Network Intrusion Detection Using CatBoost Algorithm

Vinay Jain
B.E Graduate, Netaji Subhas Institute of Technology, 2014-18

***Abstract*- With the rapid advancement in technology, the usage of devices generating digital data has surged and thus, resulted in increased network traffic. This has also raised the issues of network security commensurately as increased network traffic means increased vulnerability of data to hackers. Due to these reasons, intrusion detection system has been an important research issue. An intrusion detection system is like a defense mechanism that prevents unauthorized access to the data or network of an organization. Boosting algorithms are ensemble techniques which form a strong model from weak ones by taking into account the previous classifiers success. In this paper, an intrusion detection system is proposed using a boosting technique called CatBoost algorithm. A binary classification i.e., differentiating benign and malignant intrusions, and a multi-class classification i.e., identifying intrusions as benign or an attack type of the category DoS, Probe, U2R and R2L, is performed using CatBoost algorithm. Later the results from both types of classifications are analyzed to see the algorithm's efficiency in different detection scenarios.**

Index Terms- Intrusion Detection, NSL-KDD Dataset, Boosting, CatBoost algorithm, Classification, Accuracy, False Positive Rate, Detection Rate

I. INTRODUCTION

With the enormous growth of computer networks usage, they have become easy targets for intruders because as the network size increases, it's vulnerability to potential threats and misuse increases. It is of the utmost importance to find the best possible methods to make our system immune to such attacks. The security of a computer system is compromised when an intrusion takes place. Any kind of unauthorized access to an organization's network or an attempt to interfere with the integrity of the system can be termed as intrusion. And so, the need arises for an efficient intrusion detection system (IDS) [1], [2]. An IDS can be a software or hardware part of the system that automates the detection of security problems in a computer network or system. Boosting algorithms are ensemble techniques where they use multiple weak classifiers to create a strong classifier that has better performance than a single one. Boosting algorithms are similar to bagging algorithms [3], [4] in the sense that they both use ensemble methods for classification. They differ in the manner that bagging methods create N separate classifiers and work on them separately whereas on the other hand, boosting methods also create N classifiers but here, every classifier takes into account the success of its predecessor along with its classification process.

In this paper, CatBoost algorithm [5], [6] is used to create a machine learning model that can classify network intrusions in different scenarios. First scenario is binary classification where model differentiates between normal and attack categories. Second scenario is multi-class classification where model differentiates between benign and an attack type of the category DoS, Probe, U2R and R2L [7]. Later the classification results of the model in both scenarios is analyzed to see the algorithm is suitable for which scenario. The dataset used for classification purpose is the NSL-KDD dataset [7], [8].

The rest of this paper is organizes as follows: Section 2 is about the related work done, Section 3 briefly explains CatBoost algorithm, Section 4 summarizes the experiment and result analysis for this paper and Section 5 gives the conclusion of this paper.

II. RELATED WORK

Weiming Hu, Wei Hu and Steve Maybank [9] proposed a new method for intrusion detection using the AdaBoost algorithm. They presented a comparative analysis of AdaBoost with some other strong classifying techniques. The results were that this algorithm gave significantly low false positive rate with high detection rate and it had low computation complexity and error rates as compared to other already published approaches.

The authors of [10] proposed AdaBoost and XGBoost with K-Means clustering algorithm as a possible solution for intrusion detection. The dataset used by them was NSL-KDD dataset. They compared the performance between boosting algorithms with clustering and without clustering. They also did a comparative analysis with other classification techniques. The results were that the proposed technique of combining boosting algorithms with clustering algorithms gave much better results.

Kajal Rai, M. Syamala Devi and Ajay Guleria [11] use decision trees as an intrusion detection model for their paper. They have used NSL-KDD dataset for classification purpose. These paper gave insights into the efficiency of decision tree classifiers for intrusion detection purpose as they gave significant results in terms of accuracy and computation time as compared to other classifiers such as SOM, Hoeffding and C4.5.

The authors of [12] proposed a multi-layer machine learning model for intrusion detection. The model consisted of 3 layers. First layer uses principal component analysis to select a subset of features from the complete set of features. Second layer used genetic algorithm with negative selection to differentiate between normal and abnormal intrusions. Layer three consisted of several classifiers which labeled the detected anomalies. NBTree and RFTree proved to give the best results for detecting anomalous intrusions.

In [13], the authors perform a comparative analysis of various feature selection techniques like OneR, Relief, Chi-square and SVM on intrusion detection. The classifier used for this purpose was J48 classifier. They also proposed a combination of OneR and Relief feature selection techniques with J48 algorithm as a base classifier as a viable mean for intrusion detection purpose.

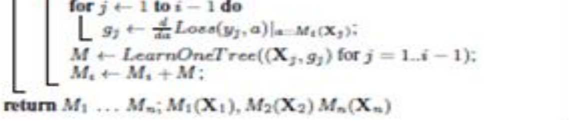
Bajaj and Arora [14] did a comparative analysis of various feature selection techniques with different classification algorithms for intrusion detection purpose. They used NSL-KDD dataset for their paper. Information gain, gain ratio and correlation based feature selection techniques were used. J48, Naive Bayes, NB tree, Multi-layer perceptron, LibSVM and SimpleCart were used for classification purpose. A summary of detection accuracy of various classifiers with above mentioned feature selection methods was presented in the paper with SimpleCart algorithm giving the best results.

III. CATBOOST ALGORITHM

Categorical features are a set of discrete values called categories with no relationship between them and hence, are difficult to evaluate via decision trees, the most popular base predictor of boosting algorithms. These features cannot be discarded in a machine learning problem as these categorical features contain important information in determining the outcome variable.

CatBoost algorithm have oblivious trees [15], [16] as their base predictors. CatBoost algorithm considers any combination of features as a new one. Every combination of features gives an even more powerful feature for the algorithm. For every new split for the current tree, CatBoost algorithm uses a greedy approach. Except for the first split, every next split includes every combination and categorical features in the current tree along with categorical features of the dataset. Every split in the tree, whether related to combinations of categorical features or categorical and numerical features, is considered as categorical with two values and converted to their numerical counterpart while execution.

CatBoost algorithm prevents overfitting by using unbiased gradients [5]. In CatBoost algorithm, for every model constructed after any number of trees/learners, every training example being evaluated is assigned a gradient value. To make sure that this gradient value being assigned is unbiased, the model needs to be trained without the particular training example. The idea behind unbiased gradients is to make sure that none of the training examples must be used for training the model. In a way this means no examples for the model to train on, which is preposterous. Therefore, CatBoost uses a second model which is never updated using a gradient estimate for this example. Later, this second model is used in scoring the resulting tree. The following algorithm [5] briefly explains this technique.



```
input : $\{(\mathbf{X}_k, Y_k)\}_{k=1}^{n}$ ordered according to $\sigma$, the number of trees $I$;
$M_i \leftarrow 0$ for $i = 1..n$;
for $iter \leftarrow 1$ to $I$ do
    for $i \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $i - 1$ do
            $g_j \leftarrow \frac{d}{da} Loss(y_j, a)|_{a=M_i(\mathbf{X}_j)}$;
        $M \leftarrow LearnOneTree((\mathbf{X}_j, g_j)$ for $j = 1..i-1)$;
        $M_i \leftarrow M_i + M$;
return $M_1 \ldots M_n; M_1(\mathbf{X}_1), M_2(\mathbf{X}_2) M_n(\mathbf{X}_n)$
```

Fig. 1: Updating the models and calculating model values for gradient estimation

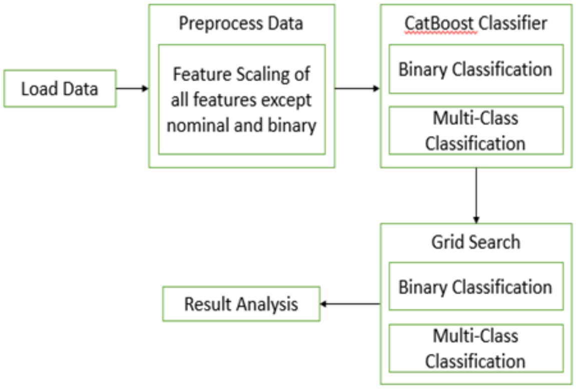
CatBoost algorithm makes the task of manually converting the categorical features to their numerical counterparts in the data preprocessing stage unnecessary. Another advantage is that unlike most boosting algorithms, CatBoost algorithm takes care of the overfitting problem inherently.

IV. EXPERIMENT AND RESULTS

A. Experiment Setup

Two scenarios are considered. First scenario is where the algorithm differentiates between benign and malignant intrusions i.e., a binary classification. Second scenario is where the algorithm identifies intrusion as benign and as DoS, Probe, U2R or R2L attack type. Grid search was also performed for both scenarios to find out the optimum parameters for best results. Later a comparative analysis of both scenarios, with before and after grid search, was also done.

Fig. 2: Experiment Setup



B. Result Analysis

Parameters used before grid search (Set 1):

1) Binary = {depth: 6, iterations: 100, learning_rate: 1, loss_function: 'Logloss', eval_metric: 'AUC', use_best_model: True}

2) Multi-Class = {depth: 6, iterations: 100, learning_rate: 1, loss_function: 'MultiClass', classes_count: 5, use_best_model: True}

Parameters used after grid search (Set 2):

1) Binary = {depth: 2, iterations: 1000, learning_rate: 0.01, l2_leaf_reg: 100, border_count: 5, loss_function: 'Logloss', eval_metric: 'AUC', use_best_model: True}

2) Multi-Class = {depth: 1, iterations: 500, learning_rate: 0.03, l2_leaf_reg: 5, border_count: 200, loss_function: 'MultiClass', classes_count: 5, use_best_model: True}

From Fig. 3 and Fig. 4, it can be seen that there is not much difference between the classification results before and after grid search in the case of binary classification. Only the false positive rate of normal category decreased and detection rate of attack category increased by about 1 unit after grid search.

Fig. 3: Classification results for Binary Classification

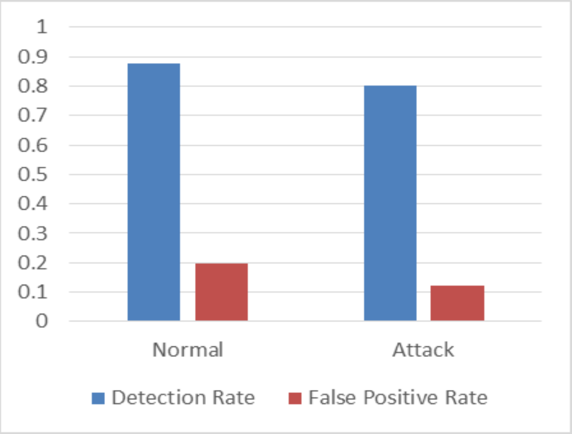
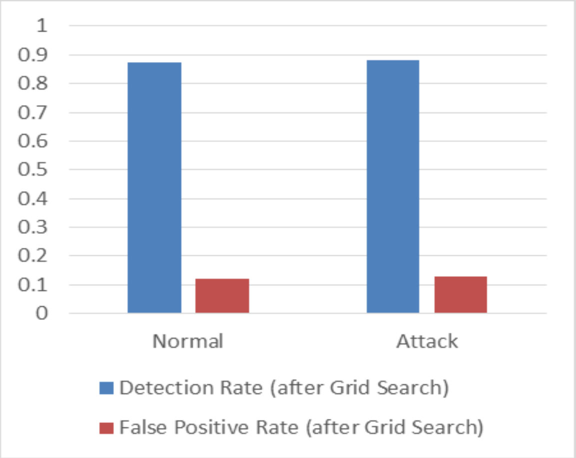


Fig. 4: Classification results for Binary Classification (after grid search)



From Fig. 5 and Fig. 6, significant changes can be observed in multi-class classification results, before and after grid search. Detection rates of Normal, Probe and DoS increased by about 1 unit after grid search. False positive rates of the same categories decreased by about 1 unit after grid search. But the most eye catching difference is the algorithm's inability to detect R2L and U2R attack categories after grid search. The reason perhaps must be the lack of sufficient number of training examples for proper training of the classification model.

Fig. 5: Classification results for Multi-Class Classification

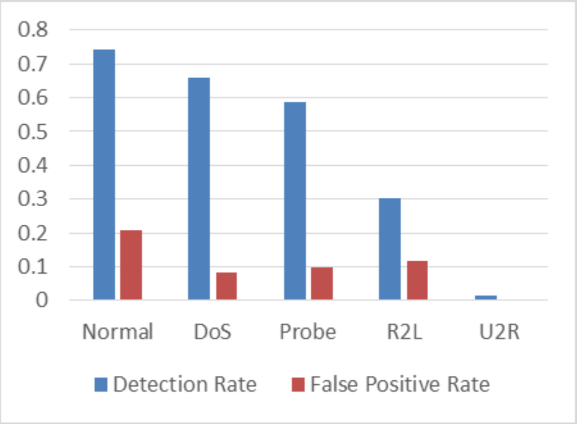


Fig. 6: Classification results for Multi-Class Classification (after grid search)

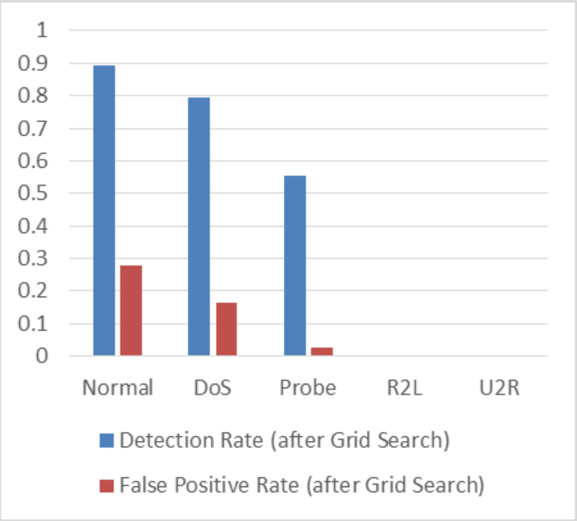


Fig. 7 and Fig. 8 represent the accuracy and computation time comparison respectively, for multi-class and binary classification before and after grid search. In case of multi-class classification, the accuracy was increased by appx 7% after grid search and the computation time of algorithm for this scenario with Set 2 parameters from grid search was appx. 15 seconds more than that with Set 1 parameters. In case of binary classification, the accuracy was increased by 5% after grid search and the computation time of algorithm for this scenario with Set 2 parameters was increased nearly by 4.5 times that with Set 1 parameters.

Fig. 7: Accuracy Comparison

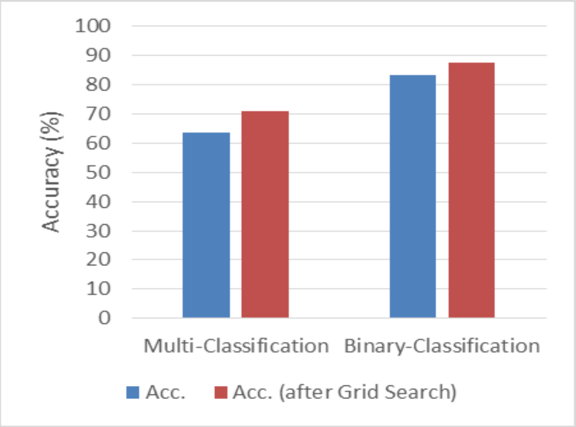
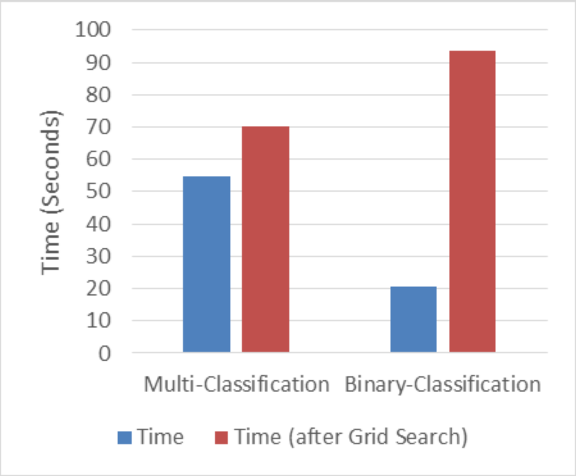


Fig. 8: Computation Time Comparison



V. CONCLUSION

The CatBoost algorithm poses itself as a viable solution for network intrusion detection. In multi-class classification scenario, where the model is supposed to classify intrusions as benign/normal or one of the attack categories (DoS, Probe, U2R and R2L), the CatBoost algorithm gives average classification results with accuracy=70.79%. However in binary classification scenario, where the model is supposed to classify intrusions as benign or malignant, the CatBoost algorithm gives astounding results with an accuracy of 87.65%. Hence, the CatBoost algorithm can be used as an efficient intrusion detection model for both scenarios.

ACKNOWLEDGMENT

Special thanks to M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani for introducing a dataset which acts as a benchmark for intrusion detection datasets and to

University of New Brunswick, Canada for making this dataset publicly available.

REFERENCES

[1] R.Bace and P. Mell, “NIST Special Publication on Intrusion Detection Systems”, 2001.

[2] Srinoy, S., Chimphlee, W., Chimphlee, S., & Poopaibool, Y., “A fusion of ICA and SVM for detection computer attacks.” In: Proceedings of the 5th WSEAS international conference on Applied computer science. World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 986-990.

[3] Kristína Machová, Miroslav Puszta, František Barčák and Peter Bednár, “A Comparison of the Bagging and the Boosting Methods Using the Decision Trees Classifiers”, ComSIS, Vol. 3, No. 2, December 2006, pp. 57-72.

[4] David Opitz and Richard Maclin, “Popular Ensemble Methods: An Empirical Study”, Journal of Artificial Intelligence Research, Volume 11, pages 169-198, 1999.

[5] Anna Veronika Dorogush, Vasily Ershov and Andrey Gulin”, CatBoost: gradient boosting with categorical features support”, retrieved from “https://arxiv.org/pdf/1810.11363v1.pdf”.

[6] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush and Andrey Gulin, “CatBoost: unbiased boosting with categorical features”, Proceedings: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018).

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[8] L.Dhanabal, Dr. S.P. Shantharajah, “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[9] Weiming Hu, Wei Hu And Steve Maybank, “Adaboost-Based Algorithm For Network Intrusion Detection”, Ieee Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 38, No. 2, April 2008, pp. 577-583.

[10] Parag Verma, Shayan Anwar, Shadab Khan and Dr. Sunil B Mane, “Network Intrusion Detection Using Clustering And Gradient Boosting”, In 9th International Conference on Computing, Communication and Networking Technologies, 2018.

[11] Kajal Rai, M. Syamala Devi and Ajay Guleria, “Decision Tree Based Algorithm for Intrusion Detection”, International Journal of Advanced Networking and Applications, Volume 07, Issue 4, pp. 2828-2834, (2016).

[12] A.S.A. Aziz, A.E. Hassanien, S. El-Ola Hanafy and M.F. Tolba, “Multi-layer hybrid machine learning techniques for anomalies detection and classification approach”, 13th International Conference on Hybrid Intelligent Systems (HIS), 2013, IEEE.

[13] Krishan Kumar, Gulshan Kumar and Yogesh Kumar, “Feature Selection Approach for Intrusion Detection System”, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 2 , No. 5, pp. 47-53, (2013).

[14] K. Bajaj and A. Arora, “Improving the Intrusion Detection using Discriminative Machine Learning Approach and Improve the Time Complexity by Data Mining Feature Selection Methods”, International Journal of Computer Science, vol. 76, Aug, 2013.

[15] R. Kohavi and C.-H. Li, “Oblivious decision trees, graphs, and top-down pruning”, In IJCAI, pp. 1071–1079. Citeseer, 1995.

[16] P. Langley and S. Sage, “Oblivious decision trees and abstract cases”, In Working notes of the AAAI-94 workshop on case-based reasoning, pp. 113–117. Seattle, WA, 1994.