forest classifiers handles missing data and imbalanced data well. Hence there is no necessity to eliminate or handle missing data. The data is then segregated to training and test data, to aid in the cross validation testing that is to be carried out on the classifier. The training data is then passed to the random forest classifier for building the classifier model.

The random forest classifier is an ensemble model that uses several decision trees on subsets of the data to build decision rules. Subset creation is carried out in such a manner that each of the subsets contain at least 66% of the original data. This is to make sure that all the classes contained in the original data has representatives in the subset. This would ensure that all the classes are considered prior to the splitting section in the decision trees.

The next phase of this process is the actual creation of the decision trees. Data subsets are passed to the decision trees. Each decision tree identifies a subset of m predictor variables from the M total predictor variables (m<M). The best predictor variable is identified from this set and a binary split is performed on it. This marks the beginning of the tree creation. The process of predictor variable identification and splitting is carried out for all the available predictors and the decision tree is constructed. Pruning is not performed on the decision tree in order to retain all the data during the rule aggregation step.

There exist three different methods to select the value of m. The Random splitter selection method, where m=1, the Breiman's bagger method, where m=M and the Random Forest method where m<<M. Brieman suggested three possible ways of selecting the values of m, namely $^{1}/_{2}\sqrt{m}$, $\sqrt{m}$ or $2\sqrt{m}$.

Each of the rules are obtained by training the decision trees on a part of the data, hence the rules returned by each of the decision trees differ. The final classifier model is built by combining all these rules. It has been observed that though each of the decision trees provided weak rules, a combination of these rules exhibits a strong classifier with improved reliability and accuracy.

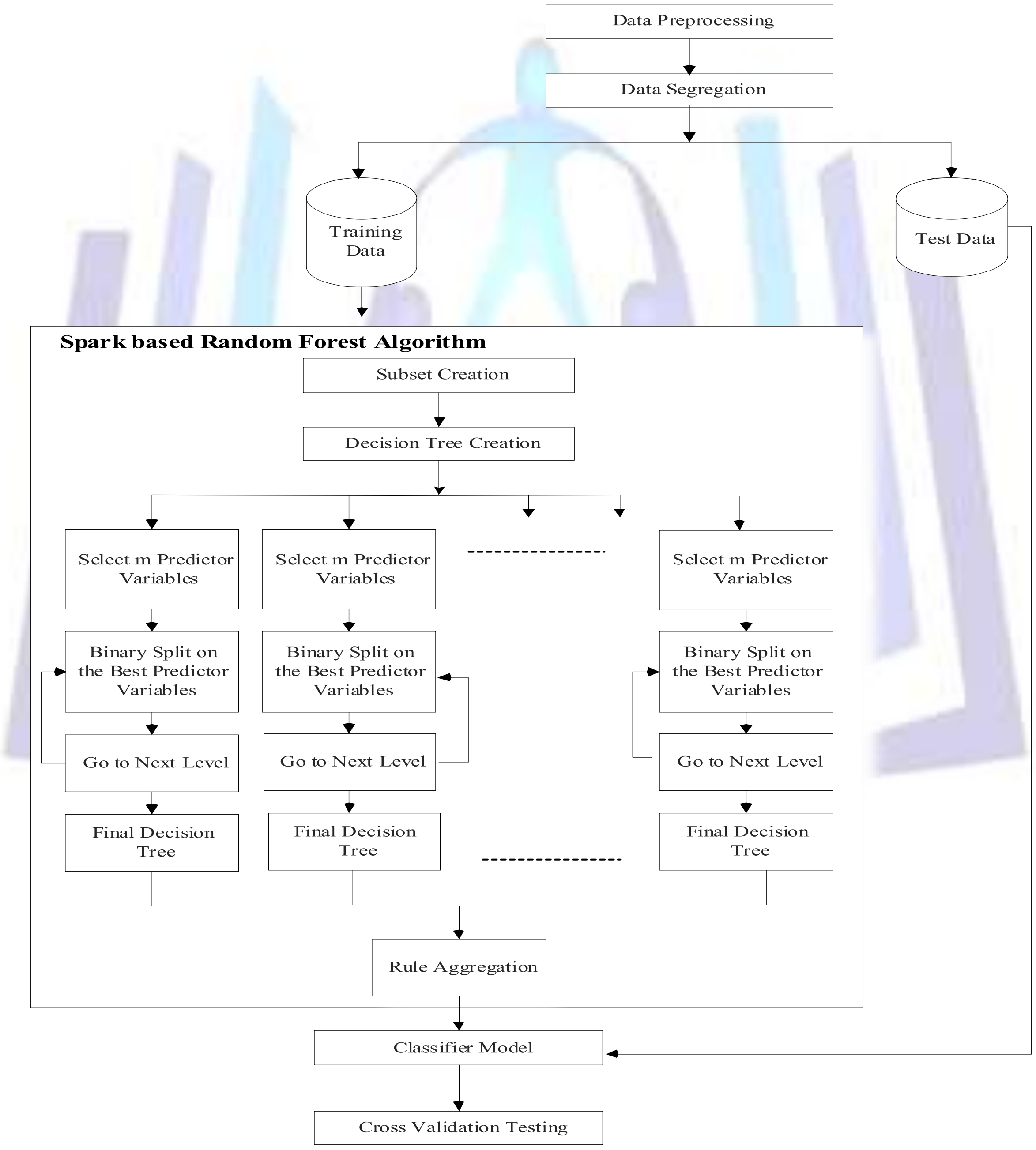


Fig 1: Spark Based Random Forest Intrusion Detection : Architecture

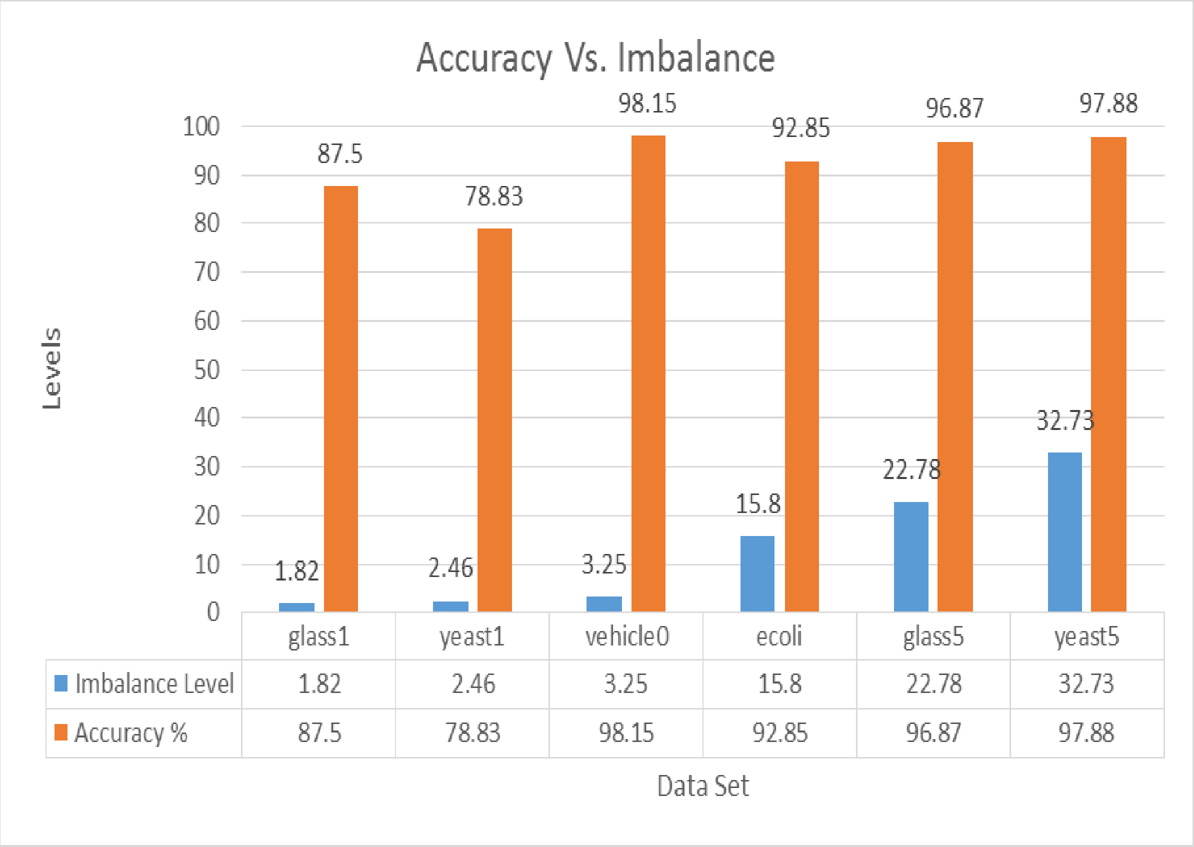
RESULTS AND DISCUSSION

Experiments were carried out using Hadoop 2 and Spark 1.4. Table 1 describes the datasets that have been used for processing and their properties. Binary Classification datasets of varied imbalance levels are used to observe their impact on accuracy. The datasets were obtained from KEEL repository [18].

Table 1: Dataset Description

Dataset	# of attributes	# of instances	# of Classes	Imbalance Level
Glass1	9	214	2	1.82
Yeast1	8	1484	2	2.46
Vehicle0	18	846	2	3.25
Glass5	9	214	2	22.78
Yeast5	8	1484	2	32.73
Ecoli4	7	336	2	15.8

Classification was performed on all these datasets to obtain the confusion matrix, accuracy and reliability scores. Analysis of the classifier was performed on the basis of imbalance levels and reliability along with the accuracy scores. Imbalance, by nature tends to increase the accuracy, but reduce the reliability levels. Hence it becomes mandatory to measure the reliability scores along with the accuracy levels.



	glass1	yeast1	vehicle0	ecoli	glass5	yeast5
🟦 Imbalance Level	1.82	2.46	3.25	15.8	22.78	32.73
🟧 Accuracy %	87.5	78.83	98.15	92.85	96.87	97.88

Fig 2: Accuracy Vs. Imbalance

A comparison between accuracy and imbalance is presented in Fig 2. The imbalance levels are varied gradually from 1.82 to 32.73. it was observed that though the imbalance levels increase, the accuracy levels do not exhibit any drastic changes. Hence it could be concluded that low to moderate imbalance do not affect a classifier's accuracy.

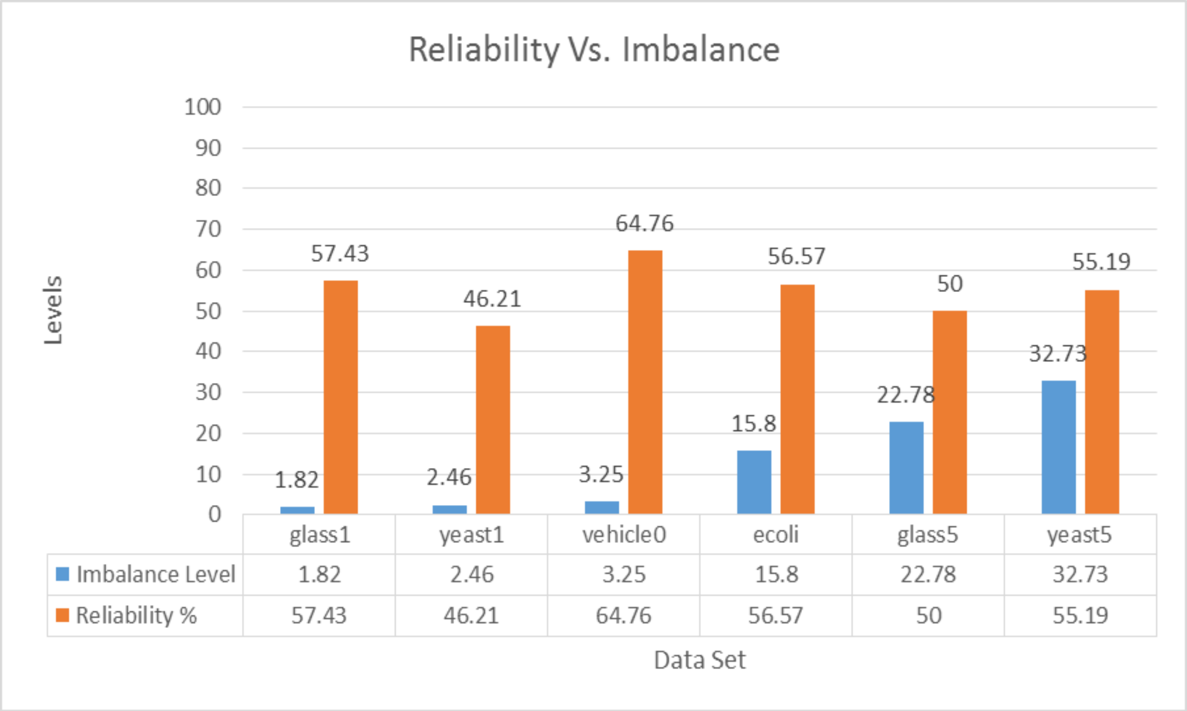


Fig 3: Reliability Vs. Imbalance

A ratio between the reliability scores and imbalance levels are exhibited in Fig 3. It was observed that the imbalance levels have low impact on the reliability scores.

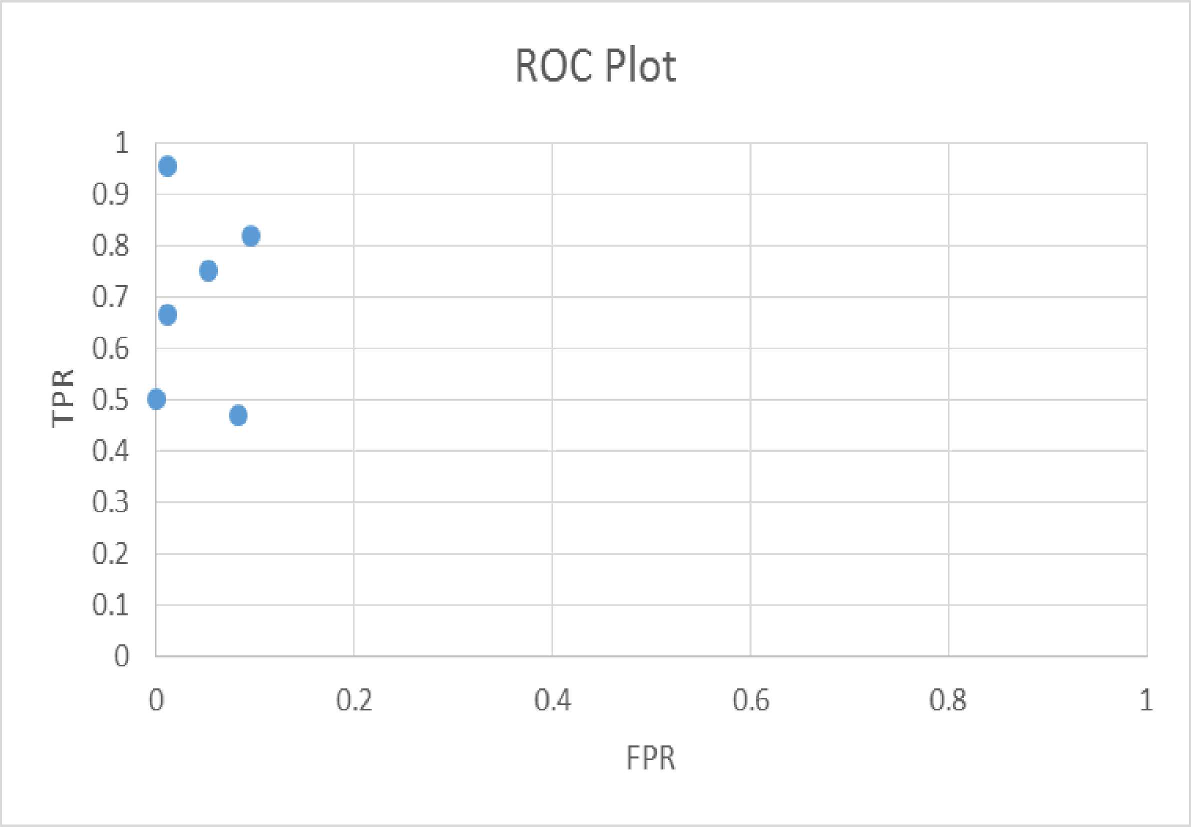


Fig 4: ROC Plot

ROC plot for the random forest classifier is shown in Fig 4. It was observed from the figure that all the points are contained in the top left corner of the ROC space exhibiting effective accuracy of the classifier.

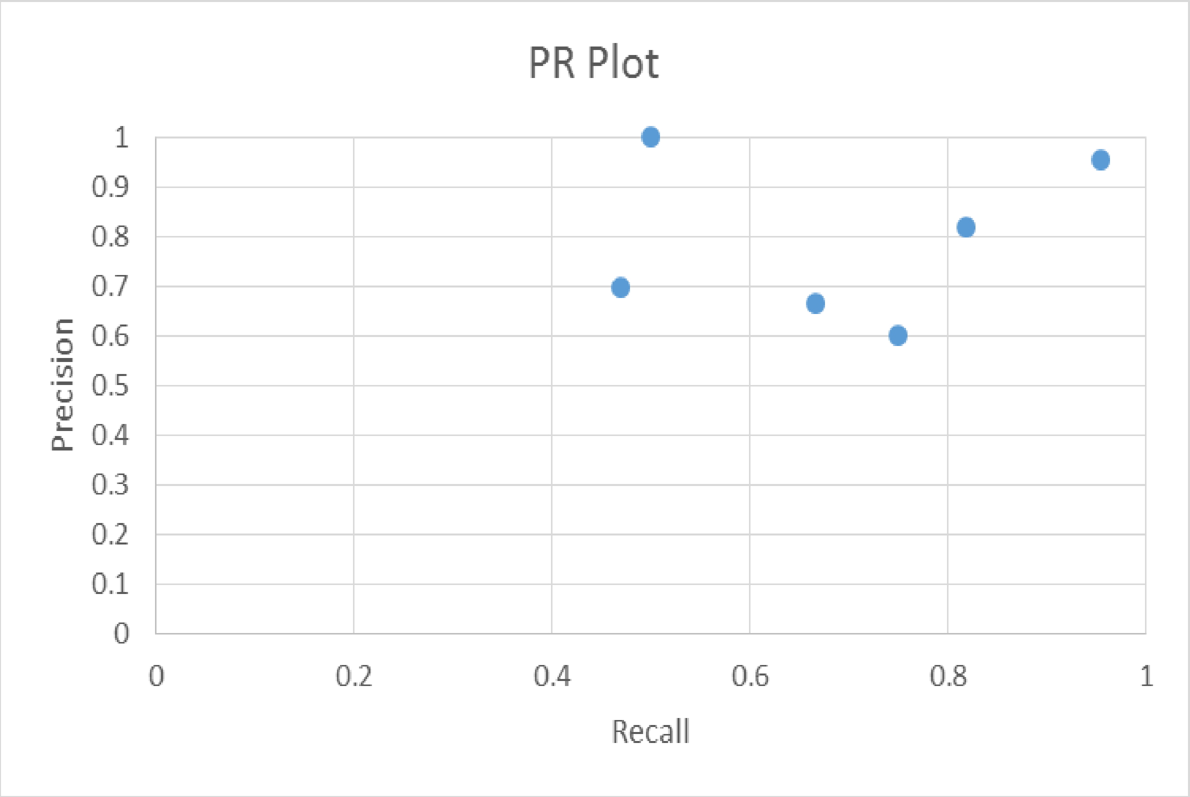


Fig 5: PR Plot

PR plot for the random forest classifier is shown in Fig 5. It was observed from this plot that the ratio of appropriate data fetched by the classifier needs to be improved further to improve the reliability levels of the classifier.

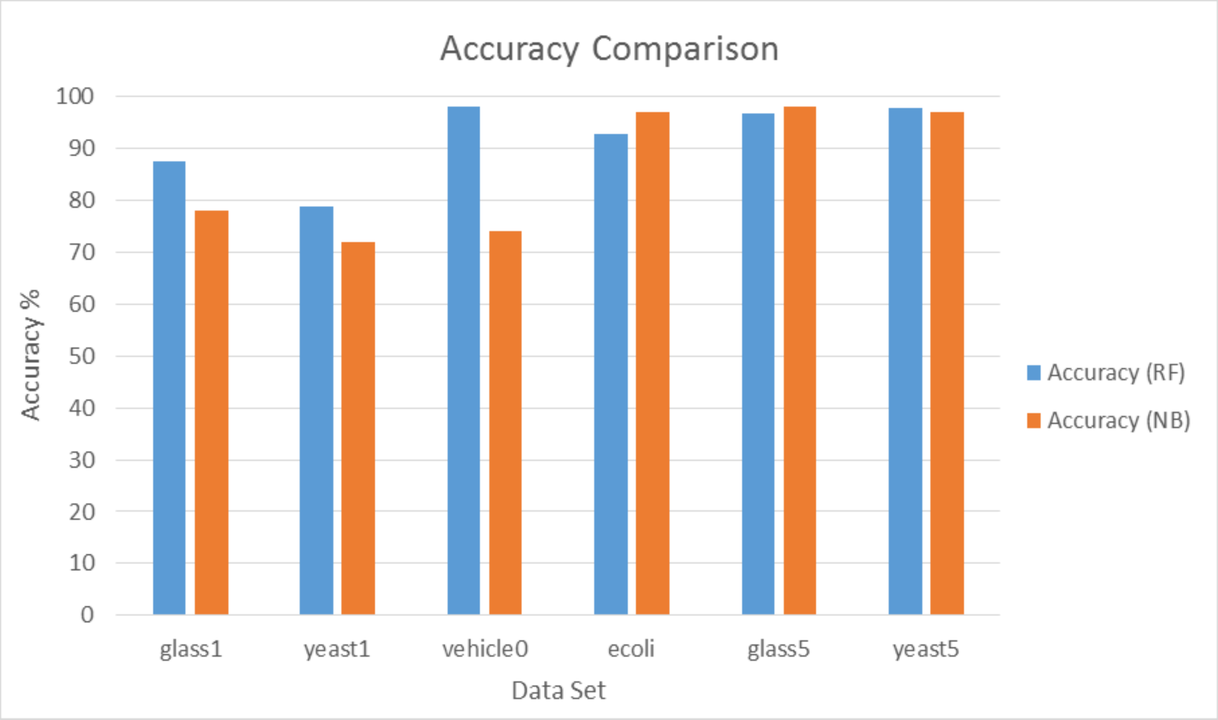


Fig 6: Accuracy Comparison

A comparison of the accuracies of the Random Forest and the Naïve Bayes classifier is shown in Fig 6. It could be observed that in most of the cases, the Random Forest classifier exhibited higher accuracy when compared to the Naïve Bayes classifier. It was observed that the Naïve Bayes classifier exhibited a mean accuracy level of 85.83%, while the Random Forest classifier exhibited a mean accuracy level of 92.16%.

CONCLUSION

Online intrusion detection is one of the major necessities due to the raise of ecommerce transactions. An effective intrusion detection technique is proposed in this paper which also handles Big Data. The proposed approach uses Random Forest algorithm, an ensemble of several decision trees to build the classifier rules. The usage of several rule creating algorithms provided an added advantage to the algorithm by making it immune to imbalance and missing data. The proposed algorithm was tested on datasets containing low to moderate imbalance and was found to scale well both in terms of accuracy and reliability. Future enhancements of this algorithm would include testing the algorithm's scalability level on datasets with huge imbalance.

REFERENCES

[1] Singh, R., Kumar, H. and Singla, R.K. 2015. An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Systems with Applications, Volume 42, Issue 22, Pages 8609-8624.

[2] Andreolini, Mauro, Colajanni, M. and Marchetti, M. 2015. A collaborative framework for intrusion detection in mobile networks. Information Sciences.

[3] Al-mamory, Safaa O., and Jassim, F.S. 2015. On the designing of two grains levels network intrusion detection system. Karbala International Journal of Modern Science 1.1: 15-25.

[4] Bul'ajoul, Waleed, James, A. and Pannu, M. 2015. Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer and System Sciences 81.6: 981-999.

[5] James and Anne. 2015. Optimisation, security, privacy and trust in e-business systems. Journal of Computer and System Sciences 81.6: 941-942.

[6] Rastegari, Samaneh, Hingston, P. and Lam, C. 2015. Evolving statistical rulesets for network intrusion detection. Applied Soft Computing 33: 348-359.

[7] Bhuyan, Monowar H., Bhattacharyya, D.K and Kalita, J.K. 2015. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recognition Letters 51: 1-7.

[8] Zheng, Kai, Cai, Z., Zhang, X., Wang, Z. and Yang, B. 2015. Algorithms to speedup pattern matching for network intrusion detection systems. Computer Communications 62: 47-58.

[9] Khalili, Abdullah and Sami, A. 2015. SysDetect: A systematic approach to critical state determination for Industrial Intrusion Detection Systems using Apriori algorithm. Journal of Process Control.

[10] Shittu, Riyanat, Healing, A., Ghanea-Hercock, R., Bloomfield, R. and Rajarajan, M. 2015. Intrusion alert prioritisation and attack detection using post-correlation analysis. Computers & Security 50: 1-15

[11] Lin, Wei-Chao, Ke, S. and Tsai, C. 2015. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Systems 78 : 13-21.

[12] Zuech, Richard, Khoshgoftaar, T.M. and Wald, R. 2015. Intrusion detection and Big Heterogeneous Data: a Survey. Journal of Big Data 2.1: 1-41.

[13] Cho, Jaeik, Shon, T., Choi, K. and Moon, J. 2013. Dynamic learning model update of hybrid-classifiers for intrusion detection. The Journal of Supercomputing 64, no. 2 : 522-526.

[14] Grzech and Piotr, A. 2009. Optimal monitoring system for a distributed intrusion detection system. Artificial Life and Robotics 14.4: 453-456.

[15] Vokorokos, Liberios, Ennert, M., Čajkovský, M. and Radušovský, J. 2014. Survey of parallel intrusion detection on graphical processors. Open Computer Science 4, no. 4: 222-230.

[16] Pawar, Nilkanth, S. and Bichkar, R.S. 2015. Genetic algorithm with variable length chromosomes for network intrusion detection. International Journal of Automation and Computing: 1-6.

[17] Wang, Hui, Zhang, G., Mingjie, E. and Sun, Na. 2011. A novel intrusion detection method based on improved SVM by combining PCA and PSO. Wuhan University Journal of Natural Sciences 16, no. 5: 409-413.

[18] http://sci2s.ugr.es/keel/datasets.php

Author's biography with Photo

The authior S.J.SATHISH AARON JOSEPH , is working as the head of the department of computer Applications in J.J.College of arts and science (Autonomous) , Pudukkottai,Tamil Nadu.At present he is a part time research scholar in Ph.D computer science under the guidance of Mr.R.Balasubramanian, Professor , P.G and Research Department of Computer Science, J.J.College of arts and science (Autonomous), pudukkottai,Tamil Nadu.