International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Network Intrusion Detection System using XG Boost

M. S. Siva Priya, Bipin Kumar Sahu, Badal Kumar, Mayank Yadav

Abstract ***Internet is the most widely used commodity throughout the world. Such widescale adoption of internet has resulted in drastic developments across various facets of life. Several studies indicate a surge in cybercrimes including incidents of personal privacy thefts. Network intrusion is any illegitimate and/or unidentified activity taking place over a network. So, an effective intrusion detection system is required to be developed. Through this paper, we propose an intrusion detection system that uses XG Boost algorithm to detect intrusions. To implement this approach, KDD-99 dataset has been used for inputs. This paper demonstrates that the efficiency and accuracy of intrusion detection system deployed using XG Boost algorithm is better than contemporary algorithms.***

Keywords: IDS, Intrusion Detection, KDD-99, XGBoost.

I. INTRODUCTION

The widespread use of internet has made it an integral part of our lives. Internet is used to perform a variety of tasks like browsing social media, executing financial exchanges, which involves sharing of private data such as passwords and sensitive details like credit card credentials. The internet has evolved to such a level that people can act remotely from anywhere in the world. Any type of intrusion over the network gives the hackers an opportunity to get away with stolen data which in turn can be sold in black markets. Hence, an effective network intrusion detection system (abbreviated henceforth, as IDS) is required for detecting the type of attacks. The intrusion detection systems can be considered complementary to firewall systems as they identify attacks missed by firewalls.

In this paper, we propose an IDS using XG boost (eXtended Gradient boosting) algorithm to detect a potential attack over a network. XG Boost is an open source, efficient and popular implementation of gradient boosted decision trees. It belongs to a broader umbrella of Distributed Machine Learning Community (DMLC). It was made by the contributions of many developers with main contributions from Tianqi Chen and Carlos Guestrin [1]. This algorithm has been used by several Kaggle competition winners since then, along with several practical industrial applications. One of the main reasons behind its success is because the library strictly prioritizes computational speed and model performances.

This project utilizes the KDD cup 99[2] dataset. It is a widely used dataset in the study of network intrusions. The dataset contains 4.8 million instances. The characteristics of the attributes used in the dataset are categorical and integer in nature. The following figure (Fig.1) shows a typical deployed Intrusion Detection System.

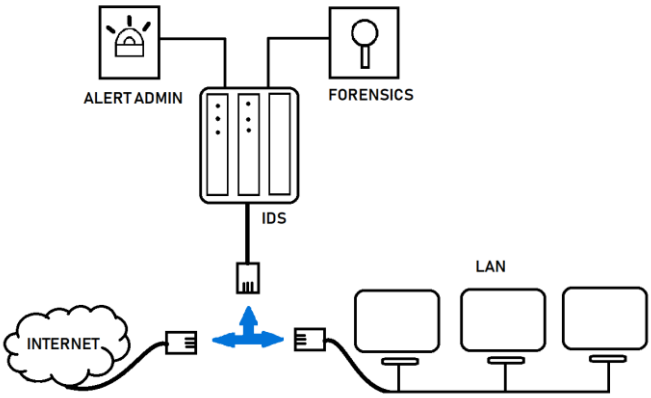

Fig.1 Model of an IDS

If the dataset is used in its original form as an input to the algorithm, it increases the complexity of the process and thus, the execution time. Hence to tackle this issue, attribute reduction is performed on the entire dataset which makes the computation less complex. After reduction, the dataset contains only 5 headers and hence a reduced KDD-99 dataset is created [3]. During the training phase sixty percent of the dataset is given as input to the XG boost algorithm.

II. THEORETICAL BACKGROUND

A. Intrusion Detection System (IDS)

An Intrusion detection System is a tool that recognizes an intrusion carried out on a network and tries to protect the system from the attack. An IDS can be integrated into a network or an individual system. It tries to determine whether the traffic or activity on a network is malicious or not. There are various techniques that allow an IDS to distinguish an attack from a genuine access such as anomaly detection or signatures of attack.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Bipin Kumar Sahu*,Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India. Email: bipinkumar_sahu17@srmuniv.edu.in
Badal Kumar Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.. Email: badal_kumar17@srmuniv.edu.in
Mayank Yadav Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.. Email: mayank_yadav@srmuniv.edu.in

Signature-based IDS detect known attacks by recognizing their pattern of behavior. This type of IDS is highly efficient in detecting known attacks with low false positive rates but fall behind in detecting newer attacks since the IDS has not been trained about the behavior of this new attack.

IDS based on anomaly detection model monitors the usual activity of a system and detect divergence from conventional behavior as an anomaly or attack. They offer an edge against signature-based IDS as they can detect unknown attacks providing better security to a network. There is another type of IDS known as hybrid IDS, which combines both the above-mentioned approaches for better detection of intrusions [4]. The security of today's computer networks can be improved through an IDS. The importance of such systems is immense as they are utilized for the detection of any hostile undertaking. Hence it is concluded that the significance of IDS is unquestionable.

B. Evolution of algorithms

In the field of machine learning, we come across various types of data and techniques to process that data. For example, unstructured data involving texts or images are best handled using neural networks but while dealing with structured/tabular data, decision tree-based algorithms have proven to be the best. The evolution of such algorithms (Fig. 2) is shown below:

1. Decision Trees: A decision tree is a tree-like graph structure that branches to represent decisions and their possible outcomes or consequences including outcomes of chance events, costs of resources, and utility.
2. Bagging: It is a simple and powerful ensemble algorithm that increases the stability and efficiency of machine learning algorithms employed in classification and regression, specifically decision trees. Bagging is also termed as bootstrap aggregation.
3. Random Forest: It is an ensemble learning method that operates by building a large number of decision trees when it being trained and gives out an aggregate of the votes or outputs from each decision tree to predict the final class of the object being tested.
4. Boosting: It is a group of machine learning algorithms that uses weighted averages for reducing bias, and variance in supervised learning and it converts weak learners to strong learners.
5. Gradient Boosting: It is a machine learning technique, giving rise to a strong prediction model as a collection of weaker prediction models, mostly decision trees.
6. XGBoost: XGBoost is an ensemble tree method that applies the principle of boosting weak learners using the gradient descent architecture. However, XG Boost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

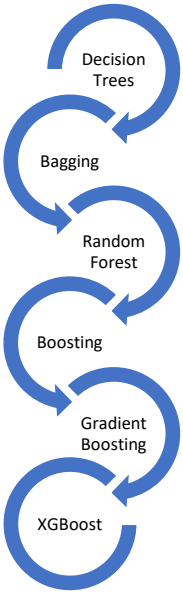

Fig. 2 Evolution of algorithms

III. PROPOSED METHODOLOGY

A. About XG Boost Algorithm

The XGBoost algorithm was created by Tianqi Chen [1] while researching on variants of tree boosting. Szilard in his study of benchmarking various classification and regression algorithms concluded that XGBoost is fast, efficient and highly accurate (Fig. 3) [6].

This algorithm stands out amongst other algorithms due to the following features:

- Cloud Integration: Supports Azure, AWS and Yarn clusters and works well with Flink, Spark, and other ecosystems.
- Versatility: Used for solving ranking, classification, regression and user-defined prediction problems.
- Cross-platform ability: Run smoothly across various operating systems like Linux, Windows, and OSX.
- Multi-language support: It has frameworks built for C++, Python, R , Julia and Scala

The various features of XGBoost include:

- **Regularization:** It performs L1 and L2 regularization to prevent over-fitting and improve overall performance.
- **Parallel processing:** XGBoost uses parallelized approach for sequential tree building.
- **Tree Pruning:** XGBoost uses ‘max_depth’ parameter, and starts paring down trees backward. This ‘depth-first’ approach improves computational performance significantly**.**
- **Sparsity awareness:** A unique attribute of XG boost algorithm is its ability to understand even inadequate amounts of data by 'learning' what can be the most appropriate missing value based on the training loss.

- **Cross validation:** Another feature of this algorithm is the fact that each iteration is characterized by its own cross validation methodology. So the search for the determination of the required number of boosting iterations during a single execution is not carried out explicitly

B. Regularized Objective Function

Here we use a regularized learning objective, which is mathematically represented as a sum of a differential convex loss function l, and a regularization function Ω.

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (1)$$

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2$

and, $l(\hat{y}_i, y_i) = -\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

in case of regression and

$$l(\hat{y}_i, y_i) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

for binary classifications.

$$\text{And }, \Omega(f_t) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

Where T is the number of leaves, and w_j^2 is the weight of j[th] leaf. The optimum weight w_j^* is given as:

$$w_j^* = -\frac{G_j}{H_j + \lambda} \quad (2)$$

In case of an iterative algorithm objective function can be written as:

$Obj^{(t)} = \sum_{i=1}^{N} L(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \sum_{i=1}^{t}\Omega(f_t)$ (3)

The performance is improved using first and second order gradient descent such as:

$$\partial_{\hat{y}_i^t} Obj^{(t)}$$
$$\partial_{\hat{y}_i^t}{}^2 Obj^{(t)}$$

C. Split Finding Algorithm

The split finding algorithms help in enumerating various splits across the features of a dataset. They are of two types:

- **Basic Exact Greedy Algorithm:** This algorithm finds every possible split throughout all the features in a dataset. It demands fairly more computational power especially when the data itself is highly unstructured
- **Approximate algorithm:** This algorithm [1] is helpful when the data doesn't entirely fit the memory. It basically proposes various splitting points as candidates based on feature distribution percentiles. Then the continuous features are mapped into buckets mapped by these split points. Based on these aggregated statistics and proposals, it finds the optimal solution.

Since the algorithm will be deployed over a distributed system or it will not be fitting entirely in the memory, we use the Approximate Split Finding algorithm.

A. Algorithm: Approximate algorithm for Split Finding

Begin

1. Initialize j to iterate through leaf nodes.
2. **for** k =1 to m do
3. Propose $S_k = \{s_{k1}, s_{k2}, \ldots s_{kl}\}$ by percentiles on feature k.
4. Proposal done on split basis locally or tree basis globally.
   end
5. **for** k =1 to m do
6. $G_{kv} \leftarrow = \sum_{j\in\{j|s_{k,v}\geq x_{jk} > s_{k,v-1}\}} g_j$
7. $H_{kv} \leftarrow = \sum_{j\in\{j|s_{k,v}\geq x_{jk} > s_{k,v-1}\}} h_j$
8. **end**

Stop

IV. IMPLEMENTATION

The proposed intrusion detection system functions according to the flowchart given in Fig. 4. Once the algorithm is deployed in the intrusion detection system, for every incoming packet that is captured, it is checked whether its registered to a flow. If it isn't registered then a new flow distinguished by few tuples like source and destination IPs and port addresses is made. A sequence file is created corresponding to this flow.

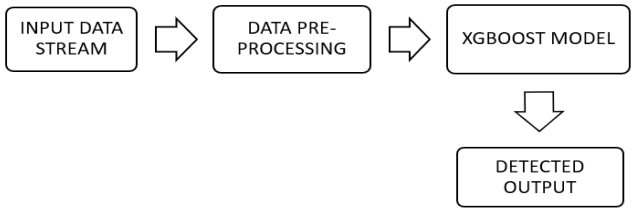

Fig. 4 Flowchart of XGBoost algorithm

When time duration exceeds threshold, it is sent for preprocessing where parameter calculation is done in order to measure the features. The code can run in parallel by taking the sequence file as input in a distributed environment. Lastly, the calculated feature values are passed to detection server where the scaled XGBoost model is deployed to determine whether the flow is a normal one or an intrusion, based on their feature values. The complete attack detection is shown in Fig. 5.

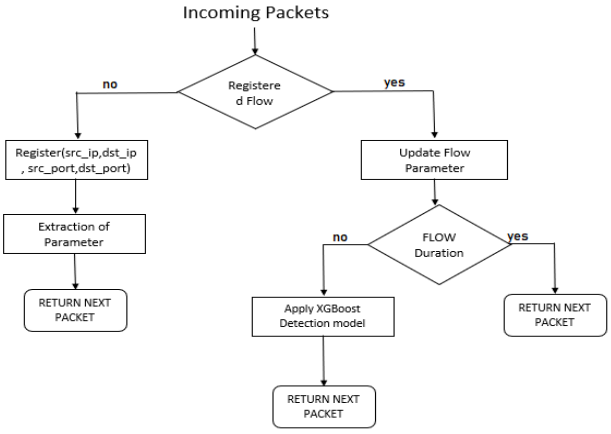

Fig. 5 Flowchart of XG Boost deployed in IDS

V. PERFORMANCE COMAPRISION

To perform a quick benchmark, we took the "Make_Classification"[7] dataset in Scikit-learn library of Python [8]. The dataset had about a million datapoints and 20 features. The prediction power & training time of different classification algorithms were measured as shown in Fig. 6 and Fig.7. As it is clear from the above-mentioned figure, XGBoost has clearly the best combination of processing power and training time. Other benchmarking studies have also given identical results [6].

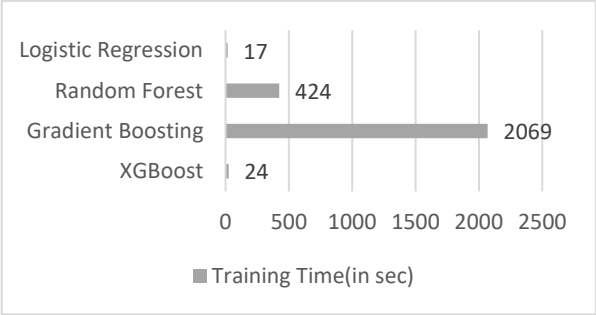

Fig. 6 Training time comparison of various algorithms

VI. CONCLUSION

This paper proposes an intrusion detection system based on XGBoost algorithm. The scalability along with added efficiency and processing speed has allowed it to realize attacks accurately, with minimum false positives in real time environments. For future works, deep learning frameworks such as MXNet can be used, to be able to do more complex computations in such networking scenarios.

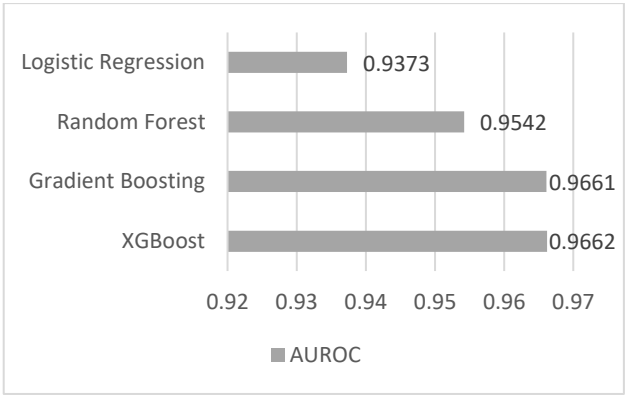

Fig. 7 AUROC Comparison of algorithms

ACKNOWLEDGMENT

We would like to take this opportunity to thank Mrs. R. Angeline for her continuous support throughout the entire duration of this project. This entire project would not have been possible without the support of Department of Computer Science and Engineering. We gratefully acknowledge SRM Institute of Science and Technology, Ramapuram for providing a platform to pursue this project. The dedication shown by our faculties ensured that the completion of this project.

REFERENCES

1. Tianqi Chen, Carlos Guestin, "XGBoost: A Scalable Tree Boosting System", *2016 SIGKDD Conference* arXiv :1603.02754v3 [cs.LG] 10 Jun 2016
2. KDD99. KDD99 cup dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.
3. Anand Sukumar J V, Pranav I, Neetish MM, Jayasree Narayanan, "Network Intrusion Detection Using Improved Genetic k-means Algorithm", *2018 International Conference on Advances in Computing, Communication and Informatics (ICACCI)*. IEEE, 2018.
4. Fan Zhang, Hansaka Angel Dias Edirisinghe Kodituwakku, J. Wesley Hines, and Jamie Coble, "Multi-Layer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data", *IEEE Transactions on Industrial Informatics*, 2019.
5. Syurahbil, Noraziah Ahmad, M. Fadly Zolkipli and Ahmed N. Abdalla, "Intrusion Preventing System using Intrusion Detection System Decision Tree Data Mining", *American Journal of Engineering and Applied Sciences*, 2009.
6. Szilard Pafka, "Simple/limited /incomplete benchmark for scalability, speed and accuracy of machine learning libraries for classification" URL: https://github.com/szilard/benchm-ml
7. sklearn.datasets.make_classification — scikit-learn 0.21.3 documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html
8. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

AUTHORS PROFILE

M.S Siva Priya has a Masters in Engineering in Computer Science and Enginnering having 2 years of industry experience along with 8 years of teaching experience. She is currently doing research (Ph.d) in Remote Sensing.

Bipin Kumar Sahu is currently pursuing his Bachelors of Technology degree as a pre final year student in SRM Institute of Science and Technology, Ramapuram, Chennai, India. His main research interest includes network information security, data science and machine learning.

Badal Kumar is currently pursuing the degree of Bachelor of Technology in Computer Science and Engineering from SRM Institute of Science and Technology, Ramapuram, Chennai. His current interest includes cyber security, machine learning and artificial intelligence.

Mayank Yadav is currently pursuing his Bachelor of Technology in Computer Science and Engineering from SRM Institute of Science and Technology, Ramapuram, Chennai. His field of interest includes cyber security, machine learning and artificial intelligence.