ISSN 2278-3091

Volume 9, No.1, January – February 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse55912020.pdf

https://doi.org/10.30534/ijatcse/2020/55912020

XGBoost Classification based Network Intrusion Detection System for Big Data using PySparkling Water

Tadepalli Anish Deepak [1] , Burra Vijaya Babu [2], Burra Meghana[3]

[1] M.TECH Student, Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Guntur District, Andhra Pradesh-522502, India. anishdeepak34@gmail.com

[2]Professor, Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Guntur District, Andhra Pradesh-522502, India. vijay_gemini@kluniversity.in

[3]Student, Prasad V. Potluri Siddhartha Institute of Technology (PVPSIT), Kanuru, Vijayawada, Andhra Pradesh-520007, India. burrameghana@gmail.com

ABSTRACT

A Machine learning is a technique for information investigation that robotizes model building. It is a part of man-made consciousness dependent on the possibility that frameworks can gain from information, recognize patterns and make data driven decisions on choices with negligible human intercession. Training machine learning algorithms with large volume of data (also known as big data) gives better result. Cloud Computing (CC) erases the barriers of handling the bigdata in terms of computation and storage. In this paper we are proposing a cloud-based Intrusion Detection System (IDS) using tree-based ensemble classification algorithm known as XGBoost classifier trained on CICIDS-2017 dataset which is a realistic cyber dataset which has benign and most up-to-date common seven different types network attacks. Sparkling Water enables clients to join the quick, versatile machine learning functionalities of H2O with the capacities of Spark. The proposed IDS using XGBOOST classifier from H2O.ai generated good results when compared with other algorithms like Random Forest (RF), artificial neural network (ANN), gradient boost (GBM), and stack ensemble method. Out of all algorithms XGBoost gave 99.8% accuracy on validation set and nearly 99.1% accuracy on test set form k-fold cross validation.

Key words: Machine learning, Ensemble Learning, Cloud computing, Cluster computing, Network security, Bigdata.

1. INTRODUCTION

Web and related innovations, for example, systems administration and distributed computing (cloud computing) are the foundation of many utilizations and administrations that are utilized for every single part of present-day of life, from work, preparing to amusement. Ericsson evaluated that by 2020 there will be over 8.6 billion associated telephones, 1.7 billion associated PCs and tablets, and 18.1 billion IoT gadgets. With the expanding reliance on-line administrations, digital security assaults symbolize a preeminent danger confronting clients and associations. The object of Intrusion Detection System framework is at distinguishing and checking malignant activities. A large portion of the advanced IDSs can be partitioned into two principal classes that are Misuse detection and Anomaly based detection. Misuse detection is based on the knowledge inferred from the pattern of pervious attack. We attempt to recognize the intrusion pattern that has been recognized and stated by the experts earlier. It cannot detect unknown errors. In order to overcome the above issue, Anomaly identification strategies were created, and it depends upon the theory that it compares the existing state of network to the usual behavior and looks for malicious behavior. Thus, anomaly detector can detect both known and unknown attacks.

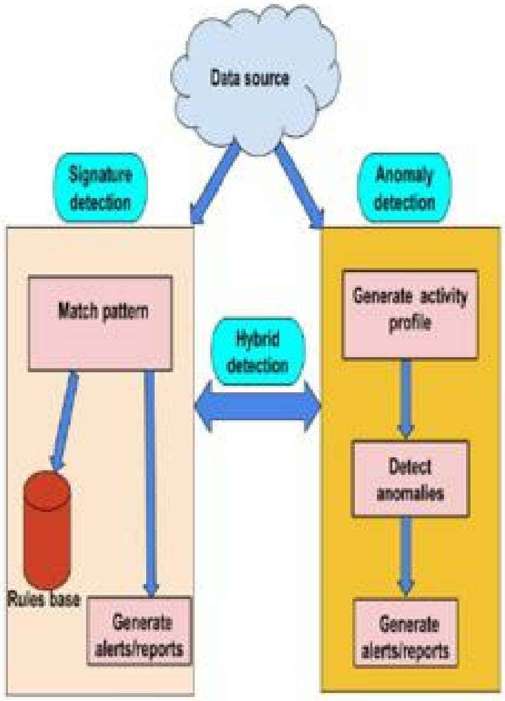

Figure 1: Detection Techniques used by IDS

The execution of anomaly intrusion detection system is a tedious task which has to analyze the patterns in huge volume of incoming traffic with standard alone servers that Are limited in computational resources [1-5]. On the other hand, for the quick change from the network technology, organizing detail transmission need broken through that

GB-per-second limit, accomplishing rates about TB for every second. Therefore, to facilitate safety against intruders in this high-volume and high-speed information environment, we are seeking for the cloud resources as to overcome the above-mentioned challenges like computing resources. The Intrusion Detection System needs to check the system traffic and discover interruptions while satisfy the necessities of taking care of the large information [8]. Distributed computing is the on-request conveyance of IT assets over the Internet with pay-more only as costs arise estimation. Rather than purchasing, owning, and keeping up physical server farms and servers, you can get innovation administration for example computing power, storage, and databases, depend upon the situation from a cloud provider. The rest of the paper is structured as follows. Section II affords precede research works on network intrusion detection. Section III exists the methodology of research study, while part IV describes the prototype system. Section V provides the contrast results, while section VI concludes the paper.

1.1 Objectives

The main objective of our analysis is:

- To find out the malicious network access.
- To produce the best Intrusion Detection model utilizing ML techniques like ANN, RF, GBM and XGBoost.
- To avoid the limitations of computation on bigdata, cloud is used for model building and spark is utilized to fasten the training process.

.

2. Related work

Before executing any Intrusion Detection System one ought to think about two principle viewpoints: the first is improving the recognition rate that is precision and second one is limiting the bogus alert rate which means IDS that neither identify an interruption as ordinary traffic nor hinder a typical traffic as interruption. To locate the concealed examples in tremendous size of information practically all organizations began utilizing the AI systems. The machine learning will analyze the hidden patterns in the huge volumes of data and find anomaly if anything is there in network traffic. The two main categories of machine learning algorithms are supervised and unsupervised. Supervised algorithms include K- Nearest Neighbor (KNN) [9,10and 11] Decision trees, Support Vector Machines (SVM) [12,13 and 14] which are come under classification and requires already labeled data for training the model. Unsupervised algorithms Include K-means clustering, Hidden Markov models which doesn't require any labeled data for training process. To improve the accuracy of IDS some researches introduced some optimization techniques like Particle Swarm Optimization (PSO) with machine learning algorithms like KNN and experimental results shown improvement in model accuracy [10].

To Implement a powerful Intrusion Detection System (IDS) with better exactness we need a dataset which incorporates late network intrusions to prepare the machine learning calculation to identify the intrusions. There are elven datasets exists from 1998 to 2017 because of the expansion in utilization of such huge number of online exchanges and cloud applications for corporate use. The previous decade confronted parcel of digital assaults which are not fully informed because the datasets do not contain proper information. CICIDS2017 dataset contains benign and the most modern basic assaults, which looks like the genuine real time information [11]. Since the dataset is enormous in dimensionality, we pick to utilize Apache spark for better performance.

Apache Spark is a unified computing engine and a set of libraries for parallel data processing on computer clusters, helps diverse extensively used programming languages (Python, Java, Scala, and R), contains libraries for various tasks going from SQL to streaming and machine learning, and run wherever from a PC to a gathering of thousands of servers. Spark is a structure subject to Hadoop developments that gives unmistakably more flexibility and usability than regular Hadoop [12]. It is probably the best tool for managing and inspecting large datasets, which are called Big Data. Moving to the Big Data Era requires considerable iterative process on huge datasets. Standard implementations of machine learning algorithms require very powerful machines to be capable to run. Depend upon high-end machines is not advantageous due to their excessive charge and improper charges of scaling up. The idea of using disbursed computing engines is to distribute the calculations to multiple low-end machines (commodity hardware) instead of a single high-end one. This simply speeds up the learning phase and allows us to create better models.

H2O is an in-memory platform for appropriated, adaptable machine learning [13]. H2O utilizes natural interfaces like R, Python, Scala, Java, JSON and the Flow note pad/web interface, and works consistently with huge information advances like Hadoop and Spark. H2O gives executions of numerous well-known algorithms, for example, GBM, Random Forest, Deep Neural Networks, and Stacked Ensembles. H2O is extensible with the goal that developers can include information changes and custom algorithms of Their decision and access them through those clients. Information accessing and storing is simple. H2O makes it quick and simple to get insights of your information through quicker and better prescient displaying. H2O permits internet scoring and modeling in a solitary platform.

Sparkling Water (H2O+Spark) is H2O's integration of their platform inside the spark project, which combines the machine learning functionalities of H2O with all of the capabilities of spark [18]. Which implies clients can run H2O algorithms on spark group for the both exploration and organization reason, that is made practical because H2O and spark share the indistinguishable JVM, that allows in for consistent advances among the 2 frameworks. The driver program in sparkling water begins spark session which thus is utilized to make a H2O configuration this enables H2O administration at the spark cluster.

Pysparkling Water is the isolation of python with sparkling water. It enables clients to begin H2O functionalities on spark cluster from python API. Inside the sparkling driver program, the spark context, which utilizes py4J to start the driver JVM and java spark-context, is utilized to make the H2O setting so that H2O benefits starts in the spark ecosystem.

3. METHADOLOGY

3.1 Ensemble learning is a machine learning worldview where various models (frequently called "weak learners") are prepared to tackle a similar issue and consolidated to improve results. The fundamental theory is that when weak models are effectively consolidated, we can acquire progressively precise as well as strong models. Most of the time, these basic models perform not so well by themselves because they have a high bias (low degree of freedom models) or they have too much variance to be robust (high degree of freedom models). At that point, the possibility of ensemble techniques is to have decreasing bias and/or variance of such weak learns by joining a few of them together so as to make a strong learner (or troupe model) that accomplish better results.

3.1.1 Bagging (Bootstrap Aggregation) is one of the ensemble creation strategy which uses homogeneous weak learners as base model, which pick test on N samples (bootstrap tests) from population of M sample. Notwithstanding, the determination is totally independent in each emphasis, with the goal that each sample is likely to be chosen in every iteration. After the bootstrap sub-samples are Made separate models are utilized to prepare on each sub sample. The absolute last yield predictions from the all subset models are mixed and basic of the considerable Number of predictions from selective trees are utilized for a final output that is more noteworthy powerful than a single decision choice tree [14].

3.1.2 Boosting is another ensemble strategy like bagging and it also use homogeneous weak learners as base model and utilizes consecutive learning system. At the end of the method, we fit back to back trees, generally irregular examples, and at each progression, the goal is to comprehend net error from the earlier trees. If a given information is misclassified by hypothesis, the misclassified sample weights will be increase with the goal that the upcoming tree will concentrate on the samples which are having more weights than remaining samples. Finally boosting changes, a weak learner into better performing models with an effective classification strategy [14].

3.1.3 Stack ensemble is also one of the ensemble techniques but instead of homogeneous weak learner as based model it uses heterogeneous learners like deep neural network and gradient boost algorithms to create a strong predictive machine learning model.

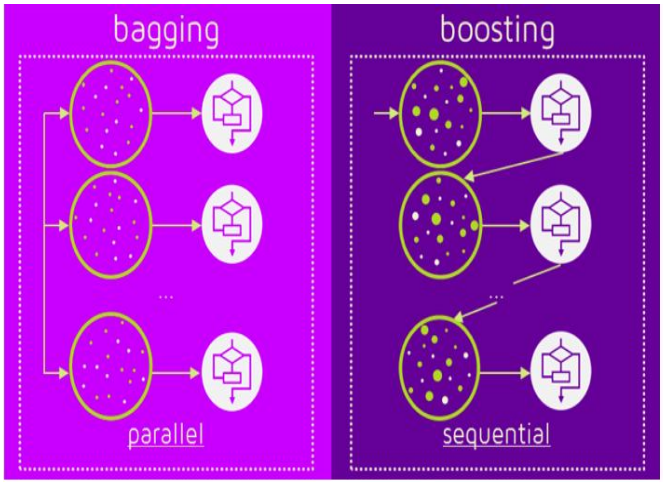

Figure 2: bagging and boosting

3.2 Machine Learning algorithms using bagging and boosting

As we seen above ensemble learning make strong classifiers by integrating different individual weak classifiers. Below we explained some of the machine learning algorithms using ensemble learning.

3.2.1 Random Forest which utilizes bagging as the ensemble strategy and decision tree is an individual mode (base learner). Joining different classifiers (e.g., decision trees) to construct a single strong learner is a propelled machine learning method with significantly improvement over single-based classifiers. As opposed to limiting Generalization error, random forest limits the training error, while being quick to prepare, demonstrated not to overfit, and computationally viable, (O(√ VTlogT),where V is the number of variables and T is the number of observations). These benefits make RF a potential instrument appropriate for adaptive classification problems [15].

3.2.2 Gradient boosting algorithm is an ensemble method which uses boosting as an ensemble technique. Gradient boost works like how the neural network minimize the error by optimizing the weights of inputs in each iteration of training process. But the gradient boost algorithm involves multiple learners in training phase, the prediction of each model is compared with the actual outcome. So, the difference between actual and predicted output is our error which is calculated by using loss function. The gradient is nothing but partial derivatives of loss function and helps us to find the direction towards the better accuracy by adjusting the parameters of weak learners. In each iteration gradient boost calculates the loss of a weak learner and adjust the parameters in next iteration based on gradient until the model reaches the accuracy threshold or predefined number of iterations [16]

3.2.3 Xtreme Gradient Boost (XGBoost) represents the Extreme Gradient Boosting [17,18]. It is a particular usage of gradient boosting procedure which utilizes increasingly exact approximations to find quality tree model. XGBoost utilizes gradient descent, that means second fractional derivatives of the loss function, which provides grater information approximately the route of the gradients and a way to get the minimum of our loss characteristics. XGBoost has better generalization and learns faster compared to the regular gradient boost by utilization of regularization methods (L1 and L2).

Why gradient Descent? Gradient Descent is a technique which includes a vector of weights (or coefficients) where we calculate their partial derivatives with particular to zero. The thought process behind calculating their partial derivatives is to locate the nearby minima of the loss function (RSS), which is convex in nature. In basic words, gradient descent attempts to optimize the loss function by tuning various estimations of coefficients to limit the loss.

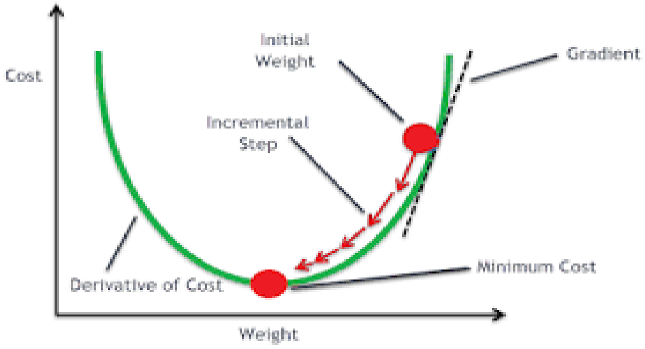

Figure3: Gradient Descent for reducing error

4. IMPLEMENTATION

Present IDS is implemented to secure constant identification in a high-volume and highspeed organize condition and to guarantee higher exactness, this intrusion detection model to foresee if there is an interruption, with a tree-based ensemble multinomial classification model to recognize the intrusion category.

4.1 Databricks Cluster

The proposed framework executed so as to grandstand the real time intrusion detection system using ensemble learning the utilization of distributed computing. The application is facilitated in Databricks. Databricks is an overseen platform for running Apache Spark - that implies that you need not to learn entangled bunch the configurations nor work repetitive support undertakings to take advantage of Spark. Databricks moreover offers a large group of highlights to enable its clients to be progressively gainful with Spark. It's a point and snap stage for those that pick a UI like data researchers or data analysts. Nonetheless, this UI is going with by means of a refined API for those that need to computerize elements of their information remaining tasks at hand with machine learning. To address the issues of endeavors, Databricks moreover comprises of focuses, for example, role-based access control and other sensible optimizations that not only improve usability for users but also decrease costs and complexity for administrators.

4.2. Spark Cluster Configuration

- Cluster mode=> standard
- Databricks run-time version=>5.5 LTS, Apache spark2.4.4, scala 2.11
- Python version=>3 and above
- H2O dependency libraries=>colorama_0.3.8, future, request, tabulate

The above dependencies have to be installed after h2O_pysparkling_2.4 package in pypi format.

4.3. Model Building

The significant reason for choosing H2O is its colossal acknowledgment for machine learning and the way that it manages the POJO based API administrations to the web application. Since the web application model would be valuable for network administrator, the online model is a vital piece of the framework.

H2o+spark means fast and scalable machine learning algorithms of h2O with capabilities of spark. Pysparkling package is used to initialize sparkling water, after that h2o's python package is used for model building. Dataset is divided into 3 parts, 70% of data for training, 15% for validation, and 15% for testing.

Dataset was stored in databricks default table as csv format then imported on to the notebook as spark dataframe and then convert to the h2O dataframe. Some preprocessing steps has done like removing NaN and Infinity type string values before passing to the XGBoost algorithm.

5. RESULT ANALYSIS

Apache spark also offering the machine learning algorithms like classification and regression from MLlib, but XGBoost from spark only supports binomial classification so that we can only identify weather the incoming traffic is begnin or attack .Therefore for handy managing and to pick out the attack name rather of simply figuring out the traffic is an attack, we are interested in XGBoost from h2O. The dataset is utilized on different algorithms available, out of all XGBoost carried out well.

Table 1: Types of Attacks and count in dataset

sno	Attack type	count
1	BENIGN	1982961
2	SSH-Patator	5897
3	Web Attack-Brute Force	1507
4	Dos solwloris	5796
5	Web Atttack- Sql Injection	21
6	DoS Hulk	230124
7	PortScan	158804
8	DoS Slowhttptest	5499
9	Bot	1956
10	Dos GoldenEye	10293
11	DDoS	128025
12	FTP-Patator	7935
13	Heartbleed	11
14	Web Attack-XSS	652

5.1 Measure the goodness of fit for classification

Logloss or logarthimic loss is one way to measure the goodness of our model at classification especially for multinomial classification. Best model has logloss value which is close to zero that means the machine learning model can predict the correct class label with higher probability. Table 2 represents the different metrics for classification on validation set. Since accuracies and R- square values for all models are very close on validation set, we preferred logloss to find out the best model which can classify each class with higher probability values.

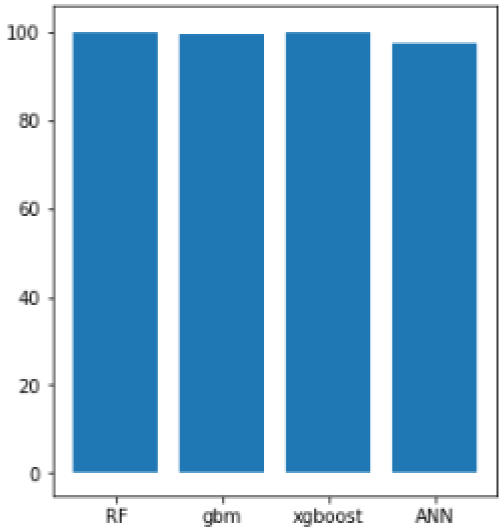

Figure 4: Accuracy for classification on validation set

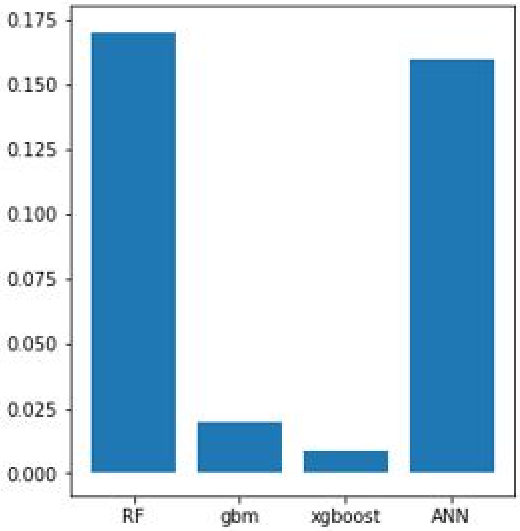

Figure 5: Logloss for classificaition on validation set

Table 2: Metrics for classification on validation set

Algorithms\ metrics	accuracy	R sqaure	RMSE	Log loss	Training time
ANN	97.3	98.9	0.19	0.16	90mins
Random Forest	99.7	99.8	0.52	0.17	60mins
Gradient Boost	99.6	99.8	0.61	0.02	33mins
XGBoost	99.8	99.9	0.37	0.009	30mins

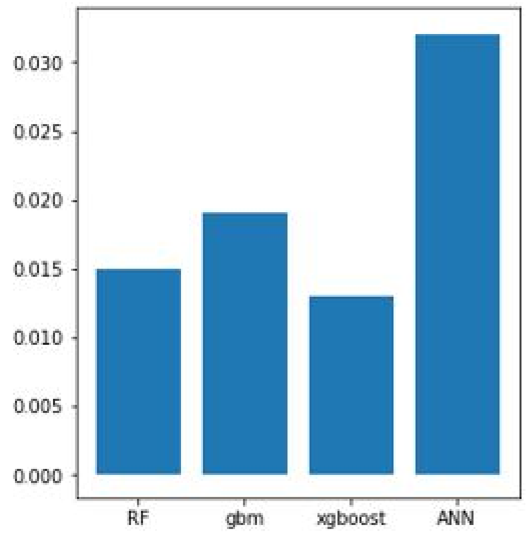

Figure 6: Logloss for classification on test set

5. CONCLUSION

The transparent integration of H2O with spark ecosystem, (MLlib and H2O side by side) we got most efficient model which can detect the 14 different types of intrusions over 7seven real time network attacks. From the above results it clearly visible that XGBoost algorithm showed very lower logloss value on validation set is 0.009 and logloss for test set is 0.011 after 5-fold cross validation and with minimum training time of 30 mins on such huge dataset. Then we can conclude that we had best generalized model to detect the real time cyber-attacks using cloud computing with affordable price.

REFERENCES

[1.] D. Stiawan, A.H. Abdullah, and M.Y. Idris, **"The trends of intrusionprevention system network, in 2010" 2nd International Conferenceon Education Technology and Computer**, vol. 4, pp. 217-221, June2010. https://doi.org/10.1109/ICETC.2010.5529697

[2] Singh R., Kumar H., Singla R.K., and Ketti R.R**. "Internet attacksand intrusion detection system: A review of the literature"OnlineInform**. Rev., 41 (2), pp. 171-184, 2017.CrossRefView Record inScopusGoogle Scholar.

[3] Liao H.-J., Lin C.-H.R., Lin Y.-C., and Tung K.-Y. **"Intrusiondetection system: A comprehensive review" Network Computing**.Appl., Rev., 36 (1), pp. 16-24,2013 https://doi.org/10.1016/j.jnca.2012.09.004

[4] A. Uzma, K.K. Dewangan, D.K. Dewangan, **"Distributed Denial of Service Attack Detection Using Ant Bee Colony and Artificial Neural Network in Cloud Computing**," in: B. Panigrahi, M. Hoda, V. Sharma, S. Goel (Eds.), Nature Inspired Computing, Advances in Intelligent Systems and Computing, Vol. 652, Springer, Singapore, Singapore, pp. 165-175, 2018.

[5] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, "**A survey of intrusion detection techniques in cloud," Journal of Network and Computer Applications**, Vol. 36, No. 1, pp. 42-57. 2013 https://doi.org/10.1016/j.jnca.2012.05.003

[6] X. Zhao, W. Zhang, **"Hybrid Intrusion Detection Method Based on Improved Bisecting K-Means in Cloud Computing**," 13th IEEE Web Information Systems and Applications Conference (WISA), Wuhan, China, pp. 225-230, 2016.

[7] S. Zaman, F.Karray. **Features selection for intrusion detection systems based on support vector machines**[C]//Consumer Communications andNetworking Conference, 2009. CCNC 2009. 6th IEEE. IEEE, 2009: 1-8.

[8] N. Araújo, R.de Oliveira, A.A. Shinoda, et al. Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach[C]//Telecommunications (ICT),2010 IEEE 17th International Conference on. IEEE, 2010: 552-558.

[9] Y. Yi, J. Wu, W. Xu**. Incremental SVM based on reserved set for network intrusion detection** [J]. Expert Systems with Applications,2011, 38(6): 7698-7707. https://doi.org/10.1016/j.eswa.2010.12.141

[10] A. R. Syarif, W. Gata**. Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm**[C]//Information &Communication Technology and System (ICTS), 2017 11thInternational Conference on. IEEE, 2017: 181-186.

[11] License: http://www.unb.ca/cic/datasets/ids-2018.html [Acessed:14-SEP-2018]

[12**] Big Data Analysis: Ap Spark Perspective** Abdul Ghaffar Shoro α & Tariq Rahim Soomro

[13] **A COMPREHENSIVE OVERVIEW OF OPEN SOURCE BIG DATA PLATFORMS AND FRAMEWORKS** Pedro Almeida1, Jorge Bernardino1,2 1ISEC – Polytechnic of Coimbra , Portugal 2CISUC – Centre for Informatics and Systems of the University of Coimbra, Portugal a21180299@isec.pt , jorge@isec.pt

[14**] Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data** Taghi M. Khoshgoftaar, Member, IEEE, Jason Van Hulse, Member, IEEE, and Amri Napolitano

[15] **Random-Forests-Based Network Intrusion Detection Systems** Jiong Zhang, Mohammad Zulkernine, and Anwar Haque

[16] **Gradient boosting machines**, a tutorial[1*] and [2]Alois Knoll[2]

[17**] XGBoost: A Scalable Tree Boosting System** Tianqi Chen, CarlosGuestrin
DOI: http://dx.doi.org/10.1145/2939672.2939785

[18] https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5