Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 215 (2022) 499–508

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Innovative Data Communication Technology and Application

Comparison of Artificial Intelligence Algorithms for IoT Botnet Detection on Apache Spark Platform

Faaiz Anwar[a], S. Saravanan[b]

[a]*Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India*
[b]*Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India*

Abstract

The amount of network traffic generated from the Internet of Things (IoT) devices is massive which leads to Big data. In present scenario, IoT devices has taken a leap ahead in terms of growing technology and insecure network traffic flow. The sensitivity of the data in the IoT network is becoming very high, which concern the security in legal and privacy issues. Traditional Intrusion Detection System (IDS) is used as a primary line security to differentiate between the attack and benign network traffic flow. As the size of the network traffic capture increases, it becomes more important to handle such big data. Similarly, when the number of classes and instances increases, the complexity of the data also increases. In this paper, we propose a big data platform based intrusion detection system which can differentiate between the types of network traffic flow generated from the IoT devices. We compared the performance of deep learning algorithms with machine learning algorithms on Apache Spark platform and found that machine learning algorithms outperform deep learning algorithms with greater accuracy and less training time for the model. We evaluated our research based on real world network traffic dataset BoT-IoT [1] for an improved intrusion detection system in IoT network traffic.

Keywords: Intrusion Detection System (IDS); Machine Learning; Deep Learning; Apache Spark; Botnet; Internet of Things (IoT).

1. Introduction

IoT is one of the most important inventions in this new age of technology. Things which were impossible to do before are now happening seamlessly. We can connect with our everyday devices—kitchen appliances, cars, AC, TV—to the internet via embedded devices and seamless communication is possible between people, processes, and things. These devices provide a seamless transfer of data through wireless network without any interaction. IoT helps

* Corresponding author. Tel.: +91-916-560-3331.
E-mail address: faaizanwar13@gmail.com

10.1016/j.procs.2022.12.052

people to live in a more efficient and smarter way. Automated IoT devices has made our life way easier than before. IoT has shown a remarkable connection between wide range of physical devices. Now, it is possible to collect any kind of data from any place to any device [2]. IoT provides a path of performance from machines through the delivery chain till logistic operations. IoTdevices has helped a lot of organizations in cost reduction of entire process by making it manual to automatic. IoT hasalready cut down the labor cost and wastage of material, mean these funds are used in improving the delivery serviceas well as providing a transparency in the transactions.

In early 1980's, when IOT was introduced to the open world It was not feasible for everyone to enjoy the benefits of this technology. However, as we progressed through time the cost of IoT services reduced exponentially. However, as this service is made feasible for everyone there are new kind of problems which we are facing. We are now collecting huge amounts of data which is causing network traffic congestion by IoT networks. According to the International Data Corporation (IDC) [3], IoT devices will generate 79.4 zettabytes of data by year 2025. Total internet data used were represented as 4.4 ZB last year. By year 2025, its miles expected to grow to one hundred seventy-five ZB, where 79.4 ZB could be because of IoT gadgets. As the number of IoT devices increases and more information is shared between devices, the potential that a hacker could steal confidential information also increases. A hacker withright of entry into an IoT community can exploit a wide variety of devices with a flood on every other gadget by requests, which is typically referred to as a Denial of Service (DoS). The gadgets in those networks are then usedas a botnet to assault other networks and services. In 2016, one of the most inimical IoT attacks was Mirai, a botnet that infiltrated the domain name server provider Dyn and took down many websites for prolonged period of time in what is considered one of the most important distributed denial-of-service (DDoS) assaults ever visible [4]. Attackersgained access to the network by exploiting poorly secured IoT devices. Hackers are not only the most effective risk tothe internet of things; privacy is every other essential problem for IoT users. For an instance, if a company that makes consumer IoT devices could use those devices to steal and sell personal data of the users. Large organizations will have to deal with massive numbers of IoT devices in near future. Collecting and managing such huge amount of data from all these IoT devices will be a breath-taking task. IDS is considered as one of the most demanding cyber security architectures. New categories of malicious attack are paving the way for the transformation of Intrusion Detection System from last 40 years [5][6]. Earlier, all malicious attacks are stored in IDS and are compared to foreign attacks for intrusion detection [7]. As the number of DoS attacks increased, it is impossible to preserve all patterns of attack. New anomaly-based architecture is introduced to perform intrusion detection through AI. The current challenge in botnet detection is the increased number of traffic generated from IoT devices. Traditional methods are not capable of handling such a big amount of data which results in compromising the training time of the model.

To overcome such challenge, we are proposing a big data platform based IoT botnet intrusion detection system. We have taken a real world IoT network traffic dataset BoT-IoT [1] and have following contributions to this research:

- We have implemented Machine learning algorithm (Random Forest algorithm, Decision tree algorithm) on 5% of the entire dataset with all features and 10 best features on Apache spark platform.
- We have run 10-Fold cross validation with machine learning algorithms on Apache Spark.
- We have performed Feed Forward Neural Network (FFNN) and Sequential Neural Network (SNN) on 5% of the entire dataset with all features and 10 best features on Apache Spark.
- We have compared Machine Learning and Deep Learning performance in terms of Accuracy and Training time of the model.

2. Literature Review

In paper [8], Garg et al. calls for self-intelligence-primarily based IDS that may hit upon the intrusion and make it wise for the next community. To make it smart, machine mastering-primarily based IDS is shown capable of getting a fulfilling stage of detection, if an adequate training set is present within the dataset. The researchers in [8] used UNSW-NB15 dataset and implemented six algorithms to decide the correctness of every example, which include decision Tree, Random Forest Gini, support Vector machine, Logistic Regression, Random Forest IG, and Gaussian Naive Bayes. The usage of the Matplot library illustrates the accuracy of all the aforementioned algorithms, with Random Forest IG having the satisfactory accuracy 92.63% and Gaussian NB having the worst accuracy with 50.46%. Towards the issue of Anomaly detection, Alazzam et al. in [9] has approached a novel way of supervised machine

learning technique to find zero-day attacks. This approach follows different clustering algorithms to differentiate types of attacks. A zero-day attack is an attack in which the model has not interacted before. To overcome this problem, their approach of differentiating and accuracy of the model got promising results. In paper [10] by Bhatia et al., many new forms of threats are introduced. These attacks are not only more difficult to detect, but they also cause the most damage. The paper demonstrated that Autoencoder-based unsupervised classifiers are successful at predicting network activity and identifying abnormalities and assaults in industrial networks when trained on innocuous traffic data. They also beat several supervised ML classifiers when it comes to recognizing new and unusual assaults.

Tzagkarakis et al. in paper [11] gives a diagnostic technique for brief IoT botnet attack detection. The goal is to prohibit the impact of attack through separating affected IoT devices on the IoT side. Due to the restricted processing competencies of IoT devices, they are eager on developing an algorithmic method that wishes a little training and checking out statistics as feasible so one can assemble a correct IoT botnet assault detector. The suggested method in [11] is to build on a sparse representation architecture, with the decision threshold determined using only benign training cases. To test the sparse representation strategy, a single hidden layer autoencoder is used in [11]. An experimental assessment revealed that the suggested strategy outperforms others in terms of F1-score. Sriram et al. in paper [12] has used a deep learning approach for detecting IoT botnet which is totally relied on a network flow framework. The cause behind solving the problem is to secure smart City applications including water treatment traffic controls and other IoT-based devices. This framework in [12] is designed in such a way that it can handle a large amount of data in the existing Big Data environment. Their deep learning methods perform much better than machine learning algorithms. Their method for extracting features from network streams aids in achieving exceptional performance, allowing the complete payload data to be evaluated and differentiated between real and botnet assault activity.

MLlib library for machine learning algorithm is a considerable advantage in Apache Spark. Gupta et al. in paper [13] elaborates the work with ML libraries in Apache spark for feature selection process. The work has used correlation and chi-squared methods for selecting the features. This IDS helped in reducing the time and process for feature selection using Spark. This method is done on DARPA's KDD'99 dataset and NSL-KDD, a cleaned version of KDD'99 dataset. Gradient Boosted, Decision Tree (DT), Random Forest (RF), SVM, regression and NB algorithms are later implemented on this dataset. To evaluate algorithms, training time, accuracy, sensitivity, prediction time and specificity are used. According to paper [13], none of the algorithms gave an accuracy more than 93%. The method proposed in this work achieved better results with KDD99 dataset when compared to NSL-KDD dataset. In paper [14], authors are able to build an IDS that can handle huge variety of data in very less time. This architecture is built on the working of Hive SQL with unsupervised algorithms, which is built on top of Spark and Hadoop platform. UNB ISCX 2012 dataset is tested for this architecture of IDS. when experimented, the accuracy of the architecture went not more than 86.2% which is not acceptable for security settings in an IDS. Hadoop MapReduce is used along with Apache spark to build an IDS in paper [15]. Whenever feature selection is required, a sorting algorithm can be used to make sure the features of NSL KDD'99 dataset. Results are with accuracy of 92.03% and the detection rate of 99.38%. Testing time of the model is 0.32 sec.

Beside the noticeable link between IoT, Big data and AI in working and business development, resources are still insufficient when AI and big data involves together in IoT network traffic flow. To use huge amount of data collected from an IoT network, big data is the most useful technology to get the insights. Due to exotic large IoT data sets and compilation problem for distributed system in Big data technology platform, research in cyber securityfor IoT network traffic flow is lacking behind. Most of the studies in this field are inspired by joining AI with big data or AI with IoT without using big data technology. This makes a need to be focused on the research with AIon big data platform for IoT networks. Following a phase of thorough research, it is found that these three elements are combined by three works in some way or the other. Using Apache Spark as a platform, a new architecture for intrusion detection is presented in [16]. Multiple classifiers are used to detect attacks including DNN, SVM, RF, DTand NB. A raw dataset is tested to examine the framework which showed that the DNN classification algorithm camewith highest accuracy and prediction time. However, the accuracy is not more than 80%. Vinayakumar et al. [17], using big data, DL algorithms, FNN and MLP executed an IDS. This work is built on top of Spark and Hadoop. Yet Another Resource Negotiator is used to carry out this work with precision, accuracy, recall, f-measure, TPR, and FPRbeing the evaluation criteria. The upshot is that, the architecture is more powerful than any usual IDS. However, it isnoted that using multiclass classification, the accuracy rate dropped below 90% for detecting some attacks. To detectSYN-DoS attacks in IoT networks, Apache Spark has a number of supervised algorithms in MLlib which are used to

construct an architecture for IDS by the author in [18]. It is noted that the framework had considerable accuracy in a short duration. However, the architecture is designed for one attack.

The literature is mainly focusing on deep learning technique for detecting botnet in the network traffic flow. Since deep learning technique is a complex architecture and time complexity is high compared to machine learning technique. We require a modern technique with less time complexity so that the model can learn unknown attacks in less time. There are few works that compare the performance of DL and ML for IDS [19][20]. However, to the best of our knowledge, none of them compared the performance of DL and ML for BOT-IoT dataset. In this paper we compare the performance of machine learning technique with deep learning on Apache Spark for a better understanding of BOT-IoT data set for a transformed IDS.

3. System Architecture

This section presents the system architecture of our system. There are 6 components in our system namely Load the file, Data Preprocesing, Exploratory Data Analysis (EDA), Feature selection, Model implementation (Machine Learning, Deep Learning), Accuracy and Training time comparison in Fig. 1.

3.1. Load Dataset

This component enables us loading the dataset from CSV file. There are four different CSV files with 10 lakh instances each, consisting of different kinds of IoT network attacks including normal traffic flow. These four different files are merged together to make the entire dataset with 3.2 million instances all together [21].

3.2. Data Preprocessing

This component describes the cleaning of the dataset.isNotNull( ) function is used to identify rows with Null values on PySpark Dataframe and na.drop("any") function is used to eliminate nan values in the dataframe. Accordingto paper [1], there are 10 best features described by the author. These features describe the statistics of the dataset ina more efficient way. Therefore, for choosing the 10 best features, we have removed other Features and choose targetcolumn for analysis of the result.

3.3. Vector Assembly

One hot encoding is done on the all the features of the data set. Numerical features are converted into vectors through Vector Indexer and the target column is transformed through String Indexer to vector feature. Class.pyspark.ml.feature.VectorSizeHint(*, inputCol: Optional[str] = None, size: Optional[int] = None, handleInvalid: str = 'error') function is used to convert target column into vector. All the vector features excluding the target column are combined together in a single Feature column and merged with the target column. This process is followed for 10 best features and all the features are merged into single feature index and target column into label index.

3.4. AI Models

We build machine learning and deep learning models for classifying botnet attack flows from non-malicious flows. We employed Random Forest and Decision Tree machine learning algorithms for building machine learning models. The decision method does not use backtracking and instead uses a top-down greedy search technique with the distance of possible branches to generate decision trees. As the call implies, a tree algorithm will normally select the choice that appears to be fine at the time. whereas random forest has a low correlation, uncorrelated fashions can produce ensemble predictions which are more correct than any of the individual predictions. Moreover, it is observed in the literature that decision tree and random forest algorithms have provided better results among other machine learning algorithms for Intrusion detection problem.

Sequential Neural Network and Feed Forward Neural network are utilized to implement deep learning models. MLlib library is used to extract multiclass classification models. This paper has used Elephas estimator which works

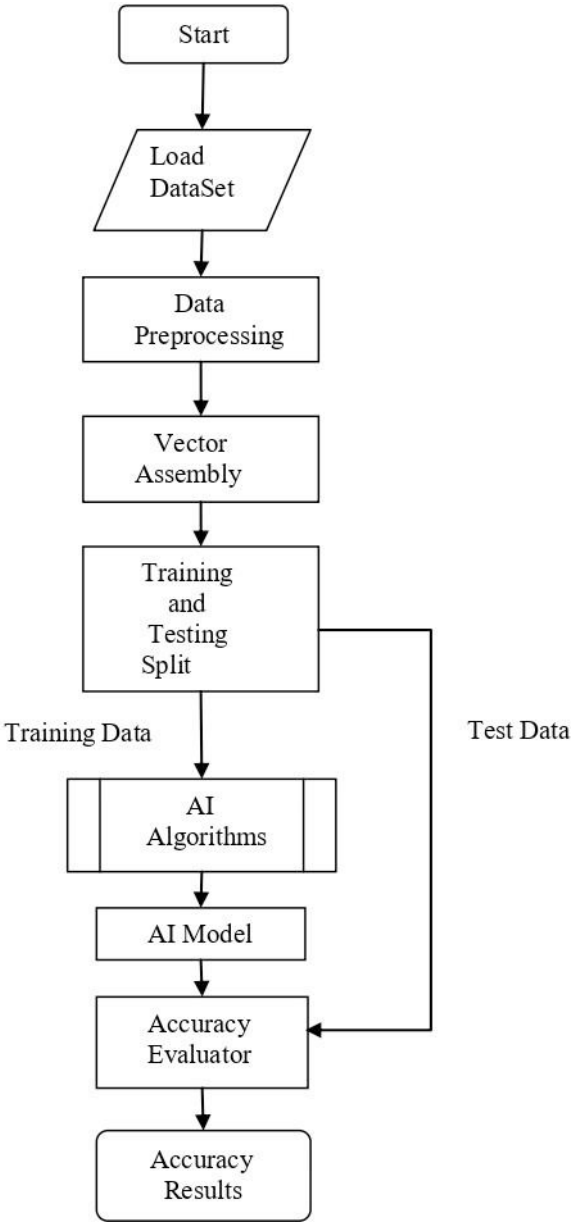

Fig. 1. System Architecture

with Keras TensorFlow library. After the DL model is constructed, a pipeline is created to run the model through Elephas estimator. Deep learning performance is compared with the same parameters used to compare the machine learning algorithms.

3.5. Accuracy and Training time

To evaluate our machine learning and deep learning models, we have collected the time to fit our training dataset into the model. This learning process is very important as this depends on how much our model is complex to learn new thing in efficient time.

4. Implementation

This section describes the system environment, the dataset and the libraries that are used to implement this system

4.1 System Environment

This work is done in a distributed architecture on Windows 10 Intel i5 10th generation 2 Socket 8 Core system with 8 GB of RAM. The work is done using Python on Apache Spark. This work is done with PySpark to access other libraries in python. Mllib, a machine learning library, is used. Platform and libraries which are used to achieve goal in this paper are discussed below:

4.1.1. Anaconda Prompt

It is a shell that runs on the command line (a program in which you type commands instead of using a mouse). The Anaconda Prompt, which consists of a dark screen and text, does not appear to be much, yet it is quite useful for Python problem solvers.

4.1.2. Pandas

Pandas is a data analysis software written in Python. Pandas is built on two important Python libraries: NumPy for mathematical computations and matplotlib for data visualization Pandas wraps these libraries, letting you use numerous matplotlib and NumPy techniques with fewer lines of code [22].

4.1.3. TensorFlow

TensorFlow offers the power and manipulate one want to construct complicated topologies with gear just like the Keras useful API and version Subclassing API [23]. Rapid prototyping and debugging benefit from eager execution.

4.1.4. MLlib

MlLib is one the most widely utlisied open-source library used for Machine Learning algorithms in Spark. Its aim is to make statistical system scalable and easy. At a high degree, it has a vast variety of ML Algorithms, commonly supervised and unsupervised algorithms along with category of regression, clustering, and collaborative filtering [24][25].

4.1.5. Elephas

Elephas brings deep studying with Keras to Spark. Elephas intends to hold the simplicity and excessive us- ability of Keras, thereby permitting for immediate prototyping of dispensed fashions, which may be run on large information sets. Elephas implements a category of data-parallel algorithms on pinnacle of Keras, using Spark's RDDs and information frames. Keras are initialized on the motive force, then serialized and shipped, alongside with records and broadcasted model parameters [26]. Spark deserialize the version, teach their chunk of facts and ship their gradients again to the driver. The "grasp" model at the motive force is updated by an optimizer, which takes gradients either synchronously or asynchronously.

4.2. Data Set

IoT networks are experiencing a new era of big data. Data set used in this paper for training and testing incorporate new types attacks with the antique varieties of traditional attacks, as many are being seen before for intrusion on IoT networks [27]. BoT-IoT dataset is created by designing a realistic network environment within the Cyber range Lab of the center of united states Canberra Cyber. The environment contains an aggregate of normal and botnet traffic. The dataset's supply files are provided in one-of-a-kind codes, together with the unique Packet Capture (PCAP) files, the generated ARGUS documents and CSV files. The files are separated based on attack class and subcategory,to help in labeling method. The captured pcap files are 69.3 GB in size, with more than 72,000,000 records. The extracted flow visitors, in csv format is 16.7 GB in size. The dataset includes DDoS, DoS, OS and service scan, Keylogging and theft attacks, with the DDoS and DoS assaults in addition organized, primarily based on the protocolused. To explore the opportunity for BOT-IoT data set, 5% of the authentic dataset is selected using select query from

pkSeqID	Row Identifier
Stime	Record start time
flgs	Flow state flags seen in transactions
flgs_number	Numerical representation of feature flags
Proto	Textual representation of transaction protocols present in network flow
proto_number	Numerical representation of feature proto
saddr	Source IP address
sport	Source port number
daddr	Destination IP address
dport	Destination port number
pkts	Total count of packets in transaction
bytes	Totan number of bytes in transaction
state	Transaction state
state_number	Numerical representation of feature state
ltime	Record last time
seq	Argus sequence number
dur	Record total duration
mean	Average duration of aggregated records
stddev	Standard deviation of aggregated records
sum	Total duration of aggregated records
min	Minimum duration of aggregated records
max	Maximum duration of aggregated records
spkts	Source-to-destination packet count
dpkts	Destination-to-source packet count
sbytes	Source-to-destination byte count
dbytes	Destination-to-source byte count
rate	Total packets per second in transaction
srate	Source-to-destination packets per second
drate	Destination-to-source packets per second
attack	Class label: 0 for Normal traffic, 1 for Attack Traffic
category	Traffic category
subcategory	Traffic subcategory

Fig. 2. List of Features in the dataset and their description

MySQL. The extracted 5%, is comprised of four documents of approximately 1.07 GB total size, and approximately 3.2 million records [21]. The records have also categorized as assault and non- assault, and the assault segment has been categorized similarly into different classes and subclasses to make the records usable for studying multiclass label problems.

For the experiment we have taken 5% of the entire dataset with all the 42 features and with 10 best features. This data set consist of 3.2 million instances including only 477 normal traffic flow. For binary classification this data set is highly imbalance and therefore this paper focus on multiclass category classification with 5 different types of multiclass category. The dataset includes five different types of categories described in Table 1 and attributes as shown in Fig. 2.

Table 1. Number of instances in the dataset.

Category	Instances
DDoS	1,514,315
DoS	1,302,148
RECONNAISSANCES	72,919
NORMAL TRAFFIC	477
THEFT	65

4.3. Data Process and Model Implementation

The data set is distributed into two patterns, set 1 with all the 42 features and set 2 with only 10 best features. Both the set are further divided into 70% training set and 30% testing set. To run the model, we mugged all the columns with feature column using vector assembly and target column as label for the feature. This paper has used feed forward neural network and sequential neural network which work as a deep neural network by adding up layers on top of each layer. In machine learning algorithm we have compared the performance of decision tree algorithm and random forest algorithm in terms of accuracy and training time of the model.

4.4. Implementation with all features

Forty-two features in the data set are merged together with vector assembly to form a feature column. For every feature column there is a feature index which is converted from the target column through string indexer. To build the model, we have input layers with 42 neurons and output layer with 5 neurons as five different categories ofattack network traffic flow. In the model, we had four hidden layers with 80,120,80 and 10 neurons in each layer.We have used multilayer perceptron classifier with iterations for feed forward neural network. In sequential model, we have used soft marks as the activation function in the hidden layers and Adams optimizer. To run the model,we have used 128 block size and one epoch to evaluate with our training data set. Machine learning algorithm are less complex compared to deep learning algorithms. After converting the data frame into features and label index we applied random forest and decision tree algorithm on our training set. We have chosen maximum number of trees and maximum number of depths as parameters for the algorithm. Machine learning algorithm comparatively took lesstime for training the data set. We evaluated the result of machine learning algorithm with confusion matrix.

4.5. Implementation with 10 best features

Thus, from the literature we find that there are 10 best features of BOT-IoT data set which describes the statistics for attack network flow prediction. We choose 10 main feature and applied deep learning algorithm and machine learning algorithms of this data set. Similarly, we compiled a data frame by combining the 10 best features in a single feature column followed by merging it with target column. In feed forward neural network, we have 10 neurons in the input layer corresponding to 4 hidden layers with 18, 36, 102 and 10 neurons on each hidden layer. We have our output layer with 5 neurons for five different classes. Our model has been executed with 128 block size and 100 iterations with learning rate 0.001 and 0.003. Likewise, for sequential neural network we have used same number of neurons per hidden layer. SoftMax has been used as activation function in between every layer and Adams optimizer with learning rate of 0.001 and 0.003.

To run the model, we used Elephas estimator with batch size 32 and 1 epoch to run on the training set. Coming to the machine learning algorithm implementation with 10 best features, we have used cross validation with different parameters. we performed number of trees starting from 10 to 15 and max depth from 5 to 10 with 10 folds for validation in random forest. When applying cross validation with decision tree we used max depth as 2, 5, 10, 20 and max bin as 10, 20,48, 80 with 10-fold for validation.

5. Results

From experiments with BOT-IoT 10 best feature and BOT-IoT all features, we found that using big data technology like Apache Spark there is an improvement in training time and accuracy of the algorithm. Using Spark implementation with machine learning, decision tree provided better accuracy with respect to random forest classifier. Decision tree took less training time to train the model. While implementing deep neural network we found that sequential neural network gave only 56.2% accuracy with 823.000 model loss while as feed forward neural network gave accuracy with 69.2% with 632.000 model loss on 10 best features in Table 2.

Table 2. Comparison of Accuracy and Training time.

Performance Summary	Accuracy (10 best features)	Training Time (SEC)	Accuracy(Full feature)	Training Time(SEC)
DECISION TREE	0.984	34.6	0.999	103
RANDOM FOREST	0.971	75.5	0.999	132
FFNN	0.693	1299	44.7	1870
SNN	0.562	1594.2	56.2	2006

In terms of training time feed forward neural network took less training time to train the model compared to sequential neural network. This gap between the training time of the dataset between machine learning algorithm and deep learning algorithm is due to model complexity. Deep learning model took 32,095 parameters to train with all

features and 12,634 parameters to train with 10 best features of the data set. Hence, deep learning models are more complex and denser compared to machine learning models.

6. Conclusion

In comparison to deep learning algorithms machine learning algorithm perform well in terms of accuracy and training time. Deep learning algorithm took more time to build the model and train the data. This work has proven that machine learning model took less time to learn the training dataset and gave satisfactory results. After performing multi class classification on big data platform Apache spark we found that decision tree out performed with all other models. Since we have worked in a constrained environment with distributed system in a single machine, a cluster of machines with a greater number of systems can be an appreciable future scope for this work as the training time of the model can be reduced and the performance can be increased.

References

[1] N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, **100**: 779-796

[2] Gupta, Reetu, and Rahul Gupta. "ABC of Internet of Things: Advancements, benefits, challenges, enablers and facilities of IoT." In 2016 Symposium on Colossal Data Analysis and Networking (CDAN), pp. 1-5. IEEE, 2016.

[3] https://www.businesswire.com/news/home/20190618005012/en/The-Growth-in-Connected-IoT-Devices-is-Expected-to-Generate-79.4ZB-of-Data-in-2025-According-to-a-New-IDC-Forecast

[4] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas and Y. Zhou (2017), "Understanding the mirai botnet",*SEC'17 Proceedings of the 26th USENIX Conference on Security Symposium*.

[5] A. Abuzuraiq, M. Alkasassbeh and A. Mohammad,(2020) "Intelligent methods for accurately detecting phishing websites", *11th International Conference on Information and Communication Systems (ICICS)*.

[6] D. Denning (1987), "An Intrusion-Detection Model" ,*IEEE Transactions on Software Engineering*, **13**,(2): 222-232.

[7] M.Almseidin, M. Alzubi, S. Kovacs and M. Alkasassbeh,(2017) "Evaluation of machine learning algorithms for intrusion detection system," *IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*.

[8] Garg, U.; Kaushik, V.; Panwar, A.; Gupta, N.(2021) ,"Analysis of Machine Learning Algorithms for IoT Botnet",*In Proceedings of the 2nd International Conference for Emerging Technology (INCET)*:1–4.

[9] Alazzam, H.; Alsmady, A.; Shorman, (2019)."A Supervised Detection of IoT Botnet Attacks.",*In Proceedings of the Second International Conference on Data Science*.

[10] Bhatia, R.; Benno, S.; Esteban, J.; Lakshman, T.; Grogan, J,(2019)," Unsupervised machine learning for network-centric anomaly detection in IoT". *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*: 42–48.

[11] C. Tzagkarakis, N. Petroulakis, and S. Ioannidis,(2019), "Botnet attack detection at the iot edge based on sparse representation," in 2019 Global IoT Summit (GIoTS)*IEEE*: 1–6.

[12] Sriram, R. Vinayakumar, M. Alazab and S. KP,(2020) "Network Flow based IoT Botnet Attack Detection using Deep Learning",*IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*: 189-194

[13] G. P. Gupta and M. Kulariya,(2016) "A Framework for Fast and Efficient Cyber Security Network Intrusion Detection Using Apache Spark",*Procedia Computer Science*,**93**:824-831.

[14] K. Kato and V. Klyuev,(2017) "Development of a network intrusion detection system using Apache Hadoop and Spark",Conference on Dependable and Secure Computing.

[15]] G.Donkal and G. K. Verma,(2018) "A multimodal fusion based framework to reinforce IDS for securing Big Data environment using Spark",*Journal of Information Security and Applications*,**43**: 1-11.

[16] K. Vimalkumar and N. Radhika,(2017), "A big data framework for intrusion detection in smart grids using apache spark", *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman,(2019) "Deep Learning Approach for Intelligent Intrusion Detection System",*IEEE Access*,**7**: pp. 41525-41550.

[18] V.Morfino and S. Rampone,(2020), "Towards Near-Real-Time Intrusion Detection for IoT Devices using Supervised Learning and Apache Spark",*Electronics*, **9**,(3): 444.

[19] Thapa, Niraj, Zhipeng Liu, Dukka B. Kc, Balakrishna Gokaraju, and Kaushik Roy.(2020), "Comparison of machine learning and deep learning models for network intrusion detection systems." *Future Internet*,**12**,(10): 167.

[20] Abdel-Wahab, Mohab Sameh, Ahmed M. Neil, and Ayman Atia,(2020), "A Comparative Study of Machine Learning and Deep Learning in Network Anomaly-Based Intrusion Detection Systems." *15th International Conference on Computer Engineering and Systems (ICCES)*: 1-6.

[21] Nour Moustafa,(2019), "The Bot-IoT dataset", *IEEE Dataport.*

[22] Joshi, Chirag, Ranjeet Kumar Ranjan, and Vishal Bharti.,(2021) "A Fuzzy Logic based feature engineering approach for Botnet detection using ANN".*Journal of King Saud University-Computer and Information Sciences*.
[23] Sujadevi, V. G., K. P. Soman, R. Vinayakumar, and A. U. Prem Sankar,(2019), "Anomaly detection in phonocardiogram employing deep learning."*In Computational intelligence in data mining*:525-534.
[24] Chen, Liguo, Yuedong Zhang, Qi Zhao, Guanggang Geng, and ZhiWei Yan,(2018), "Detection of dnsddos attacks with random forest algorithm on spark."*Procedia computer science*,**134**: 310-315.
[25] Parambalath, Gokul, E. Mahesh, P. Balasubramanian, and P. N. Kumar,(2019). "Big data analytics: a trading strategy of nse stocks using bollinger bands analysis."*In Data Management, Analytics and Innovation*: 143-154.
[26] Priovolos, A., G. Gardikis, D. Lioprasitis, and S. Costicoglou."Improving Apache Spot Using Autoencoders for Network Anomaly Detection."
[27] Krishnan, Prabhakar, Kurunandan Jain, Rajkumar Buyya, Pandi Vijayakumar, Anand Nayyar, Muhammad Bilal, and Houbing Song,(2021) "MUD-based behavioral profiling security framework for software-defined IoT networks."*IEEE Internet of Things Journal*,**9**,(9): 6611-6622