ICIC Express Letters
Part B: Applications
Volume **13**, Number **7**, July **2022**

ICIC International ©2022 ISSN 2185-2766
pp. 715–723

SMOTE AND RANDOM FOREST ALGORITHM FOR DETECTING DOS ATTACKS IN WIRELESS SENSOR NETWORKS

Gregorius Neven Yusuf and Dennis Gunawan*

Faculty of Engineering and Informatics
Universitas Multimedia Nusantara
Jl. Boulevard, Gading Serpong, Kelapa Dua, Kab. Tangerang 15111, Indonesia
neven.yusuf@student.umn.ac.id; *Corresponding author: dennis.gunawan@umn.ac.id

Received November 2021; accepted January 2022

Abstract. *Wireless Sensor Network (WSN) is a wireless network formed by a collection of sensors (nodes) used to sense and control its surrounding environment. WSN is considered a critical system, because it often handles important information. In addition, WSN is usually placed in extreme places that humans are difficult to reach. This causes WSN to be vulnerable to attacks by irresponsible parties. The most common attacks are denial-of-service attacks at the network layer. One solution to this problem is to use machine learning to detect and classify these attacks. The method used in this research was the Random Forest (RF) algorithm, which was enhanced using SMOTE (Synthetic Minority Oversampling Technique). SMOTE was used to oversample and balance the training data. SMOTE and RF algorithm were successfully implemented to classify the four denial-of-service attacks with an overall accuracy of 99.537% based on the evaluation results of the models that were formed.*

Keywords: Decision tree, Denial-of-service, Random Forest, SMOTE, Wireless sensor network

1. **Introduction.** Wireless Sensor Network (WSN) is a wireless network formed by a collection of sensors, called nodes, that communicate with each other to sense and control its surrounding environment [1]. This control can be done, because each node on the WSN is able to process the physical properties of the things being sensed and transmit that information to a central control system, so that it can be further processed and then a response can be produced [2].

The applications of WSNs in today's daily life can be considered as a critical system, because it handles sensitive data and important information [3-5]. Radhappa et al. [3] revealed that the development of WSN in this decade is at its peak. Zanaj et al. [6] also mentioned that WSN applications have grown significantly in various fields and are still growing. The applications cover environmental safety, supervision of structural robustness, animal control, precision farming, and implementation in smart buildings [3,7]. Every node in WSN is equipped with computing, sensing, and power management tools, as well as tools to send and receive radio signals, so that the sensors can communicate with each other by radio signals wirelessly [1]. At this time, tools that utilize radio signals can be obtained easily and at low prices due to its large availability in the market, and therefore irresponsible parties can easily use the tools to launch attacks against wireless networks [8].

One of the attacks that can target a wireless network is Denial-of-Service (DoS) attack, which has the characteristic of overwhelming the target with a series of false requests in large numbers with the aim to overload the target and making it unable to handle original

DOI: 10.24507/icicelb.13.07.715

requests [9-11]. Gu et al. [9] also added that nowadays, the number of DoS attacks is increasing significantly.

DoS attacks on WSNs can occur at all five layers of the TCP/IP protocol and have various types, but research by Gunduz et al. [12] revealed that DoS attacks at the network layer have the most variety. More than that, applications of WSN require these sensors to be placed at extreme and difficult-to-reach places [13-15]. In a study conducted by Kim et al. [16], it was revealed that the ineffectiveness of handling DoS attacks was caused by mis-configuration and resource unavailability to keep up with the dynamic changes of network technologies without human interference. This means automation solutions, which can apply countermeasures against attacks based on the nature and characteristics of network traffic, need to be used [5,17,18]. This automation solution can be realized by using machine learning.

Many methods or algorithms can be used to analyze DoS attacks. Tan et al. [19], Wankhede and Kshirsagar [20], and Mourabit et al. [21] conducted a comparison of various machine learning techniques to classify attacks on WSN. Based on those three studies, the Random Forest (RF) algorithm yields the best performance compared to several other classification methods, such as Naïve Bayes, multi-layer perceptron, and support vector machines. The RF algorithm is an ensemble method that uses decision tree as the basis for the classification carried out [19]. RF algorithm does not require tree pruning and is immune to the problem of overfitting [20]. RF is also not susceptible to invalid and noise data, and has good scalability to handle classification problems which have high dimensions [19].

RF algorithm implementation is improved using Synthetic Minority Oversampling Technique (SMOTE). SMOTE is an oversampling technique proposed by Chawla et al. [22] to address the problem of data imbalance. Tan et al. [19] revealed that SMOTE is an optimal technique, because it can reduce limitations of previous sampling methods using a basic theory of mathematics, linear interpolation. The authors [19] and other studies by Abdoh et al. [23] and Wu et al. [24] also concluded that the use of SMOTE and RF algorithm provides more accurate classification results.

Another study has been conducted by Almomani et al. [25] using the imbalanced WSN-DS dataset. In that study, Multilayer Perceptron (MLP) was implemented to perform classification of four DoS attacks against the network layer of WSN. Based on the classification accuracies obtained, the classifier yielded an accuracy of 75.6% for an attack class that falls into minority class category in the training data. In this research, this result could be improved by oversampling each of the minority classes data until all of the classes have similar number of records (balanced dataset) before training the data.

This research focuses on the implementation of SMOTE and RF algorithm to detect DoS attacks against the network layer of WSN. SMOTE was used to balance the imbalanced training data before implementing RF algorithm to train and construct the machine learning model. RF hyperparameters tuning was also conducted to find the best RF configuration that could produce machine learning model with the best performance to detect DoS attacks on WSNs.

The remainder of the paper is composed as follows. First, Section 2 describes the methodology used in this research. In Section 3, experiment results are discussed. Finally, Section 4 concludes this research along with suggestions for the future research.

2. Methodology.

2.1. Data collection.

The dataset used in this research is the WSN-DS dataset developed by Almomani et al. [25]. The dataset contains 374,661 records of node data in a wireless sensor network consisting of 100 sensors which are divided into 5 clusters. The records represent the normal (no-attack) behavior and the four types of attacks, namely

Blackhole, Grayhole, Flooding, and Scheduling or TDMA attack. The number of records that represent attacks is 34,595, while the number of normal data records is 340,066.

WSN-DS consists of 15 main features that are used in this research, namely cluster head status, the distance between the node and the cluster head, the number of ADV messages sent and received, the number of join messages sent and received, the number of TDMA schedule messages sent and received, the rank of the node in the TDMA schedule, the number of data packets sent and received, the number of data packets sent to the base station, the distance between the cluster head and the base station, the cluster sending code, and the energy consumption of the node.

2.2. **Research design.** The application model of this research (see Figure 1) could be explained as follows. In data pre-processing step, the original 19 columns in the dataset were filtered to exclude the unnecessary columns such as node ID, time elapsed, cluster head ID, and the attack labels. The 15 remaining columns were used as the features in model building. To start building the machine learning model, the preprocessed dataset was split into train and test data. Data oversampling using SMOTE was implemented to balance the train data. Only the minority class (attack classes) of the train data was oversampled. The oversampling process was repeated until the number of each of the minority class data was almost equal to the number of the majority class (normal). In the next step, Random Forest hyperparameters were configured and machine learning model was trained using the oversampled train data. In model evaluation step, the created model was tested using the original (not oversampled) test data and a confusion matrix was generated based on the test results. Based on the confusion matrix data, the model performance could be further evaluated using several performance metrics.

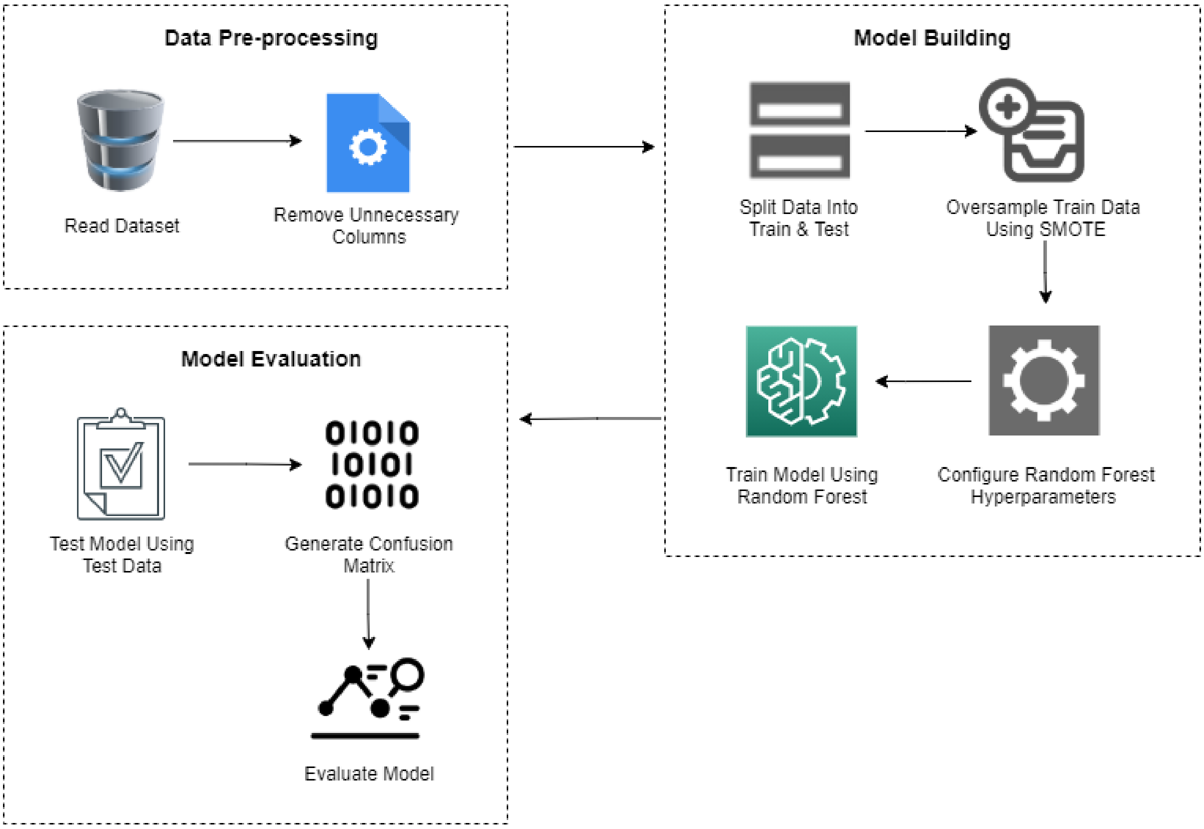
Data Pre-processing
Read Dataset
Remove Unnecessary Columns
Model Building
Split Data Into Train & Test
Oversample Train Data Using SMOTE
Configure Random Forest Hyperparameters
Train Model Using Random Forest
Model Evaluation
Test Model Using Test Data
01010
10101
01010
Generate Confusion Matrix
Evaluate Model

Figure 1. Application model of machine learning model building

2.3. **Experiment.** Experiments were conducted by building and testing several machine learning models built based on various configurations of the Random Forest algorithm used when training the model. In this research, five Random Forest hyperparameters are used as independent variables: the number of decision trees (Trees), maximum tree depth (Max Depth), minimum number of samples required for splitting (Min Samples Split), maximum number of features considered for splitting (Max Features), and maximum number of leaf nodes (Max Leaf Nodes). To aid the experiment, a set of default hyperparameters configuration values was determined as a base for configuring the Random Forest hyperparameters (see Table 1).

Table 1. Random Forest hyperparameters default configuration

Trees	**Max depth**	**Min samples split**	**Max features**	**Max leaf nodes**
100	None	2	3	None

The machine learning model testing was carried out in five scenarios. In each scenario, tunings were made to a Random Forest hyperparameter. For each parameter there were ten tuning values to be tested (see Table 2).

Table 2. Random Forest hyperparameters tuning values

Scenarios	**Values**
1 (Trees)	10, 100, 200, 300, 400, 500, 600, 700, 800, 1000
2 (Max Depth)	None, 5, 10, 20, 30, 40, 50, 60, 70, 80
3 (Min Samples Split)	2, 10, 50, 100, 200, 300, 500, 700, 1000, 1500
4 (Max Features)	1, 2, 3, 4, 5, 6, 7, 8, 10, 15
5 (Max Leaf Nodes)	None, 15, 30, 45, 60, 75, 90, 105, 120, 150

To see how each hyperparameter affects the performance of the model, only one hyperparameter was tuned in each test scenario. The rest of the Random Forests hyperparameters were tuned to the predefined default configuration values (see Table 1).

3. **Results and Discussions.** The constructed machine learning models were tested and evaluated using six performance metrics: Accuracy (A), Precision (P), True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR). Based on the results obtained, the Random Forest hyperparameter configuration that produced the highest accuracy can be seen in Table 3. Table 4 shows the performance metrics results of the model.

Table 3. Random Forest hyperparameters configuration that produced the highest accuracy

Trees	**Max depth**	**Min samples split**	**Max features**	**Max leaf nodes**
800	30	2	6	None

The results obtained from the experiment could also be summarized in Table 5 and Table 6. Table 5 shows the Random Forest hyperparameter configurations of the five models with the best accuracy, while Table 6 shows the performance metrics results of those models.

Based on the results obtained from the various test scenarios, it shows that the Random Forest hyperparameters affect the performance of machine learning models formed. The results of the first scenario (number of decision trees) (see Figure 2) show that accuracy tends to improve as the number of decision trees increases. However, the increase is not significant and it shows that the performance is quite stagnant. Increasing the number

Table 4. Evaluation results of the model with the highest accuracy

Class	A	P	TPR	TNR	FPR	FNR
Blackhole	99.877%	97.030%	98.468%	99.916%	0.084%	1.532%
Flooding	99.925%	92.857%	99.085%	99.933%	0.067%	0.915%
Grayhole	99.815%	97.606%	97.768%	99.900%	0.100%	2.232%
TDMA	99.797%	95.000%	93.079%	99.914%	0.086%	6.921%
Normal	99.660%	99.851%	99.773%	98.551%	1.449%	0.227%
Average	**99.537%**	**96.469%**	**97.635%**	**99.643%**	**0.357%**	**2.365%**

Table 5. Random Forest hyperparameters configuration of the five models with the best accuracy

Rank	Trees	Max depth	Min samples split	Max features	Max leaf nodes
1	800	30	2	6	None
2	100	None	2	6	None
3	100	30	2	3	None
4	800	None	2	3	None
5	100	None	2	3	None

Table 6. Evaluation results of the five models with the best accuracy

Rank	A	P	TPR	TNR	FPR	FNR
1	99.537%	96.469%	97.635%	99.643%	0.357%	2.365%
2	99.534%	96.390%	97.654%	99.645%	0.355%	2.346%
3	99.522%	96.344%	97.599%	99.640%	0.360%	2.401%
4	99.521%	96.340%	97.586%	99.637%	0.363%	2.414%
5	99.514%	96.305%	97.524%	99.628%	0.372%	2.476%

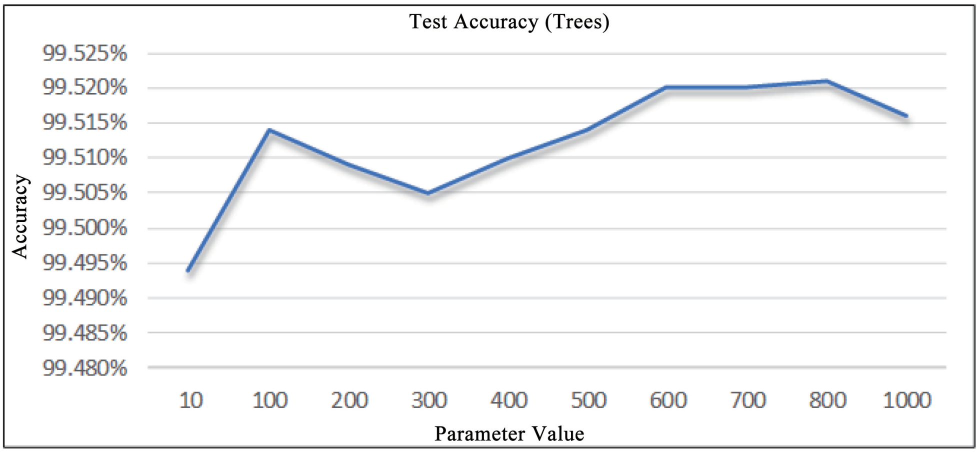

Figure 2. Test result of the first scenario

of decision trees would only increase the computational complexity, and is not significant enough to improve the performance of the model.

The results of the second scenario (maximum tree depth) (see Figure 3) show that the performance improves significantly as the maximum tree depth increases. By using a maximum tree depth value that is too low, the machine learning model would underfit and the classification results are not accurate enough. Other than that, the results of using the maximum tree depth values of 60, 70, 80, and none show the exact same results. This means that naturally, the decision trees formed have depths of no more than 60. These

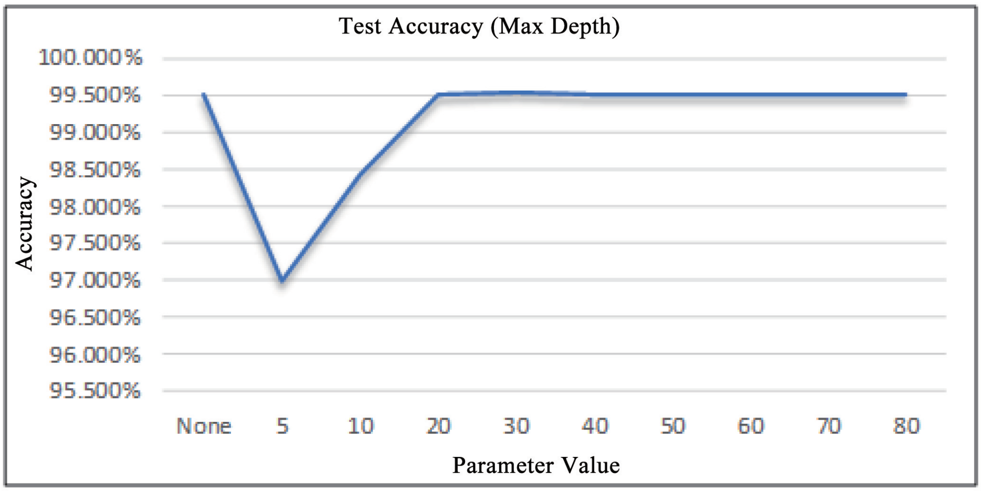

Figure 3. Test result of the second scenario

results conclude that the maximum tree depth parameter has a significant performance effect.

The results of the third scenario (minimum number of samples required for splitting) (see Figure 4) show that as the value of the parameter increases, the performance of the model decreases. If the minimum number of samples required for splitting is too high, the number of splits made by the decision trees will decrease. As a result, the decision trees formed would not provide significant voting criteria and an underfitting model would be formed. In this research, the optimal value of the minimum number of samples required for splitting is 2.

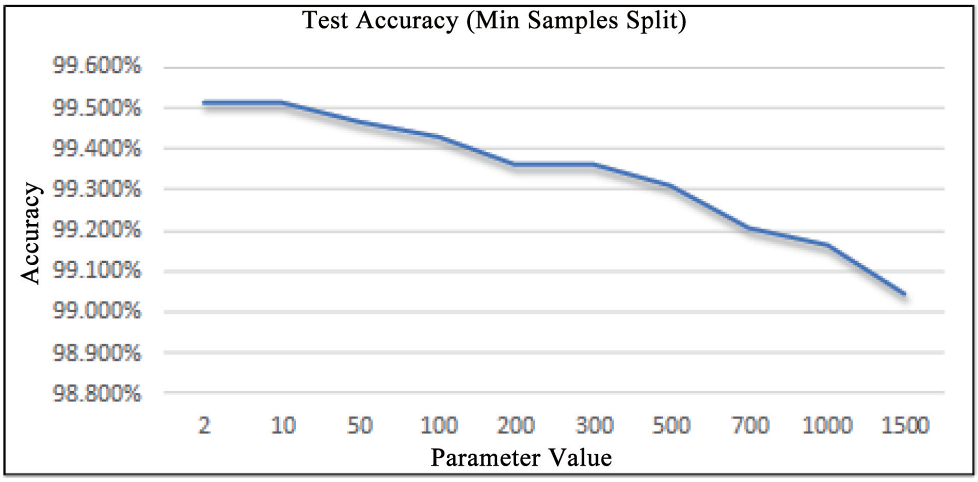

Figure 4. Test result of the third scenario

Test results of the fourth scenario (maximum number of features considered for splitting) (see Figure 5) provide an increasing and decreasing performance trend, although not significant. The highest accuracy was obtained by using parameter value of 6.

Test results of the last scenario (maximum number of leaf nodes) (see Figure 6) show an increasing performance trend. As the value of maximum number of leaf nodes increases, the performance of the machine learning model also increases. The highest accuracy is obtained by not limiting the number of the leaf nodes. Based on these results, it can be concluded that if the value of this parameter is too low, the model will underfit. In this case, by letting it grow naturally without limit, an optimized model could be formed.

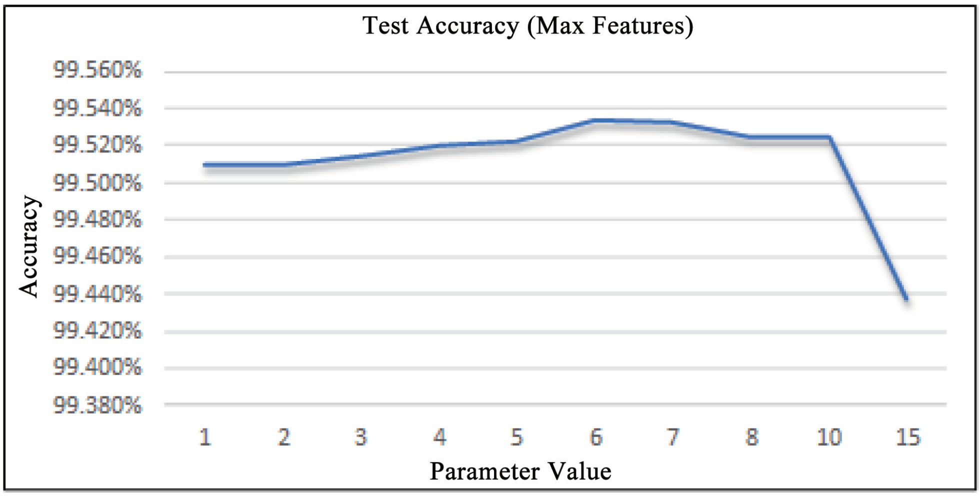

Figure 5. Test result of the fourth scenario

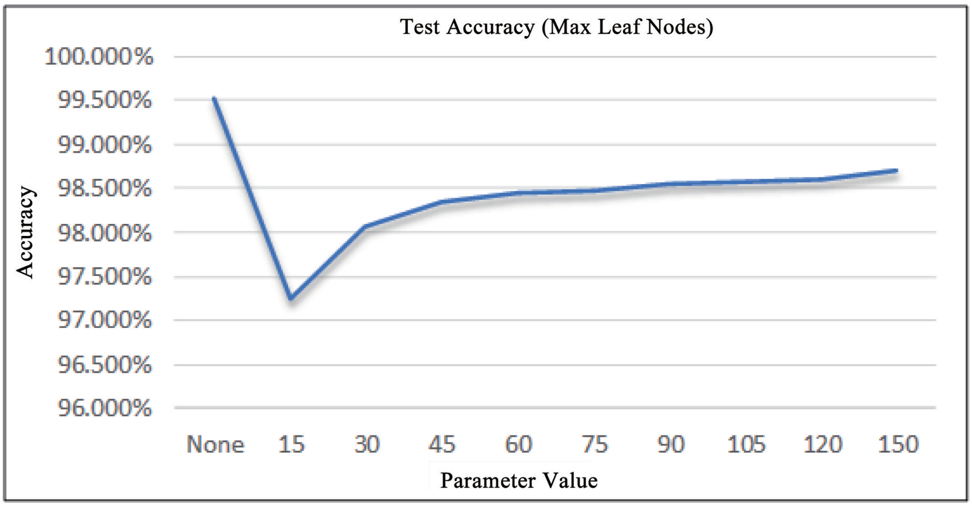

Figure 6. Test result of the fifth scenario

4. **Conclusions and Future Works.** As conclusions, SMOTE and Random Forest can be used to classify four DoS attacks in the network layer of WSN, namely Blackhole, Flooding, Grayhole, and TDMA Attack. Machine learning model with the highest accuracy is obtained with the Random Forest hyperparameters as follows: 800 decision trees, maximum tree depth of 30, minimum sample required for splitting of 2, maximum number of features considered for splitting of 6, and no maximum value of leaf nodes. The highest classification accuracy of each class obtained is 99.877%, 99.925%, 99.815%, 99.797%, and 99.660% for Blackhole, Flooding, Grayhole, TDMA, and no attack, respectively. The overall performance metrics obtained are 99.537% for accuracy, 96.469% for precision, 97.635% for true positive rate, 99.643% for true negative rate, 0.357% for false positive rate, and 2.365% for false negative rate.

In the future, this research can be extended by including and combining other Random Forest hyperparameters, such as split criterion, minimum number of samples in leaf nodes, minimum impurity decrease for splitting, bootstrapping, class weighting, and maximum number of samples. Other than that, train data oversampling using enhanced SMOTE by utilizing binarization techniques, such as OVA (One-vs-All) and OVO (One-vs-One) can also be considered. It is also possible to extend this research to construct and implement an Intrusion Detection System (IDS) to protect WSN from network layer DoS attacks.

REFERENCES

[1] M. Kocakulak and I. Butun, An overview of wireless sensor networks, *IEEE 7th Annu. Comput. Commun. Work. Conf.*, pp.1-6, doi: 10.1109/CCWC.2017.7868374, 2017.

[2] K. M. Modieginyane, B. B. Letswamotse, R. Malekian and A. M. Abu-Mahfouz, Software defined wireless sensor networks application opportunities for efficient network management: A survey, *Comput. Electr. Eng.*, vol.66, pp.274-287, doi: 10.1016/j.compeleceng.2017.02.026, 2017.

[3] H. Radhappa, L. Pan, J. X. Zheng and S. Wen, Practical overview of security issues in wireless sensor network applications, *Int. J. Comput. Appl.*, vol.40, no.4, pp.202-213, doi: 10.1080/1206212X.2017.1398214, 2017.

[4] D. Ramotsoela and A. Abu-mahfouz, A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study, *Sensors*, no.8, doi: https://doi.org/10.3390/s18082491, 2018.

[5] M. Mamdouh, M. A. I. Elrukhsi and A. Khattab, Securing the Internet of Things and wireless sensor networks via machine learning: A survey, *2018 Int. Conf. Comput. Appl.*, pp.215-218, 2018.

[6] E. Zanaj, E. Gambi, B. Zanaj and D. Disha, Customizable hierarchical wireless sensor networks based on genetic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1623-1638, doi: 10.24507/ijicic.16.05.1623, 2020.

[7] S. R. J. Ramson and D. J. Moni, Applications of wireless sensor networks – A survey, *2017 Int. Conf. Innov. Electr. Electron. Instrum. Media Technol.*, pp.325-329, doi: 10.1109/icieeimt.2017.8116858, 2017.

[8] E. M. D. L. Pinto, R. Lachowski, M. E. Pellenz, M. C. Penna and R. D. Souza, A machine learning approach for detecting spoofing attacks in wireless sensor networks, *2018 IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl.*, pp.752-758, doi: 10.1109/AINA.2018.00113, 2018.

[9] Y. Gu, K. Li, Z. Guo and Y. Wang, Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm, *IEEE Access*, vol.7, pp.64351-64365, doi: 10.1109/ACCESS.2019.2917532, 2019.

[10] O. A. Osanaiye, A. S. Alfa and G. P. Hancke, Denial of service defence for resource availability in wireless sensor networks, *IEEE Access*, vol.6, pp.6975-7004, doi: 10.1109/ACCESS.2018.2793841, 2018.

[11] C. Lyu, Selective authentication based geographic opportunistic routing in wireless sensor networks for Internet of Things against DoS attacks, *IEEE Access*, vol.7, pp.31068-31082, doi: 10.1109/ACCESS.2019.2902843, 2019.

[12] S. Gunduz, B. Arslan and M. Demirci, A review of machine learning solutions to denial-of-services attacks in wireless sensor networks, *2015 IEEE 14th Int. Conf. Mach. Learn. Appl. A*, pp.150-155, doi: 10.1109/ICMLA.2015.202, 2015.

[13] I. Almomani and A. Mamdouh, Efficient denial of service attacks detection in wireless sensor networks, *J. Inf. Sci. Eng.*, vol.34, doi: 10.6688/JISE.201807_34(4).0011, 2018.

[14] G. K. Revathi and S. Anjana, Hybrid intrusion detection using machine learning for wireless sensor networks, *Int. J. Innov. Technol. Explor. Eng.*, vol.8, no.12, pp.4867-4871, doi: 10.35940/ijitee.L3721.1081219, 2019.

[15] O. Osanaiye, A. S. Alfa and G. P. Hancke, A statistical approach to detect jamming attacks in wireless sensor networks, *Sensors*, doi: 10.3390/s18061691, 2018.

[16] H. Kim, T. Benson, A. Akella and N. Feamster, The evolution of network configuration: A tale of two campuses, *Proc. of 2011 ACM SIGCOMM Conf. Internet Meas. Conf.*, pp.499-514, doi: 10.1145/2068816.2068863, 2011.

[17] F. S. D. L. Filho, F. A. F. Silveira, A. D. M. B. Junior, G. Vargas-Solar and L. F. Silveira, Smart detection: An online approach for DoS/DDoS attack detection using machine learning, *Secur. Commun. Networks*, vol.2019, doi: 10.1155/2019/1574749, 2019.

[18] E. Baraneetharan, Role of machine learning algorithms intrusion detection in WSNs: A survey, *J. Inf. Technol. Digit. World*, vol.2, no.3, pp.161-173, doi: 10.36548/jitdw.2020.3.004, 2020.

[19] X. Tan et al., Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm, *Sensors (Switzerland)*, vol.19, no.1, doi: 10.3390/s19010203, 2019.

[20] S. Wankhede and D. Kshirsagar, DoS attack detection using machine learning and neural network, *Proc. of 2018 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA2018)*, doi: 10.1109/ICCUBEA.2018.8697702, 2018.

[21] Y. E. Mourabit, A. Toumanari, A. Bouirden and N. E. Moussaid, Intrusion detection techniques in wireless sensor network using data mining algorithms: Comparative evaluation based on attacks detection, *Int. J. Adv. Comput. Sci. Appl.*, vol.6, no.9, pp.164-172, doi: 10.14569/ijacsa.2015.060922, 2015.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, vol.16, pp.321-357, doi: 10.1613/jair.953, 2002.

[23] S. F. Abdoh, M. A. Rizka and F. A. Maghraby, Cervical cancer diagnosis using random forest classifier with SMOTE and feature reduction techniques, *IEEE Access*, vol.6, pp.59475-59485, doi: 10.1109/ACCESS.2018.2874063, 2018.

[24] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou and X. Huang, Intrusion detection system combined enhanced random forest with SMOTE algorithm, *EURASIP J. Adv. Signal Process.*, pp.1-30, doi: 10.21203/rs.3.rs-270201/v1, 2021.

[25] I. Almomani, B. Al-Kasasbeh and M. Al-Akhras, WSN-DS: A dataset for intrusion detection systems in wireless sensor networks, *J. Sensors*, vol.2016, doi: 10.1155/2016/4731953, 2016.