Classifying Botnet Attack on Internet of Things Device Using Random Forest

Classifying Botnet Attack on Internet of Things Device Using Random Forest

Irfan[1], I M Wildani[1] and I N Yulita[1]

[1]Universitas Padjadjaran, Jalan Raya Bandung Sumedang KM 21, Hegarmanah, Jatinangor, Sumedang, West Java 45363

irfan15012@gmail.com
verone.one@gmail.com
intan.nurma@unpad.ac.id

Abstract. We live in Industry 4.0 where Internet of Things (IoT) is a new developing environment. A lot of researcher is trying to develop this new technology. As this technology is starting to become big, people try to attack the system of this technology. Luckily, a dataset contains of unattacked environment and attacked environment exist. The purpose of this research is to classify the incoming data in the IoT, contain a malware or not. In this research, we under sample the dataset because the datasets contain imbalance class. After that, we classify the sample using Random Forest. We use Naive Bayes, K-Nearest Neighbor and Decision Tree too as a comparison. The dataset that has been used in this research are from UCI Machine Learning Depository's Website. The dataset shows the data traffic from the IoT Device in a normal condition and attacked by Mirai or Bashlite. Random Forest gets greatest accuracy with 99.99% value with Precision, Recall, and F-Measure get 100% value. The score is followed by Decision Tree with 99.98% accuracy, KNN with 99.94% accuracy and Naive Bayes with 99.00% accuracy.
Keywords: Data Mining, Random Forest, IoT, Botnet

1 Introduction

In the era of industry 4.0, the future of the Indonesian people is depended on the young generation. Creativity and innovation from young people will create to a variety of new economic resources that will become the force of the nation's economy in the Industry 4.0. This was conveyed by the Minister of Research, Technology and Higher Education (Menristekdikti) Mohamad Nasir.

The Internet of Things is a fundamental foundation in the industry 4.0. The internet network of Things can connect CCTV cameras (IP cameras), intersection lights, street lights and other electronic facilities. The electronic facility management feature is one of the mandatory assets for the City that wants to implement the smart city concept.

Behind the convenience offered by the existence of the Internet of Things, a new problem arises. One problem that arises is the attack of the Internet of Things network by viruses. There are two viruses that have been identified. The viruses are Mirai and Bashlite.

Mirai is a malware that turns networked devices running Linux into remotely controlled "bots" that can be used as part of a botnet in large-scale network attacks. BASHLITE is malware which infects Linux systems in order to launch distributed denial-of-service attacks (DDoS). Both viruses are very detrimental because they can infect devices that are connected to IoT networks so that the connected devices will be controlled by the attacker.

With the development of research in the biomedical field, these challenges must be solved [1]. Several previous works have been proposed. For example, the PAYL IDS which models simple histograms of packet content [2] or the k-NN algorithm [3]. These methods are either very simple and therefore produce very poor results, or require accumulating data for the training or detection. A popular algorithm for network intrusion detection is the ANN. The detection of anomalies concluded with 100% TPR. This is because of its ability to learn complex concepts, as well as the concepts from the domain of network communication [4]. This study classified the datasets that have attack patterns by Mirai and Bashlite viruses. The method is based on Random Forest along with Naive Bayes method, K-Nearest Neighbor and Decision Tree as a comparison. These methods are a popular method in data mining for a classification [4].

2 Related Work

2.1 Random Forest

Random Forest is an ensemble classifier that produces many decision trees, using a sample subset and randomly selected training variables. (Mariana Belgiu & Lucian Dragut, 2015). The procedure for modeling the Random Forest is as follows [5]:

1. Select the m feature from the M feature that is random. With the number m not more than M.
2. Calculate the best split point for the k tree based on the separation metric (Gini impurity, etc.) and separate the current node into the child node and reduce the number of M features of this node.
3. Repeat steps 1 and 2 until the maximum tree depth l is reached or the separation matrix reaches the extreme.
4. Repeat steps 1 through 3 for each tree in the forest.
5. Vote on the output of each tree in the forest

In previous study [6], Random forest is used to detect Peer-to-Peer Botnet attacks. The Random Forest algorithm is chosen because detection problems Botnet has high prediction accuracy requirements, the ability to handle multiple bots, the ability to handle data that is characterized by a very large number and various types of descriptors, ease of training, and computational efficiency. High dimensional data without reduction or sample selection descriptors make ANN and k-NN not very efficient.

2.2 Naive Bayes

Naive Bayes classifier is the simple Statistical Bayesian Classifier. It is called Naive as it assumes that all variables contribute towards classification and are mutually correlated. This assumption is called class conditional independence [7].

The Naive Bayes classifier works as follows:

1. Let D be the training dataset associated with class labels. Each tuple is represented by n-dimensional element vector, X= (x1, x2, x3, …, xn).
2. Consider that there are m classes C1, C2, C3...., Cm. Suppose that we want to classify an unknown tuple X, then the classifier will predict that X belongs to the class with higher posterior probability, conditioned on X. i.e., the Naive Bayesian classifier assigns an unknown tuple X to the class Ci if and only if $P(Ci|X) > P(Cj|X)$ For $1 \leq j \leq m$, and $i \neq j$, above posterior probabilities are computed using Bayes Theorem.

2.3 K-Nearest Neighbor

The K-Nearest Neighbor (K-NN) Algorithm is the simplest of all machine learning algorithms. It is based on the principle that the samples that are similar, generally lies in close vicinity [7].
The K-NN classifier works as follows:

1. Initialize value of K.
2. Calculate distance between input sample and training samples.
3. Sort the distances.
4. Take top K- nearest neighbors.
5. Apply simple majority.
6. Predict class label with more neighbors for input sample.

2.4 Decision Tree

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value Decision tree classification technique is performed in two phases: tree building and tree pruning. Tree building is performed in top-down approach. During this phase, the tree is recursively partitioned till all the data items belong to the same class label [7].

3. Methodology

The flowchart to find the optimal model in this study is as follows:

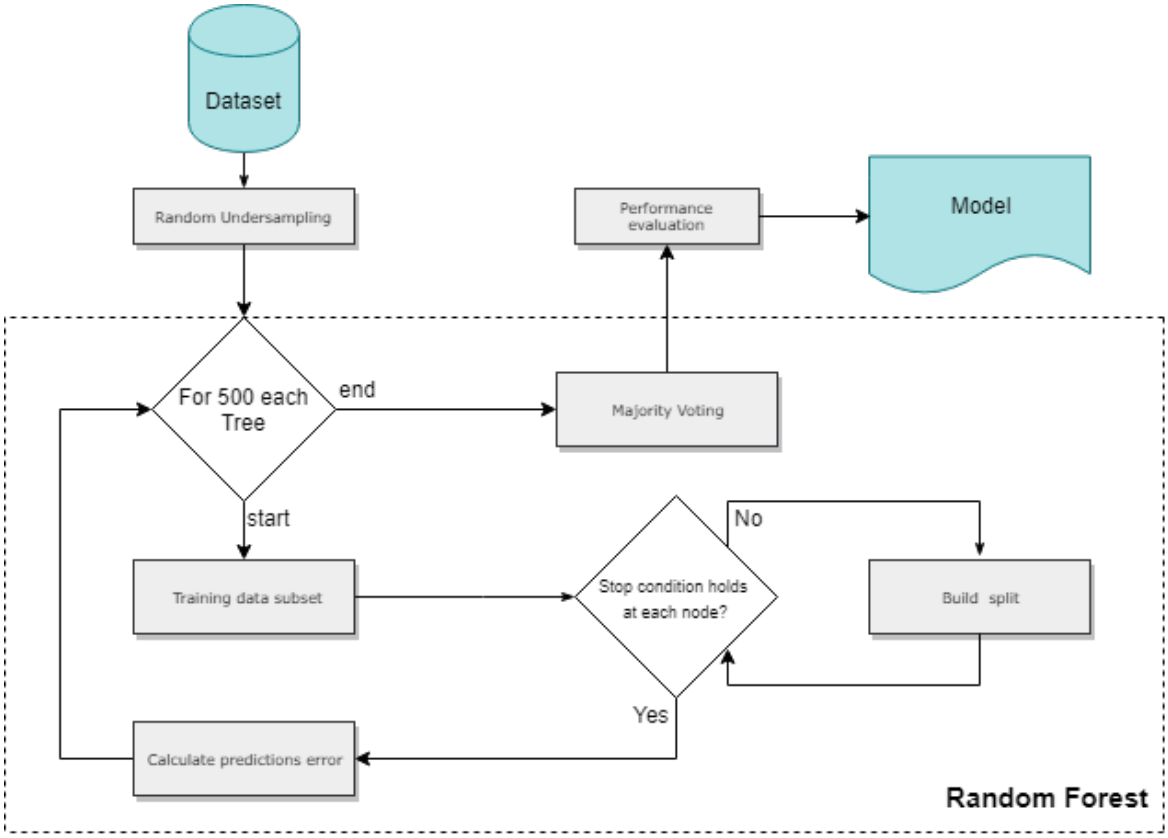

Figure 1. Experiment Design

Traffic datasets were taken from the UCI Machine Learning Repository public domain to detect traffic data on IoT devices. This dataset consists of traffic data recordings on nine diagnosed IoT devices exposed to Mirai and BashLite malware. There are 115 columns as attributes and 100000 number of data record rows. The dataset is divided into subfolders for every nine devices. Each subfolder has three types of data records from the device, based on when the device is working normally, attacked by Mirai malware, and attacked by Bashlite malware. The dataset has an imbalanced class for each device.

The imbalanced class in dataset is handled using the SpreadSubsample under sampling technique in classes 2 and 3 (Mirai and Bashlite). Then randomization is made to the dataset row to flatten the buildup of one class value. Then Random Forest classification is done with 500 trees. The majority of each voting result will build a Botnet malware prediction model.and will be compared with Naive Bayes, KNN, and Decision Tree.

The performance of the model is evaluated by using K-Fold Cross-Validation method. The dataset is divided into K parts with the same volume each part. Then one part is used as test data and the rest is used as training data, this is done as much as K times (Krstajic el al., 2014). In this study, 10-Fold Cross-Validation or K = 10 is done, because this value has been shown empirically to produce estimates of low error and variance [8].

The method proposed for detection of Botnet attack is implemented using Weka 3.8.2 software and a computer with NVIDIA 940MX with 2GB of memory, DDR4 RAM 8.00 GB memory, and Intel Core i5-7200U CPU 3.1 @GHz.. Below is the parameter that has been used in Weka 3.8.2.

Table 1. Parameter of Classification

Method	Parameter	
Random Forest	▪ bagSizePercent=100 ▪ batchSize=100 ▪ breakTiesRandomly=False ▪ calcOutOfBag=False ▪ computeAttributeImportance=False ▪ Debug=False ▪ DoNotCheckCapabilities=False ▪ maxDepth=0	▪ numDecimalPlaces=2 ▪ numExecutionSlots=1 ▪ numFeature=0 ▪ numIterations=500 ▪ OutputOutOfBagComplexityStatistics=False ▪ PrintClassifiers=False ▪ seed=1 ▪ StoreOutOfBagPredictions=False
Naive Bayes	▪ batchSize=100 ▪ Debug=False ▪ DisplayModelInOldFormat=False ▪ DoNotCheckCapabilities=False	▪ numDecimalPlaces=2 ▪ useKernelEstimator=False ▪ useSupervisedDiscretization=False
KNN	▪ KNN=3/5/7 ▪ BatchSize=100 ▪ CrossValide=False ▪ Debug=False ▪ distanceWeighting=No	▪ DoNotCheckCapabilities=False ▪ meanSquared=False ▪ NNSearchAlgorithm=LinearNNSearc ▪ numDecimalPlaces=2 ▪ WindowSize=0
Decision Tree	▪ batchSize=100 ▪ BinarySplits=False ▪ CollapseTree=True ▪ confidenceFactor=0.25 ▪ Debug=False ▪ DoNotCheckCapabilites=False ▪ DoNotMakeSplitPointActualValue=False ▪ minNumObk=2	▪ numDecimalPlaces=2 ▪ numFolds=3 ▪ ReducedErrorPruning=False ▪ SevaIntanceData=False ▪ seed=1 ▪ SubstreeRaising=True ▪ Unpruned=False ▪ UseLapse=False ▪ UseMDLcorrection=True

4. Result

The accuracy of random forest classification is compared with other algorithms to the dataset. The dataset is balanced with the (SpreadSubsample) technique before. We set the parameters for each classification method as shown in Table 1 above.

Table 2. Classifier Output

	Random Forest	Naive Bayes	KNN K = 3	KNN K = 5	KNN K = 7	Decision Tree
Accuracy	99.99%	99.00%	99.94%	99.93%	99.90%	99.98%
Precision	100%	99.10%	99.90%	99.90%	99.90%	100%
Recall	100%	99.10%	99.90%	99.90%	99.90%	100%
F-Measure	100%	99.10%	99.90%	99.90%	99.90%	100%

In the table it can be seen that the classification using Random Forest gets greatest accuracy 99.99% with Precision, Recall, and F-Measure get 100%. Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. F-Measure is the weighted average of Precision and Recall.

The comparison show the Random Forest has highest accuracy. Random Forest for detection problems Botnet has high prediction accuracy requirements, the ability to handle multiple bots, the ability to handle data that is characterized by a very large number and various types of descriptors, ease of training, and computational efficiency.

For other algorithms. KNN is supervised lazy learner because they store all training samples and do not build classifiers until new unlabeled samples need to be classified and run slowly. Naive Bayes classification requires a very large number of records to get good results and less accurate than other classifiers on several data sets. Decision trees have more fitting problems and Decision trees can have far more complex representations for some concepts due to replication problems [7].

5. Conclusion

Botnet attacks are one of the biggest challenges that security researchers and analysts face today. The Internet of Things is a fundamental foundation for industry 4.0. This paper addresses the problem on smart city that Botnet attacks on IoT device could mean a big trouble. Thus, in this paper we solve it with random forest classifier to classify the data traffic anomaly quickly and accurate.

The comparison results in the table show that Random Forest produces the highest accuracy for detection of Botnet data traffic compared to other methods. The model with the Random Forest reaches 99.99% accuracy as good as ANN algorithm in previous work. We hope this paper could be a reference for quick classification of this data set without sacrificing too much resource so the decision can be taken cheaper and accurate.

6. References

[1] Yulita, I. N., & Wasito, I. 2013. gCLUPS: Graph clustering based on pairwise similarity. In Information and Communication Technology (ICoICT), 2013 International Conference of (pp. 77-81). IEEE.

[2] Ke Wang and Salvatore J Stolfo. Anomalous payload-based network intrusion detection. In RAID, volume 4, pp. 203–222

[3] Miao Xie, Jiankun Hu, Song Han, and Hsiao-Hwa Chen (2013). Scalable hypergrid k-nn-based online anomaly detection in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 24(8):1661–1670
[4] Yulita, I. N., Fanany, M. I., & Arymurthy, A. M. (2018). Fast Convolutional Method for Automatic Sleep Stage Classification. *Healthcare informatics research*, 24(3), 170-178.
[5] Stavtos I. Dimitriadis, Dimitris Liparas (2018). "How random is the random forest? Random forest algorithm on the service of structural imaging biomarkers for Alzheimer's disease: from Alzheimer's disease neuroimaging initiative (ADNI) database
[6] Kamaldeep Singh , Sharath Chandra Guntuku , Abhishek Thakur , Chittaranjan Hota (2014). Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests. *Information Sciences*, volume 278, pp. 488-497
[7] Sayali D. Jadhav, and H. P. Channe. 2014. Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. *International Journal of Science and Research (IJSR)* Volume 5 Issue 1, pp.1842 –1845.
[8] James Gareth, Hastie Trevor, Tibshirani Robert, Witten Daniela (2013). *An Introduction to Statistical Learning: With Applications in R*. Switzerland: Springer

Acknowledgments
We would like to thank Yair Meidan from Ben-Gurion University of the Negev for providing helpful support regarding Detection of IoT Botnet Attacks dataset.
This article is presented at the International Conference on Smart City Innovation 2018 that supported by the United States Agency for International Development (USAID) through the Sustainable Higher Education Research Alliance (SHERA) Program for Universitas Indonesia's Scientific Modeling, Application, Research and Training for City-centered Innovation and Technology (SMART CITY) Project, Grant #AID-497-A-1600004, Sub Grant #IIE-00000078-UI-1.