ISSN (p) : 2407-1323 DOI : https://doi.org/10.25124/jett.v9i2.5437
ISSN (e) : 2442-4404 | Vol. 9 | No. 2 | Halaman : 1271 - 1280

JETT

Jurnal Elektro Telekomunikasi Terapan Desember 2022

SECURING THE INTERNET OF THINGS: ENHANCING INTRUSION DETECTION SYSTEM ON NODEMCU WITH ENSEMBLE VOTING

Aji Gautama Putrada[1], Nur Alamsyah 2[2], Mohamad Nurkamal Fauzan[3], Syafrial Fachri Pane[4]

[1,2,3,4]Advanced and Creative Networks Research Center, Telkom University

[1]ajigps@telkomuniversity.ac.id, [2]nuralamsyah@student.telkomuniversity.ac.id, [3]mnurkamalfauzan@student.telkomuniversity.ac.id, [4]fachrie@student.telkomuniversity.ac.id

Abstract

As part of the Internet, a NodeMCU becomes a vulnerability to cyber-attack, so the intrusion detection system (IDS) against cyber-attack on the NodeMCU becomes a poignant research challenge. Several studies have applied several machine learning techniques for IDS on NodeMCU; however, there is a research opportunity to improve the performance of existing models. This research aims to increase the IDS performance on NodeMCU with ensemble voting. We hypothesize the realization of how the implementation of an IDS in NodeMCU. Then we obtain and observe a dataset from Kaggle. The dataset comprises five attacks: misconfiguration, DDoS, probe, scanning, and MiTM. Then we design the detection with ensemble voting consisting of a decision tree and a random forest. We benchmark our proposed solution with decision trees and random forest performance. This study uses several test metrics, including $Accuracy$, $Precision$, $Recall$, and $F1-Score$.The test results show that the decision tree has better *Precision* in predicting misconfiguration attacks and scan attacks than random forests. On the other hand, the random forest has better $Precision$ in predicting normal data, DDoS attacks, probe attacks, and MiTM attacks. In terms of $Accuracy$, ensemble voting has the best performance, which is 0.996, compared to the decision tree and random forest, which are 0.836 and 0.994, respectively. We conclude that by assembling a decision tree in the random forest with ensemble voting, random forest performance can improve. The impact of our study is a novel model for IDS on NodeMCU with ensemble voting between random forest and decision tree.

Keywords: intrusion detection system, NodeMCU, ensemble voting, man in the middle, distributed denial of service, decision tree, random forest, internet of things

1. INTRODUCTION

An intrusion detection system (IDS) is a defense mechanism against cyber-attacks that ensures confidentiality, integrity, and availability (CIA) [1]. On the other hand, NodeMCU is a microcontroller that allows embedded system communication through the Internet network in an Internet of Things (IoT) concept [2]. As part of the Internet, NodeMCU is also vulnerable to cyber-attack [3]. IDS against cyber-attack on NodeMCU is a research challenge.

Several studies on NodeMCU cyber-attacks have existed. Doshi *et al.* [4] investigated attack defense against Mongolian distributed denial of service (DDoS), i.e., coordinated DDoS through small-scale attacks. DDoS is an attack on availability, which causes a service to be unusable because, for example, a request flood denies other requests [5]. This research uses an online discrepancy test (ODIT) method, which can detect anomalies when an attack occurs. Ashenafi *et al.* [6] investigated man-in-the-middle (MiTM) attacks on IoT devices, where adversaries do eavesdropping on IoT end-device networks. This research uses machine learning to detect MiTM, where the decision tree

ISSN (p) : 2407-1323 DOI : https://doi.org/10.25124/jett.v9i2.5437
ISSN (e) : 2442-4404 | Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

performs best compared to other methods. MiTM is an attack on confidentiality and integrity, where communication between two parties involves a malicious third party who can steal important information or deflect packets from reaching their intended destination [7].

Several attacks are also a threat to IoT networks. Data type probing is a method of sending the wrong data type. Hasan *et al.* [8] applied machine learning to detect data type probing, where the random forest is the most accurate method compared to other detection methods. Misconfiguration in network and security settings becomes a vulnerability and makes cyber-criminals easy to attack. Srinivasa *et al.* [9] scanned to find misconfiguration vulnerabilities on some IoT devices and turned them into honey pots to provoke and analyze cyber-attacks. Scan attack performs scanning to find open ports to carry out attacks [10]. Zhang *et al.* [11] found that implementing random forest is an efficient solution for detecting scan attacks in IoT.

Ensemble voting combines predictions from several machine learning models to get better accuracy from the voting members. Several studies have implemented ensemble voting in IDS to get better detection performance. Gao *et al.* [12] revealed that ensemble voting could combine the advantages of machine learning in detecting different attacks. Their research showed that ensemble voting is better than decision trees and random forests. Upadhyay *et al.* [13] implemented ensemble voting for IDS in the supervisory control and data acquisition (SCADA) system. The voting ensemble combines decision tree, random forest, extra tree, gradient boosting, extreme gradient boosting, and adaptive boosting (AdaBoost), where the result is better than the six predictors. Using ensemble voting to improve IDS performance on NodeMCU is a research opportunity.

This research aims to increase the IDS performance on NodeMCU with ensemble voting. We hypothesize the realization of how the implementation of an IDS in NodeMCU. Then we obtain and observe a dataset from Kaggle. The dataset comprises five attacks: misconfiguration, DDoS, probe, scanning, and MiTM. Then we design the detection with ensemble voting consisting of a decision tree and a random forest. We benchmark our proposed solution with decision trees and random forest performance. This study uses several test metrics, including $Accuracy$, $Precision$, $Recall$, and $F1-Score$.

To the best of our knowledge, there has never been a study that applies ensemble voting for IDS on NodeMCU. Here are our research contributions:

1. The use of a confusion matrix to evaluate the performance of ensemble voting for up to six classes.
2. An improvement of random forest performance by assembling a decision tree on the model with ensemble voting
3. The implementation of ensemble voting between decision tree and random forest to improve IDS performance on NodeMCU.

The composition of the remainder of this paper is as follows: Section II discusses the design of our system. Section III shows the results of testing our proposed system. Finally, Section IV highlights important findings.

2. METHODOLOGY AND DESIGN

Figure 1 shows our proposed research methodology. We hypothesize the implementation of an IDS in NodeMCU. Then we delve into a dataset from Kaggle. The dataset comprises five attacks: misconfiguration, DDoS, probe, scanning, and MiTM. Then we design the detection with ensemble voting consisting of a decision tree and a random forest. We benchmark our method with the decision tree and random forest models. This study uses several test metrics, including $Accuracy$, $Precision$, $Recall$, and $F1-Score$.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

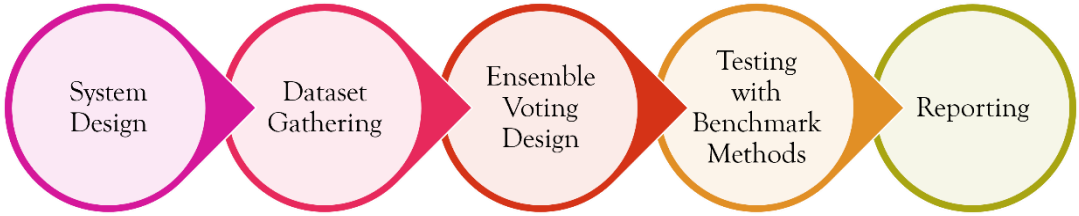


Figure 1. The research methodology.

2.1 Data Preparation

We studied several state-of-the-art designs for IDS in IoT and applied them to our NodeMCU system [14]. Figure 2 shows the design where our NodeMCU connects to an IoT network that is part of the Internet network. In the network, several adversaries can carry out several different attacks, namely misconfiguration, DDoS, probes, scans, and MiTM. We collect such data through network analysis, where such analysis can use Wireshark. We take the necessary features to make predictions. A voting ensemble that we have trained predicts whether there is an attack or not and explains the type of attack. There is an interface to show the attack to the user.

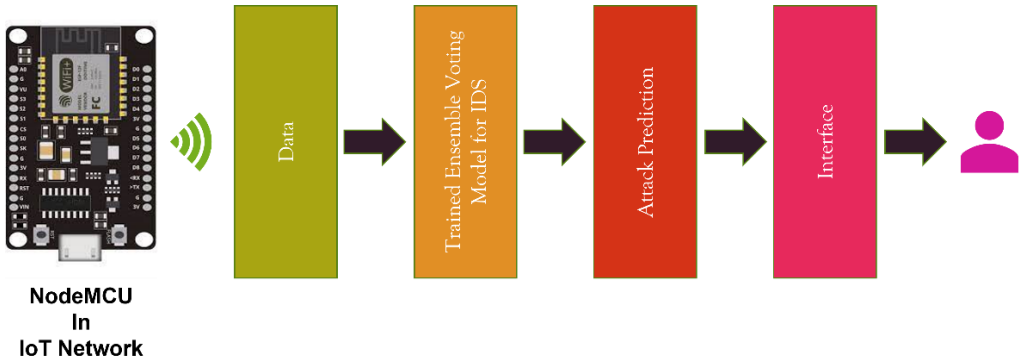


Figure 2. IDS on NodeMCU model design.

We retrieved IoT Device Network Logs for the attack dataset from Kaggle [15]. Table 1 describes each feature in the dataset. The data we have has four main parts: data link, network, transport, and application [16]. Frame time describes the arrival time of the frame at the data link layer, while IP Protocol describes what protocol the packet uses at the network layer. Value is the contents of the package.

Table 1 Dataset feature explanation

Feature	Explanation	Layer
Frame Number	Frame Identification Number	Data Link Layer
Frame Time	Frame Arrival Time	
Frame Length	Frame Size	
Ethernet Source	Frame Source Physical Identifier	
Ethernet Destination	Frame Destination Physical Identifier	
IP Source	Packet Source Logical Identifier	Network Layer
IP Destination	Packet Destination Logical Identifier	
IP Protocol	Packet IP Protocol	
IP Length	Packet Size	
TCP Length	Stream Size	Transport Layer
TCP Source Port	Stream Source Port Identifier	
TCP Destination Port	Stream Destination Port Identifier	
Value	Payload	Application Layer

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

A normality column in the dataset describes the attack type with numbers. Here we break down each number:

- 0 is for normal data
- 1 is for data misconfiguration attack
- 2 is for DDoS attack data
- 3 is for probe attack data
- 4 is for data scan attack
- 5 is for MiTM attack data

We apply normalization and standardization to the dataset in the pre-processing stage [17]. Normalization and standardization can reduce misclassification caused by different data ranges between each feature. Normalization functions to make all features have the same range. The normalization formula is as follows:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x is the feature, x_{min} is the smallest feature value, x_{max} is the largest feature value, and x' is the standard feature value.

Standardization serves to make the data average 0. The following is the standardization formula after the dataset has gone through normalization:

$$x'' = \frac{x' - \mu}{\sigma} \quad (2)$$

where μ is the feature mean, σ is the standard deviation of the feature, and x'' is the standardized normal feature.

2.3 Ensemble Voting

Ensemble voting is an ensemble classifier of stacking type or meta-classifier, a classifier that performs classification based on the results of other classifiers [18]. We propose ensemble voting between decision trees and random forests so that, hypothetically, the ensemble voting results combine each classification's predictive goodness [19]. Figure 3 shows our proposed voting ensemble. The compilers of this ensemble model are C_1 and C_2, where C_1 is the decision tree model while C_2 is the random forest model. C_1 produces P_1 and C_2 produces P_2, which respectively are prediction results. The ensemble voting produces a final prediction, or P_f, where the P_f formula is as follows:

$$P_f = mode\{C_1(x)C_2(x)\} \quad (3)$$

where x is the input data.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

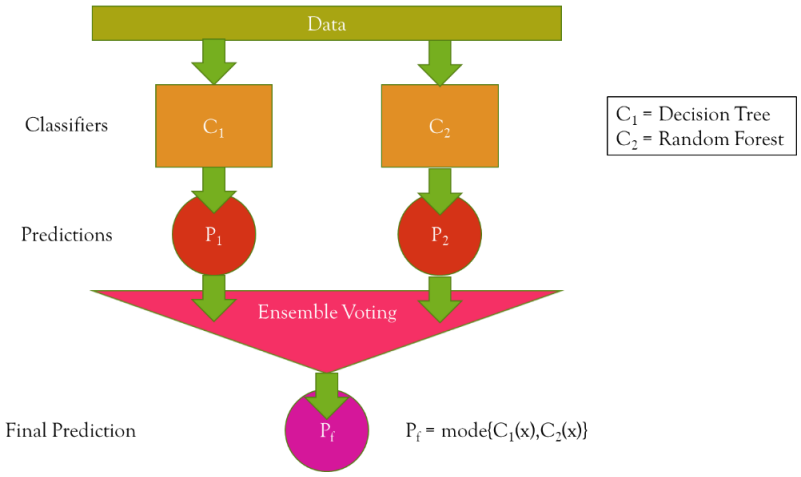


Figure 3. Ensemble Voting Architecture.

The decision tree, one of the stacks in this voting ensemble, is a decision maker whose model branches the training results from the train data [20]. In forming the tree, this algorithm uses the Gini index, which is a value that represents the inequality of a label in the feature. The Gini Formula Index is as follows:

$$Gini(D) = 1 - \sum_{i \in m} p_i^2 \quad (4)$$

where D is a feature, then p_i represents the probability that a label exists in a feature, where m represents the number of labels present. The smaller the Gini index value, the better the decision tree uses that feature as a decision branch.

Random forest is also a bootstrap aggregating (bagging) type of ensemble learning, where the bootstrapping process generates a sub-sample of the dataset and builds a weak learner based on the sub-sample [21]. The weak learner used is usually a decision tree [22]. Then aggregating is taking a decision based on these decision trees by majority voting. Figure 4 shows the algorithm. In the algorithm, N represents the number of weak learners, then F_i represents the weak learners with $i \in N$.

Algorithm 1: Random Forest Training Algorithm

Data: N, X, y
Result: $\hat{y}$
for i *in* $range(N)$ **do**
  $X_s \leftarrow CreateSubSample\ (\ X\ )$;
  $y_s \leftarrow CreateSubSample\ (\ y\ )$;
  $F_i \leftarrow Train\ (\ X_s, y_s\ )$;
end
$\hat{y} = mode(F_0(x), .., F_N(x))$; /* x is the prediction input */

Figure 4. Random Forest Algorithm.

2.4 Test Parameters

Some of the testing parameters in this study are Accuracy, Precision, Recall, and F1-Score [23]. Its formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1 - Score = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

where TP is true positive predictions, TN is true negative predictions, FP is false positive predictions, and FN is false negative predictions.

3. RESULTS DISCUSSION

3.1 Results

First, we evaluate the dataset with the Pearson correlation coefficient (PCC). PCC shows the effect of features on labels [24]. Table 2 gives the results of the PCC score for each feature. The feature with a strong correlation with the label is the IP protocol; the correlation is negative [25]. Five features have moderate correlation, namely IP length, frame length, TCP source port, TCP destination port, and TCP length, each with a negative correlation. The remaining seven features have a very low correlation: ethernet destination, IP destination, IP source, ethernet source, frame number, frame time, and value.

Table 2 The rank of features based on the PCC score.

Feature	PCC Score
IP Protocol	-0.76
IP Length	-0.54
Frame Length	-0.47
TCP Source Port	-0.46
TCP Destination Port	-0.43
TCP Length	-0.4
Ethernet Destination	0.19
IP Destination	-0.17
IP Source	-0.15
Ethernet Source	0.14
Frame Number	-0.022
Frame Time	-0.037
Value	$6.8x10^{-05}$

There are 477,426 data items in our dataset. We divide the dataset by the composition $train: test = 50:50$. We apply stratification to the dataset, which creates splits with proportional divides of labels. We build a decision tree with the default number of weak learners, 200. Then the decision tree and random forest have the same max depth, which is 3. Max depth determines the allowed furthest distance from the root to a leaf. Max depth that is too high can result in overfitting.

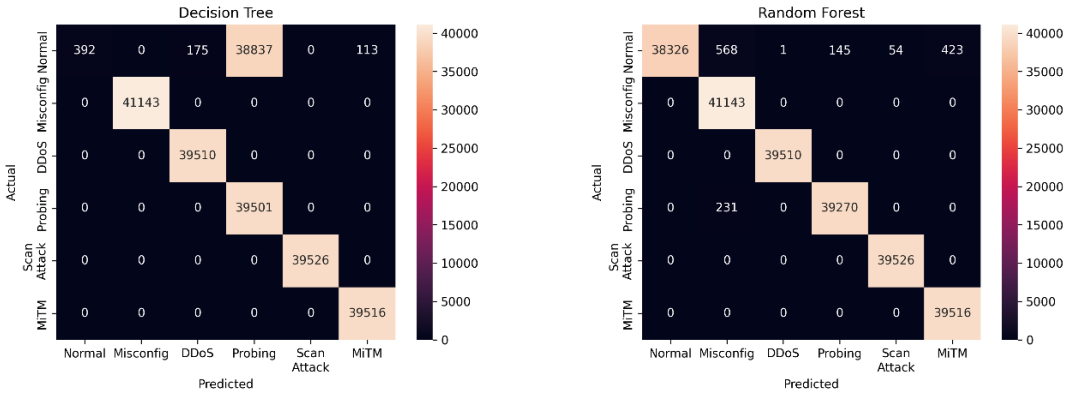
We compare the decision tree and the random forest model without ensemble voting. It allows us to show our motivation to use ensemble learning. First, we compare the $Accuracy$, $Precision$, $Recall$, and $F1 - Score$ of the decision tree and random forest. Table 3 shows the comparison of the performance of the decision tree and random forest. The random forest has the higher performance of the four metrics. The highest performance of the decision tree is Precision, which is 0.916. At the same time, the lowest performance is $F1 - Score$ with a value of 0.781.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

Table 3 Decision tree and random forest performance comparison

Prediction Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.836	0.916	0.834	0.781
Random Forest	**0.994**	**0.994**	**0.994**	**0.994**

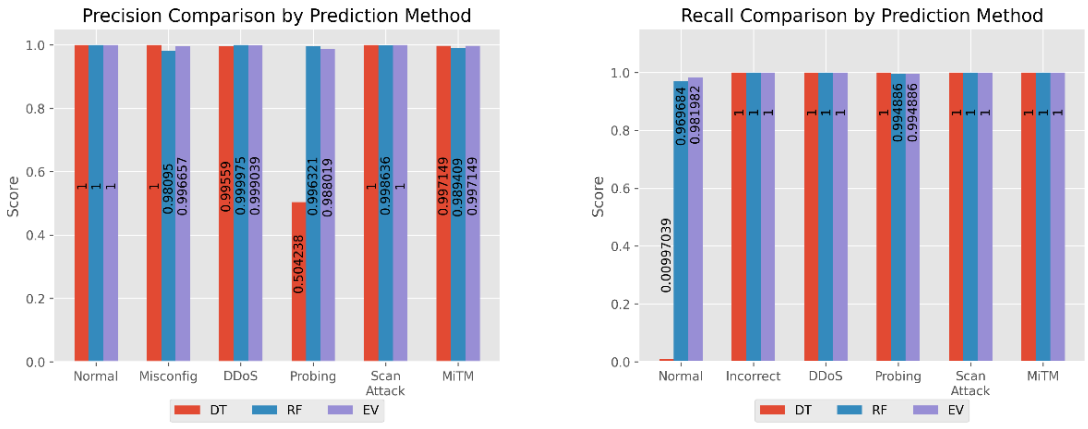
We deepen the discussion on the comparison of decision trees with random forests. Figure 5 compares the confusion matrix between the decision tree and random forest model prediction. Decision trees and random forests have different abilities to predict each class. Random forest is better at classifying normal data, misconfiguration attacks, probing attacks, and scan attacks. On the other hand, even though the decision tree has a lower performance than the random forest, the decision tree is still better at classifying DDoS attacks and scan attacks. These properties are the basis for our use of ensemble voting.



(a) (b)

Figure 5. Confusion matrix of: (a) Decision tree (b) Random forest.

In the comparison of Figure 6, we have added our proposed ensemble voting implementation. The precision bar clarifies that decision trees are better than random forests in predicting misconfiguration attacks and scan attacks. Ensemble voting adopted this, as seen from the ability of the ensemble voting, which is better than random forest in the two classes. The voting ensemble adopts the random forest capability for the remaining four classes. This performance has an impact on the recall value. As a result, the voting ensemble has a better recall than the decision tree and random forest predicting normal data.



(a) (b)

Figure 6. Performance comparison of decision tree, random forest, and ensemble voting on each class: (a) Precision (b) Recall.

ISSN (p) : 2407-1323
ISSN (e) : 2442-4404
DOI : https://doi.org/10.25124/jett.v9i2.5437
| Vol. 9 | No. 2 | Halaman : 1271 - 1280
JETT
Jurnal Elektro Telekomunikasi Terapan Desember 2022

Finally, we compare $Accuracy$, $Precision$, $Recall$, and $F1-Score$ from the decision tree, random forest, and ensemble voting. Figure 7 shows the comparison. In terms of $Accuracy$, which describes the overall model capability, ensemble voting has the best performance, which is 0.996, compared to the decision tree and random forest, which are 0.836 and 0.994, respectively. We conclude that assembling a decision tree in a random forest model with ensemble voting can improve the random forest performance.

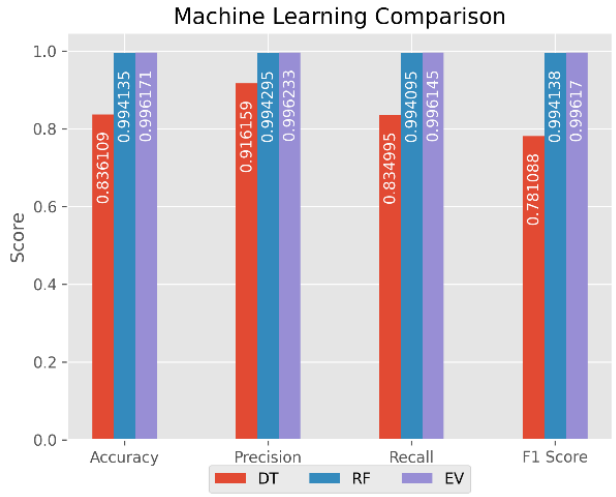


Figure 7. Performance comparison of decision tree, random forest, and ensemble voting.

3.2 Discussions

Paper [26] used a confusion matrix to analyze the effectiveness of ensemble voting in combining the two models, as we do in our research. However, the model in that study only involved two classes, whereas our model involves six classes. Our research contributes to using a confusion matrix for evaluating the performance of a voting ensemble of up to six classes.

Ensemble voting between the decision tree and random forest produces a model with better performance than the two stack models. This result is similar to studies [27], [28], which also showed better ensemble voting performance than the stack. However, the two pieces of research are in bank counterfeit and semiconductors. Our research contributes to the application of ensemble voting between decision trees and random forests to improve IDS performance on NodeMCU.

This study uses data from the paper [15], which applied a random forest to detect five different types of attacks on the NodeMCU connected to the IoT network. Here we find that the decision tree, although it performs worse than the random forest, has better predictive ability than the random forest in detecting misconfiguration attacks and scan attacks. Our research contributes to improving the performance of the random forest model by assembling a decision tree on the model with ensemble voting.

In the cyber security lifecycle, our research lands in the realm of identification [29]. It is important to follow through the process to protection so the identified intrusion can be well-handed. In a machine learning life cycle, the next stage after model testing is model deployment. The limitation of our study is to deploy our novel model then testing its performance on real cyber-attack data.

4. CONCLUSION

We have successfully presented an IDS design for the IoT network connected to NodeMCU. We implemented ensemble voting consisting of a decision tree and random forest to improve IDS

performance for NodeMCU. We compared ensemble voting with decision trees and random forests as benchmarks. The test results show that the decision tree has better Precision in predicting misconfiguration attacks and scan attacks than random forests. On the other hand, the random forest has better $Precision$ in predicting normal data, DDoS attacks, probe attacks, and MiTM attacks. In terms of $Accuracy$, ensemble voting has the best performance, which is 0.996, compared to the decision tree and random forest, which are 0.836 and 0.994, respectively. We conclude that by assembling a decision tree in the random forest with ensemble voting, random forest performance can improve.

In a machine learning life cycle, the next stage after model testing is model deployment. The limitation of our study is to deploy our novel model then testing its performance on real cyber-attack data.

REFERENCES

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, 2021.

[2] M. Kashyap, V. Sharma, and N. Gupta, "Taking MQTT and NodeMcu to IOT: communication in Internet of Things," *Procedia Comput. Sci.*, vol. 132, pp. 1611–1618, 2018.

[3] D. K. Alferidah and N. Z. Jhanjhi, "A review on security and privacy issues and challenges in internet of things," *Int. J. Comput. Sci. Netw. Secur. IJCSNS*, vol. 20, no. 4, pp. 263–286, 2020.

[4] K. Doshi, Y. Yilmaz, and S. Uludag, "Timely detection and mitigation of stealthy DDoS attacks via IoT networks," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2164–2176, 2021.

[5] A. A. Ramadhan, "Analisis Perbandingan Pengujian Distributed Denial of Service (DDoS) dan Rushing Attack pada Jaringan UDP dengan Routing AODV," 2017.

[6] A. Ashenafi, "A Model to Detect MiTM Attack in IoT Networks: A Machine Learning Approach," PhD Thesis, St. Mary's University, 2022.

[7] M. I. Fajrin, P. Sukarno, and A. G. P. Satwiko, "Perbandingan Metode K-NN dan Markov Chain Untuk Deteksi Anomali Serangan Man-in-the-Middle Pada Smart Lock Berbasis Wifi," *EProceedings Eng.*, vol. 5, no. 3, 2018.

[8] M. Hasan, Md. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet Things*, vol. 7, p. 100059, Sep. 2019, doi: 10.1016/j.iot.2019.100059.

[9] S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, "Open for hire: attack trends and misconfiguration pitfalls of IoT devices," in *Proceedings of the 21st ACM Internet Measurement Conference*, New York, NY, USA, Nov. 2021, pp. 195–215. doi: 10.1145/3487552.3487833.

[10] J. Gadge and A. A. Patil, "Port scan detection," in *2008 16th IEEE International Conference on Networks*, Dec. 2008, pp. 1–6. doi: 10.1109/ICON.2008.4772622.

[11] Y. Zhang *et al.*, "Efficient and Intelligent Attack Detection in Software Defined IoT Networks," in *2020 IEEE International Conference on Embedded Software and Systems (ICESS)*, Dec. 2020, pp. 1–9. doi: 10.1109/ICESS49830.2020.9301591.

[12] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[13] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, 2021.

[14] Z. Abou El Houda, B. Brik, and L. Khoukhi, "'Why Should I Trust Your IDS?': An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1164–1176, 2022.

[15] R. Hattarki, S. Houji, and M. Dhage, "Real Time Intrusion Detection System For IoT Networks," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1–5.

[16] A. G. Putrada, "Perancangan Software PDU Encoder dan PDU Decoder untuk Layer MAC WiMAX," *Indones. J. Comput. Indo-JC*, vol. 1, no. 1, pp. 24–36, 2016.
[17] B. A. Fadillah, A. G. Putrada, and M. Abdurohman, "A Wearable Device for Enhancing Basketball Shooting Correctness with MPU6050 Sensors and Support Vector Machine Classification," *Kinet. Game Technol. Inf. Syst. Comput. Netw. Comput. Electron. Control*, 2022.
[18] A. L. Prodromidis and S. J. Stolfo, "Cost complexity-based pruning of ensemble classifiers," *Knowl. Inf. Syst.*, vol. 3, no. 4, pp. 449–469, 2001.
[19] D. Patil and J. Patil, "Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique," *Cybern. Inf. Technol.*, vol. 18, pp. 11–29, Mar. 2018, doi: 10.2478/cait-2018-0002.
[20] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, "Machine Learning Methods in Smart Lighting Toward Achieving User Comfort: A Survey," *IEEE Access*, vol. 10, pp. 45137–45178, 2022, doi: 10.1109/ACCESS.2022.3169765.
[21] S. Karuniawati, A. G. Putrada, and A. Rakhmatsyah, "Optimization of grow lights control in IoT-based aeroponic systems with sensor fusion and random forest classification," in *2021 International Symposium on Electronics and Smart Devices (ISESD)*, 2021, pp. 1–6.
[22] A. G. Putrada and D. Perdana, "Improving Thermal Camera Performance in Fever Detection during COVID-19 Protocol with Random Forest Classification," in *2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, 2021, pp. 1–6.
[23] N. G. Ramadhan, A. G. Putrada, and M. Abdurohman, "Improving smart lighting with activity recognition using hierarchical hidden Markov model," *Indones. J. Comput. Indo-JC*, vol. 4, no. 2, pp. 43–54, 2019.
[24] M. B. Satrio, A. G. Putrada, and M. Abdurohman, "Evaluation of Face Detection and Recognition Methods in Smart Mirror Implementation," in *Proceedings of Sixth International Congress on Information and Communication Technology*, 2022, pp. 449–457.
[25] M. Selvanathan, N. Jayabalan, G. Saini, M. Supramaniam, and N. Hussain, "Employee Productivity in Malaysian Private Higher Educational Institutions.," *PalArchs J. Archaeol. Egypt Egyptol.*, vol. 17, pp. 66–79, Oct. 2020, doi: 10.48080/jae.v17i3.50.
[26] V. C. Osamor and A. F. Okezie, "Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021.
[27] R. S. Khairy, A. Hussein, and H. ALRikabi, "The detection of counterfeit banknotes using ensemble learning techniques of AdaBoost and voting," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 326–339, 2021.
[28] M. Saqlain, B. Jargalsaikhan, and J. Y. Lee, "A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 171–182, 2019.
[29] M. Mylrea, S. N. G. Gourisetti, and A. Nicholls, "An introduction to buildings cybersecurity framework," in *2017 IEEE symposium series on computational intelligence (SSCI)*, 2017, pp. 1–7.