International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 419-431

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Identification of Best Machine Learning Algorithms for Detection of HTTP Botnet Attack

[1]Morufu Olalere, [2]Yisa Rhoda Nnaba, [3]Ojeniyi Joseph A. [4]Vivian O.Nwaocha

[1, 2, 3] Department of Cyber Security Science, Federal University of Technology Minna, Nigeria
[4] Computer Science Departments, National Open University of Nigeria
Email: lerejide@futminna.edu.ng; danielrhodatsado@gmail.com; ojeniyija@futminna.edu.ng; onwaocha@noun.edu.ng

Abstract

A botnet is an internet connected compromised devices each of which is running one or more bots. Bots are software applications that perform automated tasks (scripts) over the Internet without any human interference to start them up, typically they are capable of running iterative tasks, and they can do that faster than human users could. The botmaster commands a volume of comprised computers (called bots) to carry out illegal tasks.. In this paper, a comparative study has been carried out on fifteen machine learning Algorithm-Naïve Bayes, K-Nearest Neighbor(KNN), Random Forest Decision stump, k-star(K), Decision table, Adaboost, Bagging, stacking, ZeroR, Local Weight Learning, Kernel Logistic Regression, CSforest, BFTree and BayesNet were evaluated to know which algorithm is most accurate for prediction on http Botnet detection.. Random Forest was found to be an excellent classifier that gives the best accuracy of 0.999 and a false positive rate of 0.001.*

Keywords: *HTTP Botnet, machine learning, Classifiers, Random Forest, Botmaster*

1. Introduction

This Botnet is crafted from robot and network; a botnet is an internet connected compromised devices each of which is running one or more bots. Bots are software applications that perform automated tasks (scripts) over the Internet without any human interference to start them up, typically they are capable of running repetitive tasks, and they can do that faster than human users could. The botmaster commands a volume of comprised computers (called bots) to carry out illegal tasks such as spreading spam, stealing user sensitive information. Distributed Denial of Service depriving the legitimate users of the networked services Faddhlee et al (2018). The entire network of particular organization can also be attacked; these attacks may be physical or logical Olalere et al (2018). This machines program called Botnet breaches the different network protocol like internet relay chat (IRC), Hypertext Transfer Protocol (HTTP) and even a peer – to - peer (P2P) network. Botnet network can be command and control; the structure can be centralized and also decentralized. P2P botnet is a decentralized structure, every bot behaves as a client and as well as a server Medemott et al (2014) and Selvam et al (2015).

HTTP botnet is a centralized command and control (C & C) mechanism. The HTTP mechanism requires the infected client to send an HTTP get request to a command and control website. Infected host use "pull" mechanism and can only receive commands after they have sent a request to the C & C site Tyagi et al (2015). HTTP botnet is easy to deploy and difficult in detection as they can hide their activities

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

in mist of large number of normal web, traffic and higher availability as the HTTP is widely used protocol on the Internet Eslahi et al (2015).

HTTP Botnet attacks have proven to be a long time challenge to the Internet of Things services. Various techniques have been devised to detect http Botnet attack. Also most of the techniques such as the techniques in (Muhammad& Chin, 2016) could only exploit the weakness of Botnet system revealing its presence, (Eslahi*et al.,* 2012) was able to efficiently label the periodic activities of Botnet up to 80% and (Hoang & Nguyen2018) could only detect the presence of a Botnet attack with overall accuracy of less than 90% and false positive rate of 17.10%.

In this vain, a comparative study of some selected machine learning based algorithm is experimented to identify the best technique for detecting http Botnet attacks in Internet of Things environment to achieve a more reasonable accuracy with lesser false positive rate.

Summary of the key contributions of this study are;

- To carry out an experiment on Botnet dataset using some selected machine learning classifiers.
- To evaluate the performance metrics of the machine learning algorithm
- To carry out a comparative analysis of machine learning classification algorithms

The aim of the research work is to comparatively analyze fifteen selected supervised machine learning algorithms to verify the most accurate machine learning algorithm with lesser training time for prediction of http botnet on IoT environment using a generated Botnet dataset.

The remainder of this paper is organized as follows: section II provides an overview of existing http botnet detection techniques. Section III introduces the dataset used to rain and test the various algorithms used for http botnet detection. Section IV collects the experiments performed and explains the results obtained. Finally, section V is our conclusion and future work.

2. Related Works

There are so many machine learning approaches of detecting malicious traffics generated by botnets. However, some related literatures will be discussed in this section

Nurhidayah et al (2018) used machine learning to detect HTTP botnet by using tcp dump data. KNN classifier is able to detect HTTP botnet with high detection accuracy, the random forest classifier recorded high false positive rate which produce high false alarm during the HTTP botnet detection. The weakness of this work is that the value of the results were not stated.

Naseri (2017) proposed correlation-based technique was used for HTTP botnet detection, C4.5, random forest, decision tree, Naïve Bayes, SVM, feed forward Neural Network classifiers were used together with network communication histogram Analysis. The feed forward Neural Network is able to distinguish the activities of botnet with a significant rate of accuracy and very low false positive rate. The weakness of this technique is that all other classifiers have high false positive and Naïve Bayes has least true positive rate among others.

Mathew et al (2014) proposed genetic algorithm to detect http botnet and defend the layer against http botnet. The method results in efficient detection which lowers false positive rate; however, the accuracy of this system was not measured.

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Medemott et al (2018) Proposed Deep learning approaches (BLSTM – RNN) to detect http botnet in the IoT using a Mirai Botnet Datasets. The approach was able to achieve high accuracy but is not capable of detecting other types of botnet attack since Mirai botnet is the only botnet dataset use for the approach.

Eslahi (2012) proposed a periodicity classification technique to Detect HTTP Botnet the technique used decision tree algorithms with Zeus, Droid kungfu Hot1 dataset. It was capable of efficiently label the periodic activities of Botnet up to 80% accuracy. This technique was only efficient on a small scale botnet (e.g. single bot) this is a great weakness.
Narang & Hota (2015) used Bayesian Networking (BayeNet), J48 Decision Tree and Ada boost with REP tree for detection of P2P botnet attack on application layer. They got the total accuracy of 96%, 97% and 95% respectively. But their method only detect P2P botnet.
Nurhidajah et al (2018) used Logistic Regression for threshold identification for http botnet detection. The authors achieved 95% accuracy, but this approach cannot detect botnet at secondary infection.
Kirubatti & Shama (2016) Used Adaboost, Naïve Bayes and SVM classifiers for the detection of Botnet attack. The author achieved high accuracy of 95.86%, 99.14% and 92.02% respectively.

Theoretical Review

2.1.1 HTTP Botnets

HTTP botnet is the most contemporary botnet; it uses port 80 to exchange web requests. This botnet uses the internet with an HTTP message to set up its communication with certain URL Fadhlee et al (2018). This HTTP message contains unique identifiers for the bots. The server under consideration will reply to these HTTP messages with further investigation commands.

Using the HTTP protocol to build botnet C & C approach, the following are the advantages.

- HTTP C & C use the client to server model. It can be easily constructed as compared to P2P botnet Cai & Zou (2012).
- Using the HTTP protocol to construct a botnet control channel can make the C & C flow sink in a voluminous amount of internet web communication since HTTP is a common widely – used protocol.
- HTTP botnets have a more flexible environment compared with other types of botnets, because HTTP service is a popular service so firewall sporadically blocks the HTTP service.

The traffic of the HTTP botnets flows with regular traffic; the bot packets are different from normal packets causing the detection procedure to be easy Lu et al (2015) and Fadhlee et al (2018)

2.1.2 HTTP Botnet Detection

HTTP Botnet detection faces the following complications:

- It is very difficult to track HTTP botnet, because its communication protocol is similar to the normal flows. Such Botnet usually gets commands by constructing legitimate HTTP request packages through HTTP protocol.
- Since the number of botnet in a large network is small, the amount of traffic generated by malicious communications is too small which makes it difficult to picture out C & C from the giant traffic Cai & Zou (2012) and Liu et al (2014).

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

2.1.3 Machine Learning

Machine Learning is a branch of science that deals with the design of computer programs and systems that can study rules from data, adapt to changes, and improve accuracy or performance with experience. Now, with these study rules we can obtain many algorithms, such as Random Forest, K- Nearest Neighbor, naive Bayes and K*, that are used to classify a data set from real problems Yoan et al (2016).

a) Random Forest Algorithm

Random forest is one of the strongest algorithms used for predictive modeling Mathew et al (2014). The fundamental principle is the building of numerous decision trees by the combination of random variables. That is, many decision trees are built from the given data set and the results are combined to make predictions. In order to obtain the highest accuracy in prediction, the best variables are combined. Increasing the number of trees generated can adjust the accuracy of the Random Forest algorithm. Each decision tree generated makes its own prediction, some may be right and some wrong. The particular trees that produced correct predictions strengthen each other, while wrong predictions get scraped. For this to happen, the individual trees generated must be uncorrelated. Here comes the Bagging technique which helps in generating the decision trees with minimal correlation Nurhidayah et al (2018).

b) The Naïve Bayes Algorithm

The Naives Bayes classifier algorithm which is based on Bayes's rule, which express two criteria, the probability of an event before evidence is seen, and the probability of an event after evidence is seen. This classifier is used for many reasons, for example, it is simplest to build, not needing any complex iterative parameter evaluation schemes Yoan et al (2016). It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle. The classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable Olalere etal (2018).

c) K-nearest neighbor Algorithm

The K- Nearest Neighbor (kNN) algorithm is the basic instance based learners that use several domain particular function of distance to recover the only the most analogous case from the training set. The classification of the recovered case is given as the classification for the new case Yoan et al (2016).

d) Decision stump Algorithm

A decision stump is a machine learning model consisting of a one-level decision tree. That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature Yoan et al (2016).

e) K algorithm*

The K* algorithm is a classifier that uses entropy as a distance measure, as a stronger approximation to handling symbolic, real-valued attributes and missing values. This algorithm is an instance-based learner which uses such an evaluation Yoan et al (2016).

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

f) Adaboost M1

One of the widely used for algorithm is Adaboost M1, it is used for constructing a strong classifier as a linear combination of weak classifier. This technique combines multiple classifiers by assigning weights to each of them according to their classification performance Kirubatti & Shama (2016).

g) ZEROR Classifier

ZeroR Classifier is the classification method which is simplest of the all. It disregards all the predictors and only relies on the target Bal & Shama (2016).

h) META-BAGGING Classifier

Meta-Bagging is also known as only Bagging also. Bagging is defined as bootstrap aggregation. Bagging produces bootstrap samples of the training data. It constructs the unique training set consists of numerous data sets. Numerous data sets are constructed by unsystematic sampling occurrences with replacement. Each single bootstrap sample is used for training a regression function or a classifier. Classification results are taken on maximum number of votes for classification purpose. For regression average of expected values are taken Bal & Shama (2016).

i) Logistic Regression

This refers to various regressions with an outcome variable that is a categorical variable while the predictor variables are continuous or categorical. Besides, logistic regression is convenient as any value from negative infinity to positive infinity, it can take as an input, whereas the values between 0 and 1 as an output Nurhidayah et al (2018).

j) Decision Table algorithm

It summarizes the dataset with a decision table which contains the same number of attributes as the original dataset. Then, a new data item is assigned a category by finding the line in the decision table that matches the nonclass values of the data item. Decision Table employs the wrapper method to find a good subset of attributes for inclusion in the table. By eliminating attributes that contribute little or nothing to a model of the dataset, the algorithm reduces the likelihood of over-fitting and creates a smaller and condensed decision table Mahajan & Ganpati (2014).

3. Methodology

Finding appropriate Botnet dataset for machine learning is always challenging. So an IoT botnet dataset was generated by Australian center for cyber security with proper labeled target features. The dataset is made of TPC, UDP and HTTP protocol. The Network platforms include normal and attacking virtual machines (VMs) with additional network devices such as firewall and tap. In all, 32 features which are related to different packet were generated. Kali Lixus Virtual Machines (VMs) were used to launch cyber-attacks in parallel. The various botnet attacks implemented as DDoS, DoS, information theft (key logging/Data theft) information gathering (OS fingerprinting/port scanning). The dataset is split into train and test randomly in 70-30 ratio for http botnet detection, the dataset has 6001 instances. The feature selection process was done by dropping the features with least importance. Therefore, to avoid overfitting during the model training 14 features were dropped 18 was selected for the experiment. For the purpose of this study only features related to HTTP protocol were extracted.

The performance of the classifier evaluates using the false positive rate, false negative, rate overall accuracy (OA), F-measure, error rate, MCC, Recall which are defined as follows:

Accuracy: Accuracy is a term that describes the average value of True Positive Rate (TPR) and True Negative Rate (TNR). It is denoted as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Error Rate (ER): This refer to as the percentage of wrong reading of false positive rate and false negative rate. It is denoted as:

$$Error\,Rate = \frac{FP + FN}{TP + FP + TN + FN} \quad (2)$$

F-measure: This is a statistical index that shows the performance and efficiency of the model, and
checks for the potential imbalance problems. It is denoted as:

$$\text{F –measure} = \frac{2TP}{2TP + FP + FN} \quad (3)$$

False Positive Rate: This can also be called the false alarm rate, it is the percentage of the test set that the model predicts falsely as positive when it was actually negative. It is denoted as:

$$FPR = \frac{FP}{FP + TN} \quad (4)$$

False Negative Rate: This can be referred to the percentage of the test set that the model predicts falsely as negative when it is actually positive. It is denoted as:

$$FNR = \frac{FN}{TR + FN} \quad (5)$$

Recall: Recall is the percentage of the test set that the model predicts; it is also referred to as True Positive Rate (TPR). It is denoted as:

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

Mathew's Correlation Coefficient: This is defined as ration between the observed and predicted binary classifications as presented in equation 7. That is:-

$$Mathew's\ Correlation\ Coefficient\ (MCC) = \frac{TN \times TP - FN \times FP}{\sqrt{(FP + TP)(FN + TP)(TN + TP)(TN + FN)}} \quad (7)$$

4. Results and Discussion

The experiment was carried out on Weka tool. First, The author carry out feature selection on the 32 attributes from the Botnet generated dataset, to know attributes that have high impacts and those without

International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 419-431
Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

impact on our prediction a correlation analysis was run on the attributes using Microsoft excel. Attributes with no correlation were extracted as shown on table 1.

Table 1. Selected Features and Descriptions

Feature	Descriptions
PkSeqID	Row identifier
Proto-num	Numerical represent feature
Saddr	Source IP address
Sport	Source port number
Dport	Destination port number
bytes	Total no. of bytes in transaction
state	Transaction state
dur	Record total duration
mean	Average time total records
srate	Source to destination pkts/sec
dbytes	Destination-source byte count
sum	Total duration of records
max	Maximum duration of records
min	Minimum duration of records
dport	Destination port number
dpkts	Destination-source pkt count
Attack	0 for Normal traffic, 1 for Attack Traffic

After the feature selection the experiment was done with selected algorithm, the result on table. II was obtained

Table 2. Results Obtained

Classifiers	FP	RECALL	F-MEASURE	ERROR RATE	MCC	ACCURACY (%)	TIME TAKEN(Secs)
Naïve Bayes	0.13	0.935	0.933	0.065	0.855	0.935	0.11
K-nearest neighbor	0.009	0.996	0.996	0.0047	0.99	0.995	1.09
K-star	0.508	0.748	0.691	0.1261	0.418	0.748	4.11
Random forest	0.001	0.999	0.999	0.0048	0.999	0.999	0.25
Decision Stump	0.006	0.943	0.944	0.0992	0.874	0.943	0.01
Decision Table	0.001	0.999	0.999	0.0031	0.997	0.998	0.14
Adaboost	0.004	0.998	0.998	0.0043	0.995	0.997	0.03
Bagging	0.002	0.999	0.999	0.0033	0.999	0.998	0.07
Stacking	0.669	0.669	0	0.4444	0	0.668	0.02
ZeroR	0.669	0.669	0	0.4441	0	0.668	0.01
Local Weight Learning	0.045	0.977	0.977	0.0488	0.976	0.976	5.87
KernelLogisticRegression	0.004	0.998	0.998	0.002	0.995	0.997	3.16
Csforest	0.007	0.997	0.997	0.0574	0.992	0.996	0.15
BFTree	0.006	0.997	0.997	0.0029	0.994	0.997	0
BayesNet	0.012	0.993	0.993	0.0069	0.985	0.993	0.03

International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 419-431
Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

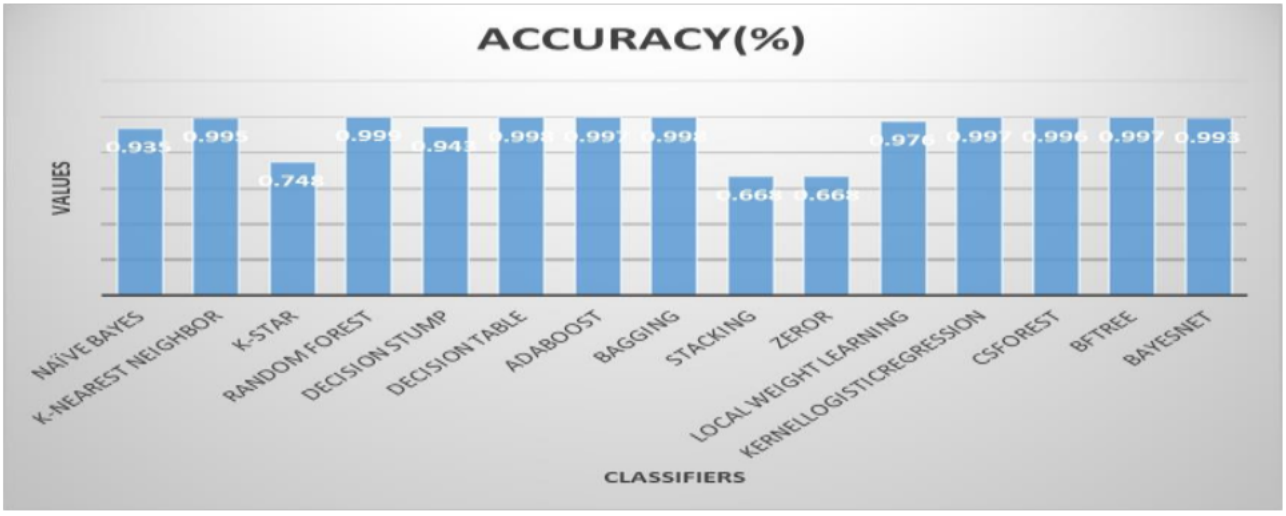

Figure 1. Accuracy

In figure 1, the performance of our algorithm in terms of accuracy in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest performs better with accuracy of 99.9% compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with accuracy 93.5%, 99.5%, 74.8%, 94.3%, 99.8%, 99.7%, 99.8%,66.8%, 66.8%, 97.6%, 99.7%,99.6%, 99.7%, and 99.3% respectively.

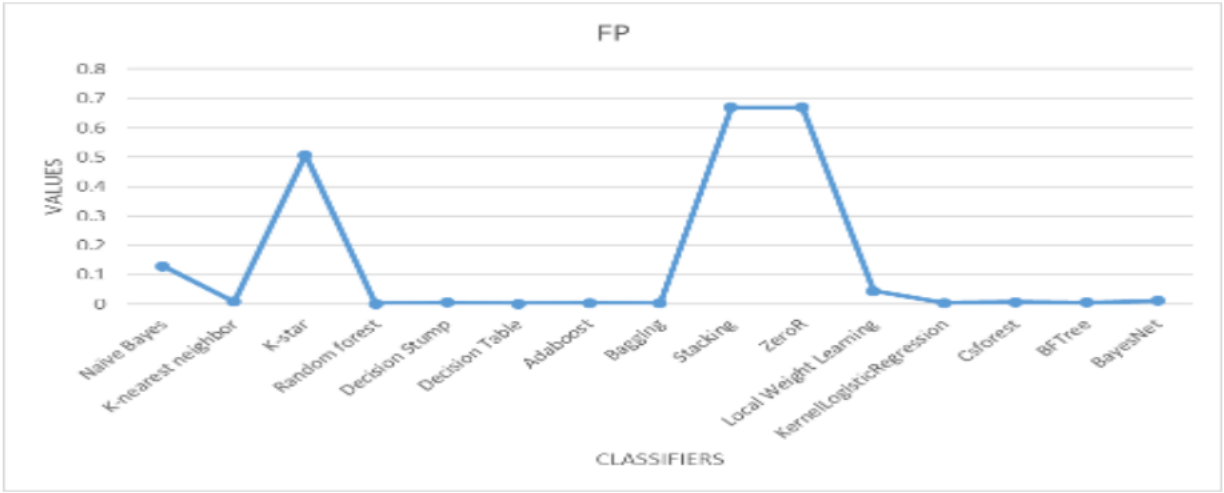

Figure 2. False positive

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Figure 2, the performance of our algorithm in terms of false positive rate in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest and Decision table have lesser false positive rate of 0.001 and 0.001 compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with false positive rate 0.13, 0.009, 0.508, 0.006, 0.004, 0.002, 0.669, 0669, 0.045, 0.004, 0.007, 0.006 and 0.993 respectively.

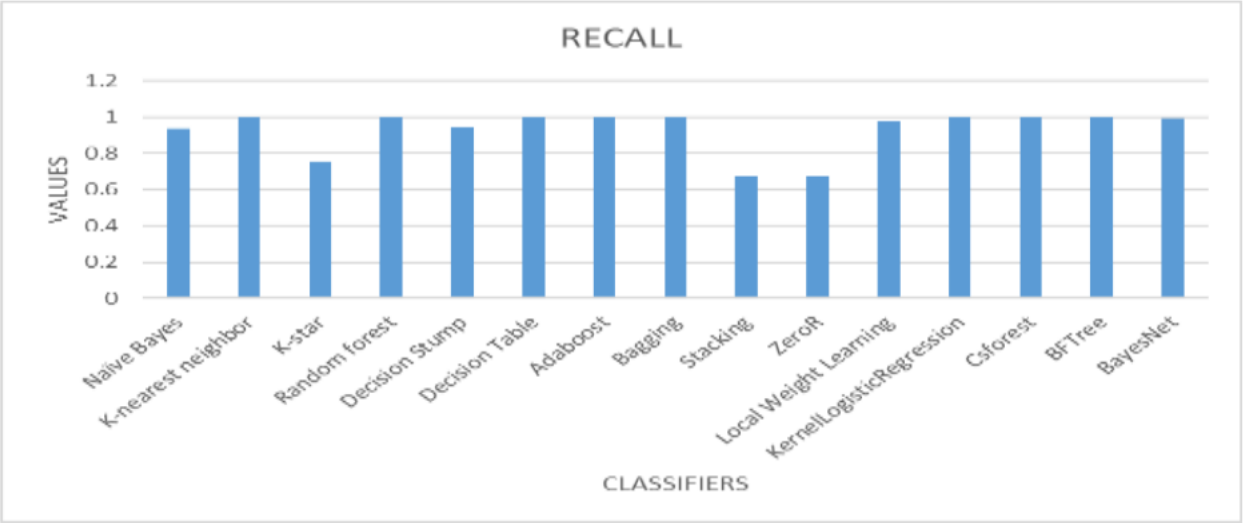

Figure 3. Recall

Fig.3, the performance of our algorithm in terms of recall in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest performs better with recall of 99.9% compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with recall 93.5%, 99.5%, 74.8%, 94.3%, 99.8%, 99.7%, 99.8%,66.8%, 66.8%, 97.6%, 99.7%,99.6%, 99.7%, and 99.3% respectively. Figure 4, the performance of our algorithm in terms of F-measure in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest performs better with F-measure of 99.9% compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with F-measure 93.3%, 99.6%, 69.1%, 94.3%, 99.8%, 99.7%, 99.8%, 0.00%, 0.00%, 97.7%, 99.8%,99.7%, 99.7%, and 99.3% respectively.

.

International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 419-431
Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

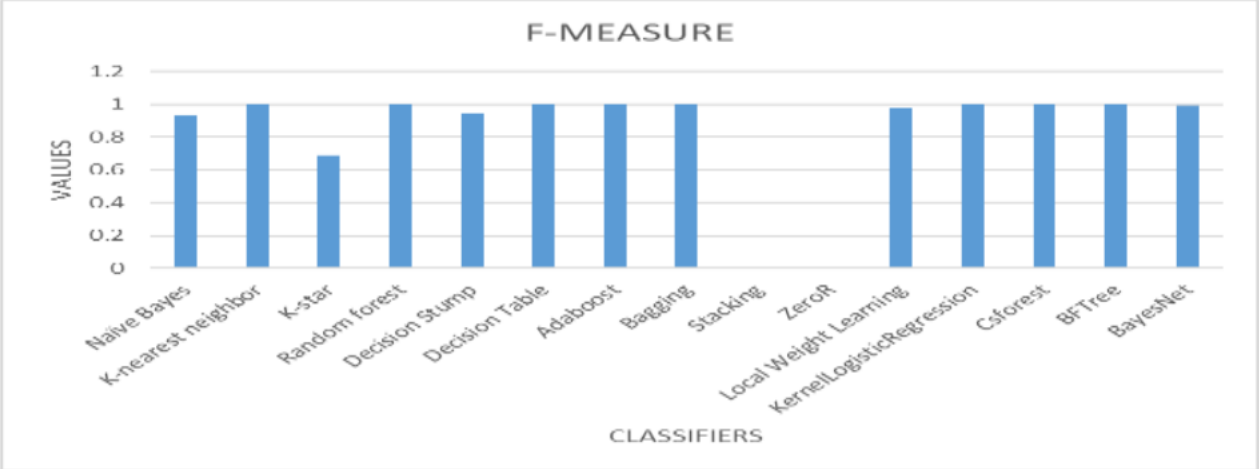

Figure 4. F-measure

Figure 5. Error rate

Figure 5, the performance of our algorithm in terms of error rate in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest have error rate of 0.0048 compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with error rate 0.069, 0.0047, 0.1261, 0.0992, 0.0031, 0.0043, 0.003, 0.4444, 0.4441, 0.0488, 0.002, 0.0574, 0.0029, and 0.0069% respectively.

International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 419-431
Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

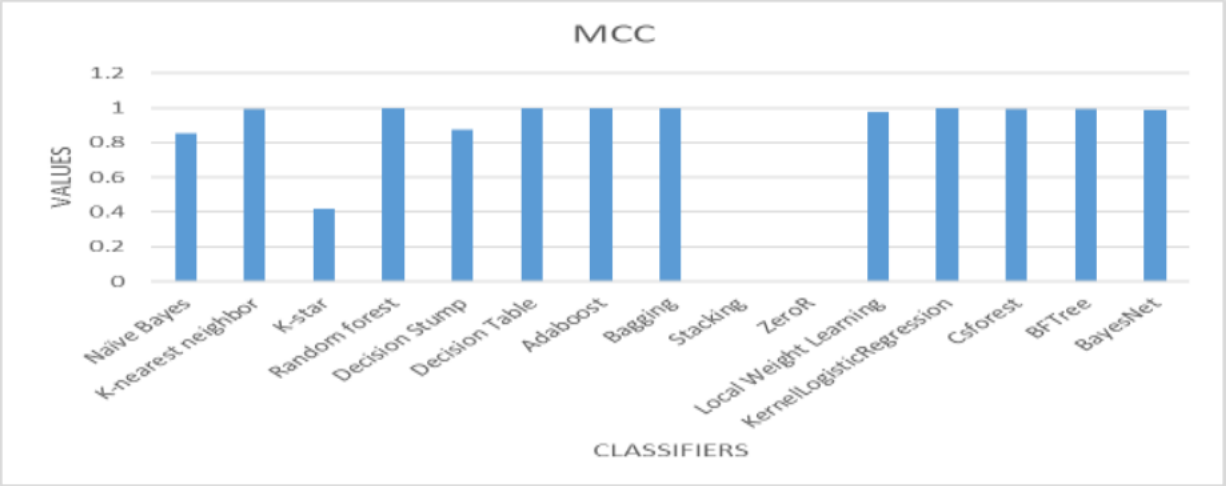

Figure 6. MCC

Figure 6, the performance of our algorithm in terms of MCC in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest performs better with MCC of 99.9% compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with MCC 85.5%, 99.0%, 41.8%, 87.4%, 99.7%, 99.5%, 99.8%, 0.00%, 0.00%, 97.6%, 99.5%, 99.2%, 99.4%, and 98.5% respectively.

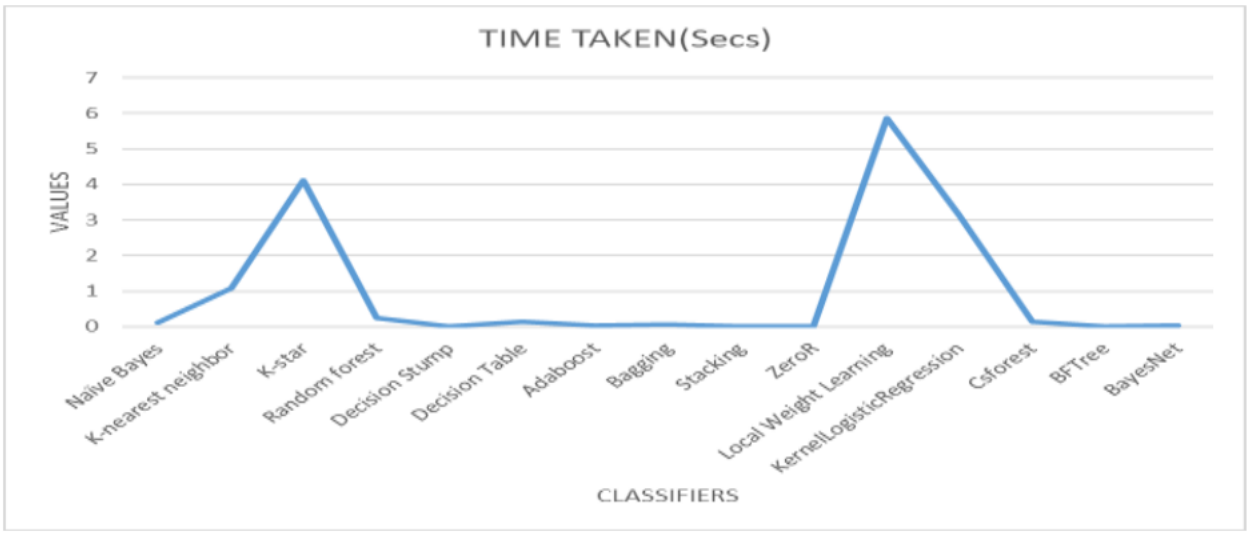

Figure 7. Time Taken

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Figure 7, the performance of our algorithm in terms of time taken in comparison with Naïve bayes, KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet. Random Forest have time taken of 0.25 secs compare with Naïve Bayes , KNN, K-star, Random forest, Decision stump, Decision table, Adboost, Bagging, Stacking, ZeroR, Local weight learning, Kernel logistic regression, Csforest BFtree and BayesNet with time taken 0.11secs, 1.09secs, 4.11secs, 0.1secs, 0.14sec, 0.3secs, 0.07secs, 0.02, 0.01secs, 5.87secs, 3.16secs, 0.15sec, 0.0sec, and 0.03secs% respectively.

5. Conclusion And Recommendation

In this paper, the author have evaluated and made comparisons of Fifteen different machine learning algorithm for the detection of HTTP botnet attack on IoT environment. Generated Botnet dataset was used for this experiment and all the features used are all HTTP protocol related. 70% of the dataset was used for the training of the models and 30% for testing each model. The results obtained clearly shows that random forest has the best performance than Meta Bagging, Decision Table, Adaboost M1, Kernel Logistic Regression, BF tree, CS Forest K-NN, BayesNet Local Weight Learning, Decision Stump, and Naïve Bayes, the worst performed classifiers are K*, Meta Stacking and ZeroR.

Random Forest classifier with highest accuracy of 99.9%, false positive rate of 0.001 and minimal training time of 0.25secs is rated as the best classifier. It is therefore recommended to be use as intrusion detection system in IoT environment to detect HTTP botnet attack. The authors encourage further experiment be performed on other protocols with different botnet dataset source using the same machine learning algorithms.

References

Bal, R., & Sharma, S. (2016). Review on Meta Classification Algorithms using WEKA, *35*(1), pp38–47,

Cai, T., & Zou, F. (2012). Detecting HTTP Botnet with Clustering Network Traffic, (1), 5–11.Patil, S. P. (2011). Botnet-A Network Threat, pp28–35

Eslahi, M., Rohmad, M. S., Nilsaz, H., Naseri, M. V., Tahir, N. M., & Hashim, H. (2015). Periodicity Classification of HTTP Traffic to Detect HTTP Botnets, (April). https://doi.org/10.1109/ISCAIE.2015.7298339

Eslahi, M. (2012). Bots and Botnets : An Overview of Characteristics , Detection and Challenges, (April 2014). https://doi.org/10.1109/ICCSCE.2012.6487169

Fadhlee, R., Dollah, M., Faizal, M. A., Arif, F., Zaki, M., & Xin, L. K. (2018). Machine Learning For HTTP Botnet Detection Using Classifier Algorithms. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1), pp 27–30

Fadhlee, R., Dollah, M., Faizal, M. A., Arif, F., Zaki, M., & Xin, L. K. (2018). Machine Learning For HTTP Botnet Detection Using Classifier Algorithms. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1), pp 27–30

Kirubathi & Anitha (2016) Chapter 5 botnet detection via mining of traffic flow. (n.d.), pp 96–132

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, 10(3), pp779–796. https://doi.org/10.1016/j.future.2019.05.041

Lu, T. T., Liao, H. Y., & Chen, M. F. (2011). an Advanced Hybrid P2P Botnet 2.0, *5*(9), pp273–276. https://doi.org/10.5220/0003504102730276

Mathew, S. E., Ali, A., & Stephen, J. (2014). Genetic Algorithm based Layered Detection and Defense of HTTP

Botnet. *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation GECCO 09*, *5*(1), pp 128–133. https://doi.org/10.1145/1569901.1570111

Mahajan, A., & Ganpati, A. (2014). Performance Evaluation of Rule Based Classification Algorithms, *3*(10), pp3546–3550.

Mcdermott, C. D., Majdani, F., & Petrovski, A. V. (2018). Botnet Detection in the Internet of Things using Deep Learning Approaches, (August). https://doi.org/10.1109/IJCNN.2018.8489489.

Narang, P., Ray, S., & Hota, C. (2015). PeerShark : Detecting Peer-to-Peer Botnets by Tracking Conversations.

Naseri, M. V. (2017). Correlation-Based HTTP Botnet Detection Using Network Communication Histogram Analysis, 7–12

Nur Hidayah. M. S, Faizal. M. A, Wan. A. R. W. Y &Rudy.F. M. D (2018). Threshold identification for http botnet detection. *Journal of Theoretical and Applied Information Technology*, *96*(14), pp 4428–4438

Olalere M., Egigogo R. A., Ojeniyi J. A.., Idris I., & Jimoh R. G. (2018) : A Naïve Bayes Based Pattern Recognition Model for Detection and Categorization of Structured Query Language Injection Attack. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)* 7(2): pp 189-199 Communications.

Olalere . M, Waziri V. O., Ismaila I., Adebayo O. S., Ugwu J. N. (2018) : Cyber Security Education: A Tool for National Security Attack. *International Journal of Data & Network Security*,4(1).

Selvam, N., Vanitha, V., & Sumathi, M. V. P. (2015). General-framework-for-detection-of-botnet-using-Random-forest-in-real-time.doc, *6*(4).

Liu, M., Liu, X., Li, J., Ding, C., & Jiang, J. (2014). Evaluating total inorganic nitrogen in coastal waters through fusion of multi- temporal RADARSAT-2 and optical imagery using random forest algorithm. *International Journal of Applied Earth Observation and Geoinformation*, *33*(9), pp192–202. https://doi.org/10.1016/j.jag.2014.05.009

Tyagi, R., Paul, T., & Manoj, B. S. (2015). A Novel HTTP Botnet Traffic Detection Method, 1–6.

Yoan Martínez-López , Julio Madera-Quintana , & Ireimis Leguen de Varona (2016). Study of the Performance of the K* Algorithm in International Databases. 12,(23), pp51-56,