# WSN-BFSF: A New Dataset for Attacks Detection in Wireless Sensor Networks

Murat Dener, Celil Okur, Samed Al, and Abdullah Orman

***Abstract*—The popularity of Wireless Sensor Networks (WSN) increases as the usage areas and the number of integrated systems increase, and this situation attracts the attention of attackers. Attackers carry out attacks aimed at infiltrating, capturing, and manipulating the network. These attacks are implemented differently according to the layers. After these attacks on sensor networks, network traffic data is examined and malicious traffic and node behaviors are analyzed to prevent possible future attacks. The raw data received from the network are made usable by learning models by some preprocessing. The data analyzed with the models are categorized according to the network traffic types and the attacks carried out on the network are detected. In WSN, attack detections made with learning models are performed with high accuracy percentages compared to classical detection methods. In this study, Blackhole, Flooding, and Selective Forwarding attacks, which are WSN network layer attacks, were created and implemented in Network Scenario (NS 2) simulation environment. The WSN-BFSF data set was obtained, and the obtained data set was made ready to be examined with learning models after the necessary preprocessing. The WSN-BFSF dataset consists of 312106 rows. The data set was examined with 4 different machine learning models Random Forest, Decision Tree, Naive Bayes, and Logistic Regression, and 8 different deep learning models Multilayer Perception (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), CNN-LSTM, LSTM-CNN, CNN-GRU, and GRU-CNN. The experimental results obtained with the models are presented in detail.**

***Index Terms*—Wireless sensor networks, DoS attacks, Intrusion detection, Deep learning, Big data, Machine learning, Classification**

## I. INTRODUCTION

The use of Wireless Sensor Networks (WSN) has increased with the developments in recent years and its importance in the sector has increased. It is a modular, user-friendly system that can be used in many areas in line with its purpose, offers practical solutions to the user, and does not require infrastructure [1-3]. WSN market volume is expected to reach 1.8 million dollars in 2024 [4, 5]. In short, WSN is a synchronized organization consisting of wireless service and sensors that provide the perception of the environment [1, 6]. It can be changed and integrated into the environment according to the conditions of the environment. WSNs differ from information systems in terms of the convenience it provides. Although information systems provide convenience to the user in daily life, they are troublesome compared to WSNs due to situations such as installation, sustainability, and maintenance. Although WSN works with a central system, it is not managed with absolute central management as much as information systems. Especially since it is auto-synchronous, it does not require separate installation for each node. With this feature, it is used with many technologies and systems such as the Internet of Things (IoT). Sensor network technology, which is used in every field today, attracts the attention of users at all levels when it is integrated with IoT technology as well as projects such as smart homes and smart cities. Its applications are seen in many different fields such as the military field, natural life, industry, health and daily life [7]. The security vulnerabilities of a system used in such a wide and diverse field are questioned by attackers. Although its security is ensured, sensor networks have some security problems depending on the environment and protocols they are used in. Being physically accessible harms the network and communication in malicious access. Software attacks are carried out internally and externally. External attacks are attacks on the closed network system by joining the network with a new node from outside. Generally, software attacks are called network attacks because they are carried out over the network. Internal attacks; It is carried out by taking the current working node in various ways and using it by the attacker. Attacks differ according to their purpose and the layers in which they are carried out. In this study, Blackhole, Flooding and Selective Forwarding attacks, which are network layer attacks, were carried out in the NS-2 simulation environment. After the attacks were carried out, all the traffic data were taken and the raw state of the WSN-BFSF data set was recorded. WSN-BFSF dataset consists of attack traffic data

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE. It is IEEE style to display support information, including sponsor and financial support acknowledgment, here and not in an acknowledgment section at the end of the article. For example, “This work was supported in part by the U.S. Department of Commerce under Grant BS123456.” The name of the corresponding author appears after the financial information, e.g. *(Corresponding author: M. Smith).* Here you may also indicate if authors contributed equally or if there are co-first authors.

Murat Dener is with the Information Security Engineering Department, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey (e-mail: muratdener@gazi.edu.tr)

Celil Okur is with the Information Security Engineering Department, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey (e-mail: celil.okur@gazi.edu.tr)

Samed Al is with the Information Security Engineering Department, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey (e-mail: samed.al1@gazi.edu.tr)

Abdullah Orman is with the Computer Technologies Department, Ankara Yıldırım Beyazıt University, Ankara, Turkey (e-mail: aorman@ybu.edu.tr)

(Blackhole, Flooding, Selective Forwarding) and Normal traffic data. The recorded raw data set is not used directly with the machine and deep learning models. For this reason, they were made usable in models by pre-processing them gradually. The data analyzed by learning models were classified according to network traffic types and the results were recorded.

Contributions of the study to the literature;

- Step-by-step explanation of how to do Blackhole, Flooding, Selective Forwarding attacks in a WSN simulation environment,
- Increase the limited number of data sets in the WSN field with the created WSN-BFSF data set,
- Explanation in detail of the stages of creating the data set to guide while creating new data sets,
- In the study, it is possible to rank the WSN-BFSF data set as examining both machine learning models and deep learning models with high accuracy results.

In the second part of the study, the related studies were examined by scanning the literature. In the third chapter, the working structure of the AODV protocol used in the study is mentioned. In the fourth chapter, the attack models performed in the study are mentioned. In the fifth chapter, the stages of the creation of the data set and its features are mentioned. In the sixth chapter, the application stages and the results are given. In the seventh chapter, the results of the study are given.

## II. Background and Related Works

Data sets are obtained as a result of both information systems and sensor network studies. Although there are network traffic data sets belonging to quite different information systems in the literature, it is not possible to say this about sensor networks. It was seen that there were also studies using the WSN data set when the literature was searched. However, in most of the studies, information such as the names, properties, and methods of obtaining the data sets was not included, and only the data set analysis results were included. In addition, the data set that is commonly used in different studies is the WSN-DS data set. Web of Science, Science Direct, Eric, and Elsevier databases were scanned during the literature review. Since data sets and machine or deep learning models are used together, especially in network security studies, the literature review has been carried out not only on the data set, but also on the studies on machine learning and deep learning in sensor networks. The results of the literature review are given in Table 1.

TABLE I
SUMMARY OF THE LITERATURE

| Reference | Author-Year | Method-Model | Name of the data set | Attack Type | Platform | Evaluation metrics and results |
|---|---|---|---|---|---|---|
| [8] | Almomani, Al-Kasasbeh, Al-Akhras<br>2016 | ANN<br>MLP<br>Three hidden Layer | WSN-DS | Blackhole<br>Grayhole<br>Scheduling<br>Flooding | NS-2 | Accuracy<br>Blackhole: 92.8%<br>Flooding: 99.4%<br>Scheduling: 92.2%<br>Grayhole: 75.6%<br>Normal Case: 99.8% |
| [9] | Ifzarne, Tabbaa, Hadi, Lamghari<br>2021 | SVM<br>Naive Bayes<br>Random Forest<br>Devision Tree | WSN-DS | Blackhole<br>Grayhole<br>Scheduling<br>Flooding | NS-2 | Accuracy<br>SVM 89%<br>Naive Bayes 94%<br>Random Forest 94%<br>Devision Tree 94%<br>Model 96% |
| [10] | Garcia, Garrigues, Pous<br>2016 | One Class SVM | Row sound data for 14 days period | Jamming<br>Selective Forwarding | Catalia 3.3 | TPR<br>At least 56%<br>FPR<br>Max 5% and 26% |
| [11] | Alrajeh, Khan, Lloret, Loo<br>2014 | ANN | ---------- | Flooding<br>Routing loop<br>Fake Channel reques | NS-2 | Accuracy<br>Flooding 98%<br>Routing loop 95%<br>Fake Channel request 55% |
| [12] | Otoum, Kantarcı, Mouftah<br>2019 | Machine Learning<br>Deep Learning | KDD’99 | DoS<br>R2L<br>U2L<br>Probe | ----- | Accuracy<br>Machine Learning 99.12%<br>Deep Learning 99.91% |
| [13] | Alshinina, Elleithy<br>2018 | SWSNM<br>CNN<br>Decision Tree<br>ANN<br>SVM | NSL-KDD<br>----------- | Malicious Node Detection | NS-2 | Accuracy<br>CNN 87%<br>SWSNM 86.5%<br>Decision Tree 81.5 % |

| | | | | | | |
|---|---|---|---|---|---|---|
| [14] | Pawar, Anuratha<br>2021 | LSTM | No name | Blackhole<br>Wormhole | Python | Accuracy<br>90.06% |
| [15] | Bahsi, Nomm, La Torre<br>2018 | Decision Tree | N-BaIoT | DDoS<br>Mirai-Bashlite-Beign | IoT network | Accuracy<br>99.97% |
| [16] | Sokolov, Iliev, Stoyanov<br>2019 | Deep Neural Networks | N-BaIoT and ML Repository | DDoS<br>Mirai-Bashlite | IoT network | Accuracy<br>83% |
| [17] | Nomm, Bahsi<br>2018 | Local Outlier Factor (LOF), One-Class SVM, Isolation Forest (IF) | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | Accuracy<br>99% |
| [18] | Anthi et. al.<br>2019 | Naive Bayes, Byessian Network, J48, ZeroR, OneR, Simple Logistic, SVM, Multi-Layer Perceptron, Random Forest | ---------- | Reconnaissance<br>DoS/DDoS<br>MITM<br>Replay Spoofing | IoT network | F- Measure<br>96.2% 90% 98% |
| [19] | Kumar, Lim<br>2019 | RandomForest, K-NN, Gaussian Naive Bayes | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | F1 Score<br>96% |
| [20] | Possebon et. al.<br>2019 | Ensamble-KNN, DT, MLP | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | Recall 99%<br>Precision 99%<br>Accuracy 99%<br>F1-Score 49% |
| [21] | Meidan ve diğerleri<br>2017 | GBM, Random Forrest, XGBoost | ----------- | IoT cihaz türü | IoT-PC network | Accuracy 99% |
| [22] | Bai, Liu, Zhang<br>2019 | CNN | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | Accuracy<br>99.57% |
| [23] | Karanja, Masube, Jeffrey<br>2019 | KNN, Naive Bayes, Random Forest | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | KNN 89%<br>Naive Bayes 80%<br>Random Forest 95% |
| [24] | Cvitic, Perakovic, Persa, Botica | KNN, SVM, DT, Random Forest And Artificial Neural Networks | MTC | DDoS | IoT network | Accuracy<br>100% |
| [25] | Liu, Wang, Li, Hao, Feng<br>2018 | CNN | N-BaIoT | Blackhole | OMNET++ | PDR 99.9%<br>Residual Energy 88.5% |
| [26] | Anwer, Farooq, Waseemullah<br>2021 | Support Vector Machine (SVM), Gradient Boosted Decision Trees and Random Forest | NSL-KDD | DoS<br>Malicious | IoT network | Accuracy<br>85.34% |
| [27] | Okur, Dener<br>2021 | Random Forest | WSN-DS | Blackhole<br>Grayhole<br>Scheduling<br>Flooding | NS-2 | Accuracy<br>99.72% |
| [28] | Okur, Dener<br>2020 | Decision Tree | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | Accuracy<br>99.95% |
| [29] | Okur, Dener<br>2022 | Random Forest | N-BaIoT | DDoS<br>Mirai-Bashlite | IoT network | Accuracy<br>99.92% |
| [30] | Warzali, Ahmad<br>2021 | Gboost Algorihtms-<br>KNN<br>LR<br>SVM<br>Decision Tree<br>LSTM<br>MLP | WSN-DS | Blackhole<br>Grayhole<br>Scheduling<br>Flooding | NS-2 | Accuracy<br>99.6%<br>F-1 Score<br>98.8%<br>FPR<br>0.4%<br>FNR<br>0.13% |
| [31] | Branitskiy, Kotenko<br>2016 | SVM | NSL-KDD<br>KDD'99 | DoS<br>Malicious | IoT network | Accuracy<br>KDD 99.96%<br>NSL-KDD 99.6% |

| | | | | | | |
|---|---|---|---|---|---|---|
| [32] | Fatani et. al.<br>2021 | CNN<br>SVM | CIC2017, NSL-KDD, BoT-IoT, ve KDD99 | DoS<br>Malicous | IoT computer network | Accuracy 98.97% |
| [33] | Otair et. al.<br>2022 | K-mean , SVM | NSL-KDD | DoS<br>Malicious | IoT network | Accuracy 98.97%<br>74.48% |
| [34] | Ahmad et. al.<br>2022 | RaNN<br>SQP<br>PSO | DS2OS, USNW-NB15 ve ToN_IoT | Malicious | IoT network | Accuracy 98.64%, 99.12% 99.57%. |
| [35] | Ahmed et. al.<br>2022 | CNN | NSL-KDD, CICIDS2017, and Bot-IoT | DoS<br>Malicious | IoT network | Recall 99.54%<br>Precision 99.53%<br>F 1 Score 99.53%<br>Accuracy 99.52% |
| [36] | Alzubi et al.<br>2022 | PSO<br>GWO | NSL KDD'99<br>UNSW-NB15 | DoS<br>Malicous | IoT Network | False alarm rate 4%<br>Detection accuracy 0.3% |
| [37] | Dahou et al.<br>2022 | CNN<br>RSA | KDDCup-99, NSL-KDD, CICIDS-2017, and BoT-IoT | DoS<br>Malicious | IoT Network | Accuracy 92.34%<br>Precision 94.33%<br>F1 Score 92.76%<br>Recall 92.34% |

Although different learning models have been used for attack detection in [8, 30] studies, it is seen that the data set and classification logic are the same. Almomani, Al-Kasasbeh and Al-Akhras [8] created the WSN-DS dataset, which is used in many studies in the field of WSN. In the NS-2 environment, a total of 5 different network traffic, four different attacks and one normal traffic, were examined with the WEKA tool. Ifzarne, Tabbaa, Hadi and Lamghari [9] developed detection methods in their study. At the same time, they used the results they obtained in the model they developed. Garcia, Garrigues and Pous [10] examined the attack and normal traffic in 14-day sound sensor data. Alrajeh, Khan, Lloret and Loo [11] stated that the data set they produced contained an energy consumption attack and they stated that they detected the attack with the ANN learning model. Otoum, Kantarcı, and Mouftah [12] developed two different algorithms with machine learning and deep learning models and compared the accuracy percentages and result times of these algorithms. Alshinina and Elleithy [13] developed a model for end-to-end safe traffic with unsupervised learning. They evaluated the results by examining the False Positive Rate (FPR), latency, efficiency and amount of energy usage. Pawar and Anuratha [14] conducted a study on the detection and prevention of Wormhole and Blackhole attacks. The LSTM model was used to detect the attack traffic. Thanks to the algorithm developed with this model, the optimum and non-attack path is determined.

Bahsi, Nomm and La Torre [15] made attack detection with the data set. In addition to attack detection, they reduced the IoT botnet attack traffic from 115 features to 3 features by applying feature reduction. They categorized the traffic by examining the data set with the decision tree model. They stated that the system they developed can be used in IDS/IPS by working in integration with the network. Sokolov, Iliev and Stoyanov [16] classified spam content and images with different models and compared their results with previous studies in the literature. Nomm and Bahsi [17] classified the malicious traffic by examining the parameters of IoT devices with unsupervised learning models. They applied the entropy operation for feature reduction. The SVM model with 3 features and the Isolation Forest model with 5 features gave the highest results in the classification process. Anthi et al. [18] developed an intrusion traffic detection model for IoT systems. In this 3-layer model, normal traffic is categorized in the first layer, malicious traffic is categorized in the second layer, and in the third layer, and it is categorized according to attack traffic type. Kumar and Lim [19] analyzed the IoT botnet attack dataset and detected the attack with 96% accuracy. They stated that the proposed model offers distributed and modular solutions for large network models, and the model was developed for the detection of IoT malicious network activities. Possebon et al. [20] combined different models with the meta-learning technique and examined their use together and separately. The results of the different algorithms developed when used together and the results when used separately were compared. Meidan et al. [21] analyzed IoT devices with deep learning models and classified them according to their brands and models. It has been stated that the model can be used to prevent unauthorized access. Bai, Liu and Zhang [22], especially in their study, analyzed the data set by transforming the data set into a 23x23 matrix using the CNN algorithm. They reduced the data set from 115 features to 23 features with the Z score tool. Karanja, Masube and Jeffrey [23] analyzed the data set with KNN, Random Forest, and Naive Bayes models. They converted the digital dataset into pictures in shades of gray. The method emphasizes the importance of low mathematical computations and independent platform features. Cvitic, Perakovic, Persa and Botica [24] examined MTC and HTC traffic, which is the traffic generated by IoT devices such as smart home devices. It has been stated that the DDoS traffic generated by SHIoT devices is detected with a 95% accuracy percentage using MTC. Liu, Wang, Li, Hao, and Feng [25] combined the effective referral model with the trust mechanism. With the developed model, it is stated that high performance is provided in topics such as network security, efficient energy use, and packet distribution rate. Anwer, Farooq and Waseemullah [26] analyzed the NSL-KDD dataset with the Random Forest algorithm and reached the highest accuracy rate of 85.34%. They have applied classification processes to attacks such as unauthorized remote access, DoS, remote super root authority, and port scanning. Okur and Dener [27] analyzed the WSN-DS data set, which has been accepted in the field of WSN, with 7 different machine learning models. They achieved the highest accuracy rate of 99.72% with the Random Forest model. Okur and Dener [28] examined the IoT botnet attack traffic with machine learning models and reached the highest accuracy rate of 99.95% with the Decision Tree algorithm. Okur and Dener [29] analyzed the N-BaIoT attack dataset with 23 different machine learning models and reached the highest accuracy rate of 99.92% with the Random Forest algorithm. Warzali and Ahmad [30] analyzed the WSN-DS dataset with different learning models and they classified those according to the type of attack. They also included the percentage results of the False Positive Rate and False Negative Rate. Branitskiy and Kotenko [31] mentioned that they developed a model with a hybrid approach. It is stated that there are different components in the proposed model. It is mentioned that the SVM learning model is also used with these components and the proposed model consists of several layers which can be used in detection mechanisms. Abdulaziz Fatani et al. [32] mentioned that a new feature selection and feature extraction method was developed in their study. CNN model was used for feature extraction. The swarm intelligence algorithm and Aquila optimizer (AQU) are used for feature selection. They also mentioned that they use the advantages of the swarm intelligence (SI) algorithm in the IDS system they developed. Four data sets, which are accepted in the literature, CIC2017, NSL-KDD, BoT-IoT, and KDD99, were used in the study. F1 score, accuracy, recall and precision results are shared for each data set. Otair et al. [33] stated that they developed a technique for attack detection in their studies. The importance of feature selection is mentioned in the study. Gray Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) were used as hybrids for feature selection. NSL-KDD data set was classified in the study. The classification was carried out with K-Means and Support Vector Machine (SVM) models. In the classification process, accuracy, detection rate, false alarm rate, number of features, and execution time values were examined. Ahmad et al. [34] mentioned that fast and reliable attack detection is done by examining the data obtained with the Industrial Internet of Things (IIOT) with the Deep Random Neural Network model. It has been mentioned that the RaNN model is an improved variation of the ANN model. It was stated that Particle Swarm Optimization (PSO) and Sequential Quadratic Programming (SQP) were used as hybrids to detect with a high percentage of accuracy from the RaNN model. The binary and multiclass classification was made in the study. Three data sets, DS2OS, USNW-NB15 and ToN_IoT, were used in the study. Ahmed et al. [35] used the CNN model for feature extraction in their study. Gorilla Troops Optimizer (GTO) algorithm was modified and used for the feature selection process. It has been stated that the detection accuracy percentage has increased with these innovations. In the study, three data sets, NSL-KDD, CICIDS2017, and Bot-IoT, were classified with the KNN model. The results of the classification process were compared in terms of accuracy, precision, F1 score and recall. Alzubi et al. [36] used the hybrid method for intrusion detection by combining Gray Wolf Optimization and Particle Swarm Optimization algorithms in their study. NSL KDD'99 and UNSW-NB15 datasets were used in the study. According to the normal results with this method, the accuracy results increased by 0.3% to 12%, the detection rate increased by 2% to 12%, the false alarm rate decreased by 4% to 43%, the number of features decreased by approximately 31% to 75%, and the execution time value decreased by 14% to 22%. Dahou et al. [37] stated that they used learning and metaheuristic (MH) optimization algorithms for feature extraction and selection in their study. It is stated that the CNN algorithm is used as the basis of feature extraction. They performed feature selection with the Reptile Search Algorithm (RSA) they developed. It is stated that data processing is carried out by selecting the most important features with the RSA mechanism. KDDCup-99, NSL-KDD, CICIDS-2017, and BoT-IoT datasets were examined in the study.

It was seen that the common aspect of the studies numbered 32, 33, 35, and 37 focused on the feature extraction process. Although different approaches have been used for feature selection, it has been seen that GWO and PSO optimization methods are used in common in related studies.

When the studies are examined, as seen in Table 1, in most of the studies in the field of WSN, the WSN-DS data set was used. This is due to the limited number of data sets in the literature. In other studies where the WSN-DS dataset was not used, descriptive and clear information about the analyzed datasets was not provided and the datasets were not shared in the literature.

- The contribution of this study to the literature is that with the obtained data set, the limited number of data sets that can be used in the WSN field has been increased.
- WSN-BFSF dataset includes Blackhole, Flooding and Selective Forwarding attacks. Unlike the WSN-DS dataset, it also contains data belonging to the Selective Forwarding attack. In this respect, another contribution of the study is that it will contribute to the work of developers who want to analyze Selective Forwarding attack data.
- When the size of the data set (anonymous) obtained in the study of Garcia Garrigues and Pous is examined, it is much less than that obtained in this study. In deep learning models, it is desired that the size of the data set examined is large. In this respect, the data set obtained is suitable for the developers to examine with deep learning models.
- It is seen that the WSN-DS data set has been examined with

almost all learning models in different studies. New data sets are needed to examine with models in the literature. In this respect, it is thought that it will contribute to the literature.

## III. AODV Protocol Structure

In the application part of the study, network traffic routing was carried out with the AODV (Adhoc On Demand Vector) protocol, which is one of the MANET (Mobile Ad Hoc Networks) protocols. MANET (Mobile Ad Hoc Networks) is a technology used in self-synchronizing wireless networks, which provides communication between nodes and does not require a fixed infrastructure and backbone. The AODV protocol is a reactive and on-demand protocol. Since it is a reactive protocol, it creates a routing table if needed. To create this table, which node sends direction request packets (RREQ) and how many steps away to create alternative routes to the destination [38] are considered. The direction request package; contains information such as the address of the source node, the address of the destination node, and the number of passed nodes. This packet is sent from the source node to all nodes. This information is recorded in the direction table of the source node and stored for a certain period to be used in the next communication. This information in the table creates an extra load for the limited memory of the nodes. Determining the route for the destination node from this information recorded by the source node, the most up-to-date sequence number of the request packets and the least number of nodes to be traversed until reaching the destination is preferred. After the route and route are determined for communication, the real communication starts after the hello packet, which can be called the final verification, is sent and its accuracy is confirmed before the communication starts. If there are breaks in the communication path created, an error message is issued, and the routing operations are re-established. The manipulation of request packets; causes traffic delay creating a density in the network, which causes conflicts-packet drops due to excessive traffic formation resulting in nodes consuming their energy in a short time due to heavy traffic.

## IV. Attack Models

Blackhole, Flooding, and Selective Forwarding, attacks carried out in the study are examples of network layer attacks as well as DDoS attacks. However, the most important feature that distinguishes DoS and DDoS attacks, as the name suggests, is a denial of service (DoS) and distributed denial of service (DDoS) attacks [31]. The purpose of both is to make the goal inaccessible and to prevent users from reaching the goal. In the first, a single attacker performs this attack, while in the second, more than one attacker performs it. In recent years, the use of DDoS attacks has increased with malicious software and manipulation of devices that use this software with or without notice. Although DDoS is more synchronous and troublesome than DoS attacks, it produces more effective and harmful results [19]. The following attack types are used in both DoS attacks and DDoS attacks. Although how the attacks are carried out is different, their purpose is the same [32]. Examples of DoS attacks in WSN are Blackhole, Flooding, Selective Forwarding, and Grayhole attacks.

Although all of the attack types in the study are DDOS attack types, there are significant differences in terms of the way they work. As a result of the literature review in the second part, it was seen that although there are many attacks traffic creation and attack detection studies in the literature, there is not enough information about which attacks occur and how they are implemented. Generally, it has been seen that theoretical information about attacks is included in the literature. In this study, unlike other studies, not only theoretical information is given about Blackhole, Flooding, and Selective Forwarding attacks, but also how the attacks occur is explained step by step, and pseudo codes are also shared. In addition, as seen in the literature review section, there is a limited number of data sets in the WSN field. It is seen that a limited number and type of attacks were carried out in each of these datasets. In this study, while determining which attacks will be carried out, attention has been paid to the fact that they are both DDOS attacks and that the selected attacks have not been done in the same study before. In other words, the reason why these types of attacks are preferred in the study is that the attacks carried out are both DDOS attacks and although each of them has been applied in different studies, not all of them have been used at the same time. In this respect, it is thought that it will contribute to the literature and the developers in this field.

### *A. Blackhole Attacks*

Blackhole attacks are attacks made to break the communication in the network totally or partially by one or several nodes by attracting them in terms of parameters such as the node trust value, remaining energy amount, and the shortest path. A blackhole attack is a type of DDoS attack. The working algorithm of the blackhole attack model is shown in Figure 1.

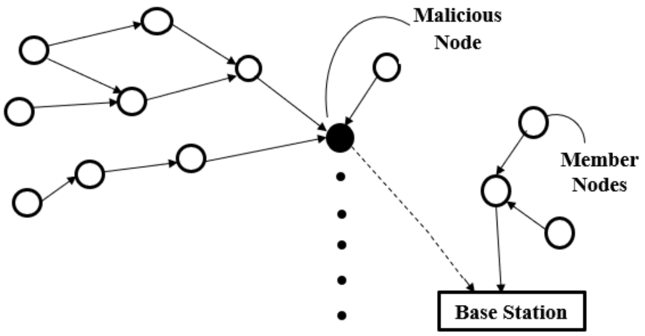
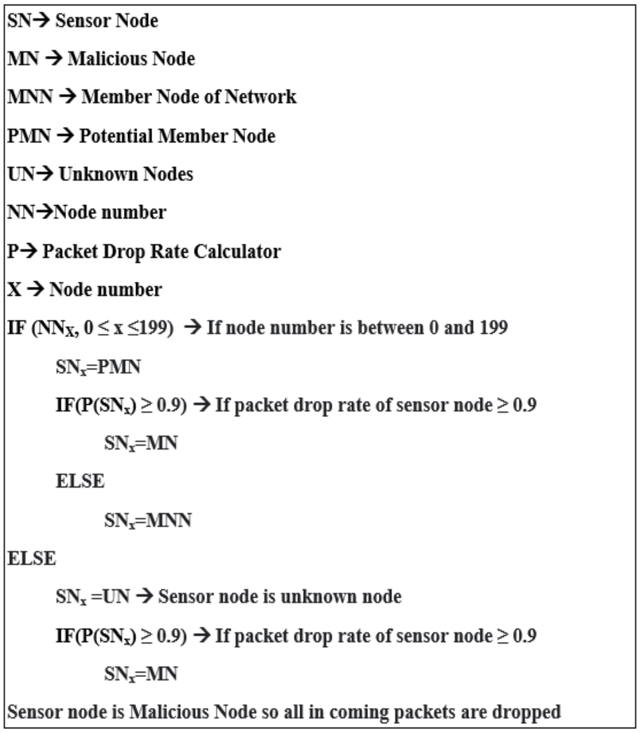


**Fig. 1.** Blackhole attack model.

Malicious nodes hide by responding to RREQ request packets as if they were ideal and normal nodes. In communications, the attacker node drops the incoming packets instead of transmitting the packet sent to the target, causing the communication to be completely or partially broken [2]. Malicious node reduces the packet traffic according to the packet content, the number of the sending node, the number of the destination node, the communication time, and the specified groups.

Pseudo codes describing the detection of blackhole attacks are shown in Figure 2.



```
SN→ Sensor Node
MN → Malicious Node
MNN → Member Node of Network
PMN → Potential Member Node
UN→ Unknown Nodes
NN→Node number
P→ Packet Drop Rate Calculator
X → Node number
IF ($NN_X$, $0 \le x \le 199$)  → If node number is between 0 and 199
    $SN_x$=PMN
    IF($P(SN_x) \ge 0.9$) → If packet drop rate of sensor node ≥ 0.9
        $SN_x$=MN
    ELSE
        $SN_x$=MNN
ELSE
    $SN_x$ =UN → Sensor node is unknown node
    IF($P(SN_x) \ge 0.9$) → If packet drop rate of sensor node ≥ 0.9
        $SN_x$=MN
Sensor node is Malicious Node so all in coming packets are dropped
```

**Fig. 2.** Pseudo code of blackhole attack.

*B. Flooding Attacks*

In the installation of the WSN, request packets are sent from the node to all other nodes to provide communication synchronization between the sensors. On the other hand, nodes that are newly included in the network need request packets before communication starts. In light of the information obtained from these packages, tables are created and the direct route is determined according to these tables. However, these packets impose a heavy load on the network [41]. The working algorithm of the flooding attack model is shown in Figure 3.

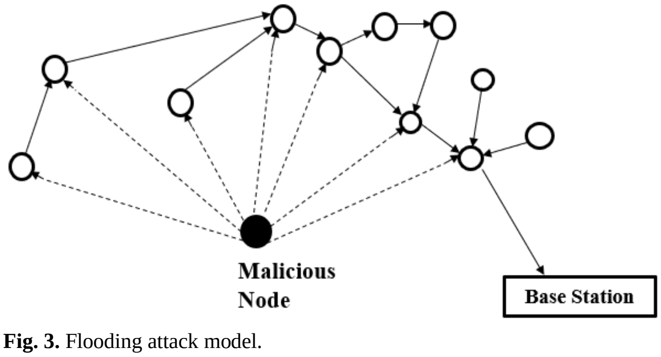
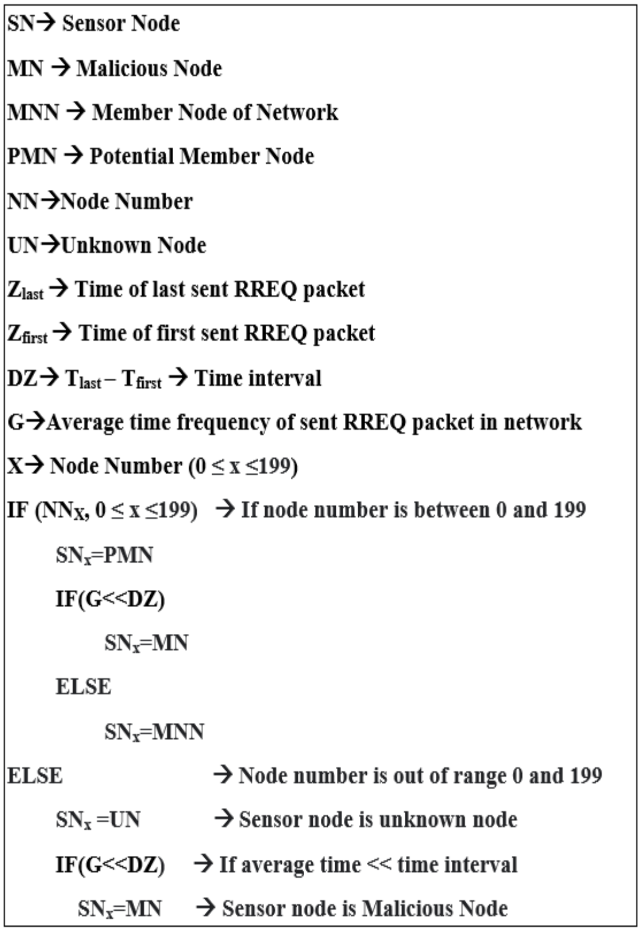


**Fig. 3.** Flooding attack model.

Even when it is used for the purpose, it becomes a load on the network, and if this situation is continuous throughout the communication period; it causes delays-drops-slowdowns in real communication between nodes, large increases in network traffic density and depletion of sensor energies in a very short time [41]. Unlike normal nodes, the attacker node aims to consume the resources of the network and make the network inaccessible by sending the request packets at random times or continuously. The working model for the flooding attack is shown in Figure 3. In Figure 4, pseudo codes explain how to distinguish between the node performing the flooding attack and the normal node of the network.



```
SN→ Sensor Node
MN → Malicious Node
MNN → Member Node of Network
PMN → Potential Member Node
NN→Node Number
UN→Unknown Node
$Z_{last}$ → Time of last sent RREQ packet
$Z_{first}$ → Time of first sent RREQ packet
DZ→ $T_{last} - T_{first}$ → Time interval
G→Average time frequency of sent RREQ packet in network
X→ Node Number ($0 \le x \le 199$)
IF ($NN_X$, $0 \le x \le 199$)   → If node number is between 0 and 199
    $SN_x$=PMN
    IF(G<<DZ)
        $SN_x$=MN
    ELSE
        $SN_x$=MNN
ELSE                    → Node number is out of range 0 and 199
    $SN_x$ =UN           → Sensor node is unknown node
    IF(G<<DZ)    → If average time << time interval
      $SN_x$=MN     → Sensor node is Malicious Node
```

**Fig. 4.** Pseudo code of flooding attack.

*C. Selective forwarding attacks*

It can simply be defined as a selectively permeable attack type. Selective Forwarding attack is a type of attack, which sends or drops some part of the packets based on values such as the content of the packets, the sending time, the number of the sending node and the number of target nodes instead of forwarding all the incoming packets. This means a DoS attack for traffic-affected (source-destination) nodes [34]. The working model of the Selective Forwarding attack is shown in Figure 5.

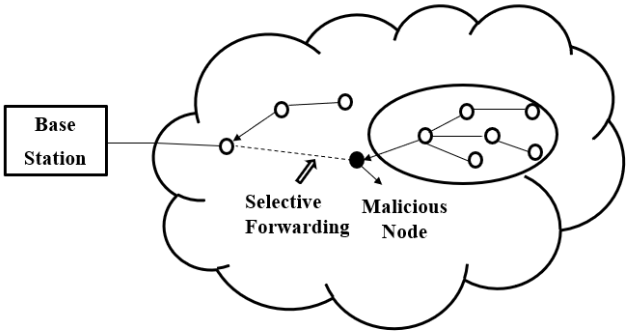
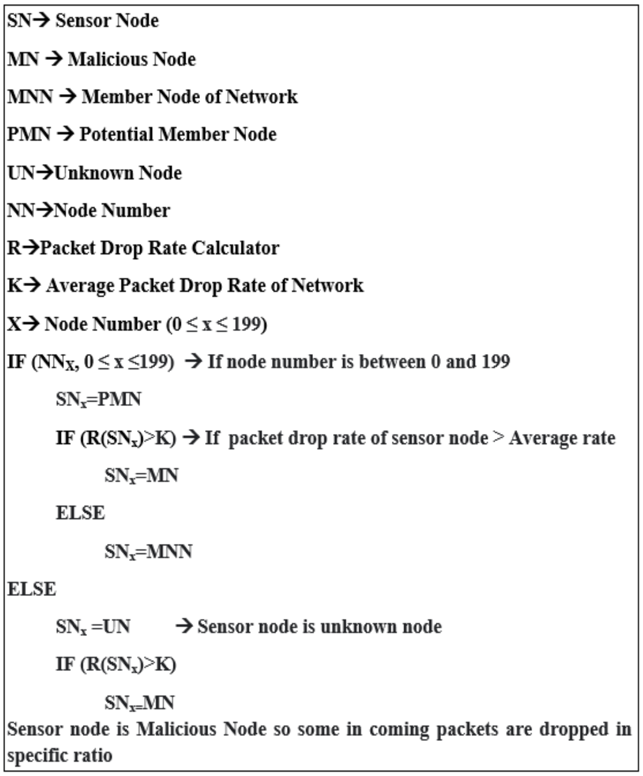


**Fig. 5.** Selective forwarding attack model.

Malicious nodes that perform Selective Forwarding attacks take part in multi-step communications and act as intermediate nodes. Another purpose of Selective Forwarding attacks is to delay the forwarding of incoming packets. The goal here is to confuse network routing by not sending packets on time but sending them later on. Packets sent to the target node by waiting on the malicious node cause information pollution in the target node. Since the transmitted data is incomplete, it does not make sense to the user. The main purpose of the attack is to violate data integrity.

Pseudo codes for Selective Forwarding attack detection are shown in Figure 6.



```
SN→ Sensor Node
MN → Malicious Node
MNN → Member Node of Network
PMN → Potential Member Node
UN→Unknown Node
NN→Node Number
R→Packet Drop Rate Calculator
K→ Average Packet Drop Rate of Network
X→ Node Number (0 ≤ x ≤ 199)
IF (NN_X, 0 ≤ x ≤199)  → If node number is between 0 and 199
    SN_x=PMN
    IF (R(SN_x)>K) → If  packet drop rate of sensor node > Average rate
        SN_x=MN
    ELSE
        SN_x=MNN
ELSE
    SN_x =UN       → Sensor node is unknown node
    IF (R(SN_x)>K)
        SN_x=MN
Sensor node is Malicious Node so some in coming packets are dropped in specific ratio
```

**Fig. 6.** Pseudo codes for selective forwarding attack.

## V. WSN-BFSF DATASET DESCRIPTION AND CREATION

A data set with 16 features and 312 106 rows was obtained when the raw data set was prepared after the necessary pre-processing was performed. The data set consists of 4 different traffic Blackhole, Flooding, Selective Forwarding attack traffic and Normal traffic. The characteristics and explanation of the obtained data set are given in Table 2. The dataset has been uploaded to the Kaggle.com website. Dataset download link:https://www.kaggle.com/datasets/celilokur/wsnbfsfdataset

TABLE II
FEATURE DESCRIPTION OF THE WSN-BFSF DATASET

| No | List of 16 features used in the experiment | Explanation of the abbreviations |
|---|---|---|
| 1 | Event | It contains information about the operation performed on the traffic. It is represented by the value 1 for s (sending), 2 for r(receiving), 3 for f (forwarding), 4 for d (dropping), and 5 for N (Energy information). |
| 2 | Time | The time of the event performed in the row |
| 3 | S_Node | Source node number |
| 4 | Node_id | The node number of the relevant node |
| 5 | Rest_Energy | The remaining energy of the relevant node |
| 6 | Mac_Type_Pck | MAC type of the packet |
| 7 | Source_IP_Port | Port number of the source node |
| 8 | Packet_Size | Forwarded packet size |
| 9 | TTL | The lifetime of the forwarded traffic in the network |
| 10 | Hop_Count | Number of nodes passed |
| 11 | Broadcast_ID | The ID number of the broadcast packets |
| 12 | Dest_Node_Num | ID of the target node |
| 13 | Dest_Seq_Num | Sequence number of the traffic forwarded to the destination |
| 14 | Src_Node_ID | Source node ID number |
| 15 | Src_Seq_Num | Source sequence number of traffic forwarded to destination |
| 16 | Class | Type of classified network traffic |

The first of the features, the Event feature, verbally expresses which operation (receiving, sending, dropping, forwarding) takes place in that line. The Time property specifies the time of occurrence of the event specified in the line. The S_Node property represents the node number of the source node. This property provides information about the node generating the traffic. The node_id property gives information about the node in the row where the event occurred. Rest_Energy property, on the other hand, shows the remaining energy amount of the node the line is talking about. The Mac_Type_Pck property provides information about the mac variant of the package. It takes values between 0-800. The Source_IP_Port property provides information about the port number of the source node. The Packet_Size property provides information about the size of packets transmitted between nodes. The TTL feature occupies the network when the packets between the nodes enter the loop while transmitting. In order to prevent such situations, unnecessary traffic is prevented by determining the number of nodes to which the packet can be transmitted. Thus, packets are not transmitted indefinitely, and the packet is dropped when it reaches the TTL value. Hop_Count property indicates how many nodes the packet has reached and transmitted until the specified line. The Broadcast_ID property is a property usually found on lines that express packet traffic transmitted to the entire network, such as request packets. Indicates the number of the broadcast package. The Dest_Node_Num property specifies the number of the destination node to which the packet (traffic) will be forwarded. The Dest_Seq_Num attribute represents the packet sequence number that indicates the freshness of the data in the WSN. The sequence number of the transmitted packets provides information about the up-to-dateness of the data. The Src_Node_ID property provides information about the source node generating the traffic. The Src_Seq_Num property sends the data to be forwarded to the destination in a sequence. In this respect, the order in which the data is transmitted is important. The 16th feature (last) in the dataset specifies the class of traffic.

While the data set is examined with supervised learning in the literature and this study, it is important for the traffic class label learning to be correct. The distribution of the data set according to the traffic class is shown in Table 3.

TABLE III
WSN-BFSF DATA DISTRIBUTION

| Class | Row distribution | Percentage |
|---|---|---|
| Normal | 262 851 | 84.22 |
| Flooding | 29 844 | 9.56 |
| Blackhole | 11 766 | 3.77 |
| Forwarding | 7 645 | 2.45 |
| Total | 312 106 | 100 |

The information about the data set studies made in the field of WSN in the studies included in the literature review section is shown in Table 4.

TABLE IV
DATA SETS IN THE FIELD OF WSN

| Name of the Data Set | Type pof Attack | Number of Samples in the Data Set | Size of the Data Set | Number of Features of the Data Set | Environment in which the Data Set was Obtained |
|---|---|---|---|---|---|
| No name | Jamming<br>Selective Forwarding | 5 344 | ------ | 8 | Real Environment / Castalia 3.3 |
| WSN-DS | Blackhole<br>Flooding<br>Grayhole<br>Scheduling | 374 661 | 25.3 MB | 23 | NS-2 |
| WSN-BFSF | Blackhole<br>Flooding<br>Selective Forwarding | 312 106 | 24.7 MB | 16 | NS-2 |

Before the study was carried out, it was aimed to be different from the existing studies and to contribute to the literature by scanning the literature. For this reason, the feature of the attacks preferred and carried out in the study is that they were not carried out at the same time in the literature. As can be seen in Table 5, although the attacks carried out in this study were carried out in different studies, each of which was scattered in the literature, all of them were carried out at the same time only in this study. In addition, in this study, contrary to many studies in the literature, the theoretical and application stages of attacks are explained in detail.

## VI. Experiments and Results

### A. Steps of Creating the Data Set

In the study, first of all, the Virtual Box platform was built on the Windows operating system. Ubuntu 18.04 virtual machine has been installed on the Virtual Box platform. While defining the virtual machine features, considering the complexity of the scenario to be run in the simulation environment, 4 GB of ram and 100 GB of memory were defined for the virtual machine. NS-2 simulation environment is loaded on the Ubuntu operating system with related codes. The command entry process is carried out with the administrator role on the Ubuntu terminal. After the NS-2 has completed the installation process, checksum commands are used to determine whether the installation was successful. After the installation of the NS-2 platform, on which the scenario will run, was completed, the preparation phase of the scenario was started. The scenario is designed to have a total of 4 different network traffic, namely Blackhole, Flooding, Selective Forwarding and Normal. Although the scenario was originally designed with the NSG 2.1 tool, all of the network and node-related changes were made through the .tcl file in the later stages. While creating the scenario, the values given in Table 5 have been especially determined to achieve the desired size of the attack and normal data traffic and to ensure that the scenario works flawlessly in the NS-2 environment.

TABLE V
SCENARIO PARAMETERS

| Network Parameters | Values |
|---|---|
| Number of nodes | 200 |
| Time | 1600 minutes |
| Positioning Area | 2000x1000 |
| Package Size | 512 |
| Package Header | 25 |
| Protocol Used | AODV |
| Maximum Transfer Distance | 500m |
| Mac Protocol | CSMA |
| Starting Energy | 600J |
| Harmful Traffic Density | 16% |

There is a direct relationship between the operating time of the nodes and the initial energy amount. As the scenario run time increases, energy use increases. In addition, the energy value was determined by considering the malicious nodes and the generating-transmitting nodes of the traffic. On the other hand, working time is also important for the data set to reach the desired size. After the necessary adjustments were made in the energy, the processes related to network and communication were started. Since a 200-node network is created, an error message is frequently received if the above-mentioned parameters are not given appropriate values while running in the simulation environment. In the scenario, the Selective Forwarding attack was first performed. A selective Forwarding attack is an attack on data integrity. A forwarding attack is an example of a DoS attack. The node determined as the attacker sometimes transmits the incoming packets to the next node and sometimes does not transmit it, damaging the communication. The Blackhole attack type, which is the second attack carried out in the study, is an example of the internal attack type. A blackhole attack is a DoS attack. The attack is aimed at violating accessibility and integrity, which are among the main topics of information security. In this attack, malicious nodes reduce all incoming traffic to be transmitted, preventing communication between the relevant nodes. Unlike the Selective Forwarding attack, it drops all incoming traffic packets. The flooding attack, which is the last attack in the study, is a DDoS attack and this attack is carried out to violate accessibility and usability, which are the basic elements of information security. Although RREQ packets are used while performing the attack, it does not have any communication purpose. In terms of traffic density, RREQ packets occupy the network, delaying normal communication or reducing packets. While performing this attack, only request packets were sent without normal communication between the relevant nodes. In addition to attack traffic, normal traffic is also needed in the data set. While the attack is carried out in the daily applications of WSN, normal network traffic continues at the same time. In addition, learning models need normal network traffic data while categorizing the data. Therefore, normal traffic is also included in the study. Since the dataset includes Blackhole, Flooding, and Selective Forwarding attack traffic, the name of the dataset was created by listing the initials of the attacks in alphabetical order. The created data set is named as Wireless Sensor Network Blackhole, Flooding, Selective Forwarding (WSN-BFSF).

To have a balanced distribution in the WSN-BFSF data set and to ensure that the determined scenario continues to work in the simulation environment, the scenario was realized by selecting the specified times and nodes and especially. The design phase of the scenario was completed and the next phase, the NS-2 platform, started to run. The trace file produced after the script is run contains detailed information about network traffic, network packets and nodes. The file size varies depending on variables such as the number of nodes used, network traffic density, network uptime, and traffic type. The trace file produced in the scenario is the raw data set of the application. The file consists of rows and properties (columns). The number of features (columns) varies according to the content of the rows. There are verbal abbreviations and the values of the abbreviations in the file. In general, the content of the trace file is as follows:  the node consists of three types of lines as being energy line, request traffic line, and normal traffic line. When the rows in the raw form of the WSN-BFSF dataset are examined, it is seen that most of the rows are composed of node energy rows. It is seen that the node energy lines contain only the status, time, node number and node energy value information. This information is not sufficient for the analysis and classification of traffic. In addition, the feature numbers of the three types of lines mentioned above are different. Therefore, column synchronization is applied. During column synchronization, the missing columns of the rows with less than the value to be synchronized are equalized by giving a zero value. Since the node energy rows have fewer columns, equating is achieved by giving the value 0 to the missing columns except the status, time, node number and energy value columns. However, since these rows consist of 4 columns, when the remaining column value is 0, it is seen that most of the data set consists of 0. In this case, the data set was tried with various learning models, but it was seen that the models failed to reach an objective and precise result. When the node energy lines are removed from the data set, it has been observed that there will be no information deficiency when examining with learning models, since the information in these lines (state information, node number, energy amount) is also found in other lines. While synchronizing the columns in the remaining rows (normal traffic row, request traffic row), all the columns in the two rows were taken and examined. While examining the common columns in both rows, the columns whose values did not change were removed. The common columns have been simplified by such operations. On the other hand, the columns that are in the request line but not in the normal traffic line are equated with a value of 0 to the missing columns in other studies in the literature. In addition, when different types of rows are brought together, since similar columns are in different rows, the places of the relevant features have been changed to ensure that they are in the same column. After the preprocessing in the WSN-BFSF dataset, it is ready to be analyzed with learning models. The ready-to-use dataset consists of 16 feature sizes and 312 106 rows.

### *B. Machine Learning and Deep Learning Analysis*

In the study, a new dataset was created for the detection of DoS attacks in WSN. The created dataset is named WSN-BFSF. The WSN-BFSF dataset consists of three different DoS attacks and normal traffic samples: Blackhole, Flooding, and Selective Forwarding. In this section, the traffic obtained is analyzed with the machine and deep learning models and the results are shown below.

*1) Machine Learning and Deep Learning Algorithms:* Twelve different algorithms mentioned below were used to classify DoS attacks on the dataset created in the study. This section provides basic information about these algorithms.

*1.1) Decision Tree:* It is a supervised machine learning algorithm that is frequently used in classification and regression applications. In the decision tree algorithm, each node represents an attribute, while the branches connect the nodes to show the classification conditions. The variable with the highest information gain creates the root node and initiates the branching.

*1.2) Random Forest:* The Random Forest algorithm consists of evaluating the results of multiple decision trees together. The final decision is made by majority voting on the results obtained from the decision trees. It is a machine learning algorithm based on extreme learning. As the number of trees in the algorithm increases, the accuracy performance of the algorithm increases.

*1.3) Naïve Bayes:* It is a supervised machine learning algorithm based on Bayes' theorem. It evaluates each feature independently of the other and accepts that it contributes equally to the calculation. According to Naïve Bayes, the probability of occurrence of each feature occurs independently of other features. It is a probabilistic classifier that provides high scalability.

*1.4) Logistic Regression:* It is a supervised machine learning algorithm used for classification and regression problems. It estimates the targeted dependent variable based on the independent variables. The logistic curve is produced by positioning the outputs obtained with the sigmoid function between 0 and 1.

*1.5) Multilayer Perceptron (MLP):* MLP is a feed-forward neural network with three layers, the input layer, the hidden layer, and the output layer. The supervised trained MLP algorithm requires a large number of labeled data. It uses backpropagation in the classification process. The output layer must contain the number of neurons equal to the number of classes.

*1.6) Convolutional Neural Network (CNN):* Convolutional neural networks are specialized types of neural networks. In convolutional networks, the mathematical convolution operation is run. In the convolution operation, a filter is applied to an input. It's a linear operation that involves multiplying an input value with a set of weights. CNN consists of Convolution Layer, Activation Function, Pooling Layer and Fully Connected Layer. Convolutional and pooling layers perform the feature extraction task. In the Convolution layer, linear operations including a series of mathematical calculations such as the convolution operation are performed. As a result of this layer, feature maps are obtained. The pooling layer is also referred to as the down-sampling layer. It is aimed to reduce the computational cost and prevent overfitting by reducing the size of feature maps in the pooling layer. In the Activation layer, non-linear functions such as ReLu, sigmoid or tanh are run. In the Fully Connected Layer, data classification is performed. In this layer, a bias value is added after the input values are multiplied by a weight matrix. At the output of the layer, the probability that the inputs belong to each class is calculated. In addition, in the Flattening process, feature maps are transformed into 1D dimensional arrays before moving to the Fully Connected Layer.

*1.7) Long Short-Term Memory (LSTM):* LSTM is a type of Recurrent Neural Network (RNN). It is designed to overcome the vanishing gradient problem encountered by standard RNNs. LSTM, as the name suggests, is a deep learning algorithm capable of learning long-term dependencies. Sequential data is kept in memories. Thanks to these capabilities, it is frequently used in algorithms such as language processing, video processing and speech recognition. Unlike standard feedforward neural networks such as LSTM, CNN and MLP, it has feedback connections. LSTM consists of memory blocks called cells. These memory blocks are the core component of LSTM. In addition, the cell structure is divided into three sections: Forget gate, Input gate and Output gate. The input gate and output gate represent the input and output of the data at time t. Forget gate, on the other hand, decides on the forgetten information by comparing the instant data entry with the previous data state.

*1.8) Gated Recurrent Unit (GRU):* It is a type of feedback neural network designed to solve the gradient loss problem in recurrent neural networks (RNNs). It has similar structures to the LSTM algorithm. GRU cells have two gates an upgraded gate and a reset gate. The upgrade gate is the gate that decides how much of the historical information obtained up to a certain point will be transferred forward. The reset gate is the gate that decides how much of the information should be removed from memory at a given moment.

*1.9) Hibrit Deep Learning Algorithms:* In this study, four different hybrid deep learning algorithms, CNN-LSTM, LSTM-CNN, CNN-GRU and GRU-CNN, were used to detect DoS attacks on the generated dataset. In hybrid deep learning algorithms, the above-mentioned algorithms are applied sequentially to the data set, and classification is performed. The basic framework and procedure of hybrid deep learning approaches can be seen from in our study [42].

*2) Data Processing:* After the raw data set extraction and transformation processes, the WSN-BFSF data set was made ready for analysis with models. Three-stage preprocessing was carried out before the data set was examined with models and the results were produced. In the first stage, verbal expressions (Blackhole, Flooding, Selective Forwarding and Normal) in the last column of the data set, which indicate which traffic type the relevant row belongs to, were converted into numerical values. This column with verbal expressions is called the traffic class column. These verbal expressions in the class column are called the labels of the traffic in the related row. These labels are important when classifying with models. Especially in supervised learning, the classification process is carried out according to the label. However, there are verbal tags such as Blackhole, Flooding, Selective Forwarding and Normal in the traffic class column. These statements are meaningless for models. Therefore, numerical transformations were applied to these verbal labels. Table 6 includes the results of the digital transformation (Label Encoding).

TABLE VI
NUMERICAL TRANSFORMATION PROCESS

| Value | Type of Traffic |
|---|---|
| 0 | Blackhole |
| 1 | Flooding |
| 2 | Selective Forwarding |
| 3 | Normal |

With the One-Hot Encoding process, categorical values in the data set are assigned to numerical values. This process is called Label Encoding. The label coding (Label Encoding) stage was carried out in the Google Collab. an environment with the Label Encoding method from the Scikit-Learn library. The verbal (string) values in the Class column have been converted to numeric (integer) values. The class label of the Blackhole attack is given a value of 0, the class label of the Flooding attack is given a value of 1, the class label of the Selective Forwarding attack is given a value of 2, and the class label of Normal traffic is given a value of 3.

In the second step, feature reduction is performed. When experiments were made with the data set made ready for the use of the models, it was seen that some of the features took the same value in all rows and therefore did not contribute to the result. In addition, although there are different values in some columns, it is seen that error messages are received when examined with models, since most of them consist of 0 values. Therefore, the data set is reduced to 16 columns (features) by applying filtering to some columns of the data set. While selecting the relevant features, first of all, the data set was defined in a variable in the Google Collab. environment. After the definition, the feature selection process was carried out with the selection method over the variable.

After the feature selection, the third step is the normalization process of the data set. The purpose of the normalization process is to place these values between 0 and 1, according to their values, if the range of values taken by the relevant features of different rows with the same feature is large. A large range of values causes the result to be determined by large values. However, the same feature of each line contributes to the result at different rates. However, the fact that rows with large values change the result unevenly affects the correct detection rate. Therefore, the normalization process limits the effect on the result by fitting large differences between the values of the same feature in each row within a certain range. The normalization process shown in Equation 1 is performed.

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

In Equation 1, x represents the original value x′ normalized value and mean and standard deviation values, respectively. In this study, the normalization process of the features was carried out by importing the normalizer tool into the MLib library. The normalization process has been carried out using the normalizer method in the use of the command and the variable to which the data set is assigned.

*3) Analysis Procedures:* The correlation between features in the dataset is shown in Figure 7 using the Pearson Correlation Coefficient.

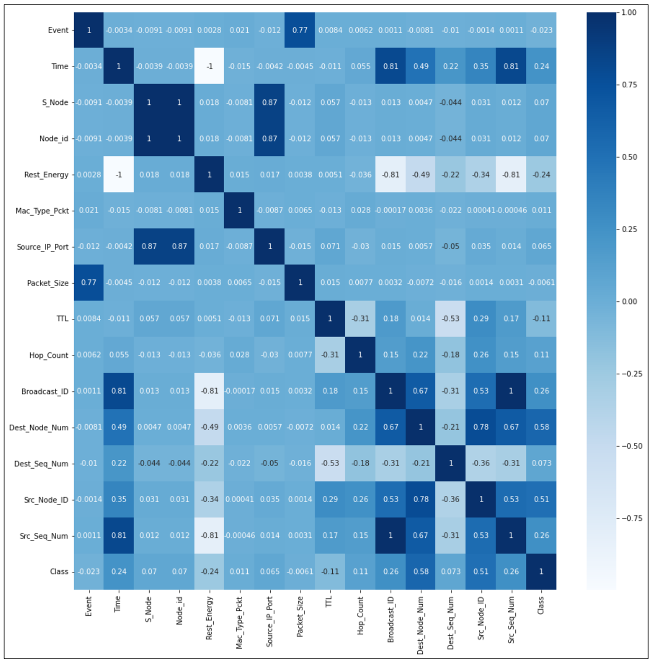


**Fig. 7.** Pearson correlation matrix results for the WSN-BFSF dataset.

In Figure 7, feature analysis showing the relationships of each feature in the WSN-BFSF dataset with other features in the dataset is presented. The Pearson Correlation Matrix gives the degree of relationships between features. In these relationships, 1 represents the highest relationship and -1 represents the lowest relationship. When looking at the diagonal axis in Figure 7, since the same features intersect on this axis, the relationship values have the highest value of 1. Relationship values are distributed between +1 and -1 value ranges. According to the relationship values they received, the relationship values from dark blue to white were represented by colors. If the correlation value is 1, it is shown in dark blue color, if the relationship value is -1, it is shown in white color. In this section, the classification performances of the dataset created for DoS detection are compared using various machine learning and deep learning algorithms. The classification process is summarized in Figure 8.

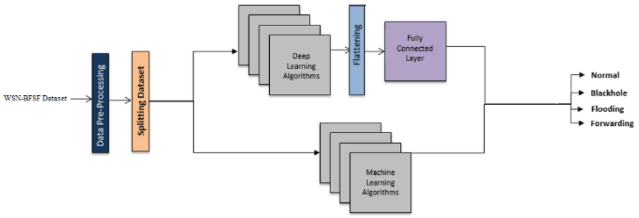


**Fig. 8**. Machine/deep learning flowchart.

As seen in Figure 8, the ready-to-use data set was preprocessed (labeling, feature selection and normalization). In the next step, the data set is divided into two test and training sets. The training part of the separated data set was used in the training phase of 8 different deep learning and 4 different machine learning models. The test set was used to determine whether the learning of the models was successful or not. The studies were carried out using the Pyspark tool, which provides Python programming language support on the Apache Spark big data platform in the Google Colab environment. For machine learning and deep learning algorithms, Scikit-Learn and Keras libraries, which are included in the PySpark MLib library, were used, respectively. The proposed method was compared with 12 different machine learning and deep learning algorithms, evaluations were made and the results were interpreted.

The most commonly used parameters in the literature, such as accuracy, precision, F-score, Recall, ROC, and Precision-Recall curves, were used to evaluate the results. These parameters are derived from the error matrix data. The basic elements of the error matrix are true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN). TP represents the number of instances correctly classified as an attack. TN represents the number of samples correctly classified as normal. FP refers to the misclassification of normal samples as attack samples. Similarly, FN refers to the misclassification of attack samples and accepting them as normal samples. The accuracy parameter is defined as the ratio of all correctly classified samples (TP, TN) to all samples (TP, TN, FP, and FN). It is shown in Equation 2. Sensitivity is the ratio of all correctly classified attacks (TP) to the number of correctly classified attacks (TP) and misclassified normal samples (FP). Precision is shown in Equation 4. Recall expresses the ratio of the number of correctly classified positive samples to the number of all correctly classified samples and is shown in Equation 5. The harmonic mean of the Sensitivity and Recall parameters is known as the F-score and is shown in Equation 3.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$F - Score = \frac{2TP}{2TP+FP+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3292209

$$Recall = \frac{TP}{FN+TP} \qquad (5)$$

Figure 9 presents the comparison of the classification results of various machine learning and deep learning algorithms, which are frequently used in the literature, on the WSN-BFSF dataset, according to the accuracy parameter. As can be seen in Figure 9, the GRU deep learning algorithm gives the best accuracy compared to other methods with an accuracy value of 99.02% in the classification of attacks. In this study, the dataset is split into two, 70% for training and 30% for testing. For the hyper parameters in the deep learning algorithms used, firstly, the generally accepted values are given and tuning is done for the best results. It is important for the validity of the data set to divide the data set into training and test sets and to use separate sets for training and testing. In the study, 70% of the data set was used for training the models. The remaining 30% of the dataset was never used until the testing phase. In other words, the model succeeded in classifying data (30% of the data set) that it did not know at all during the test phase with a high percentage of accuracy. The results in Figure 9 and Figure 10 show the validity of the WSN-BFSF dataset.

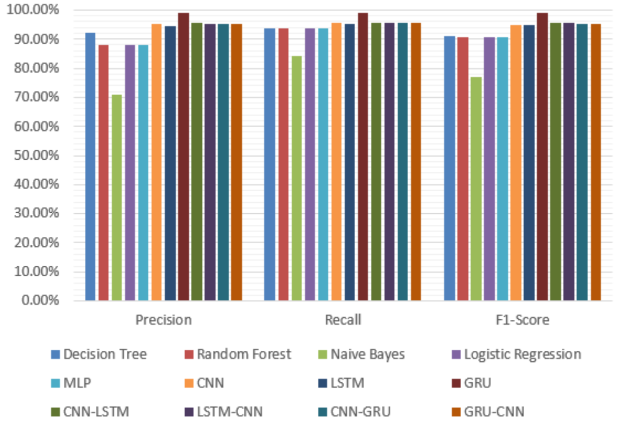


**Fig. 9.** Comparison results according to Precision, Recall and F1-Score parameters.

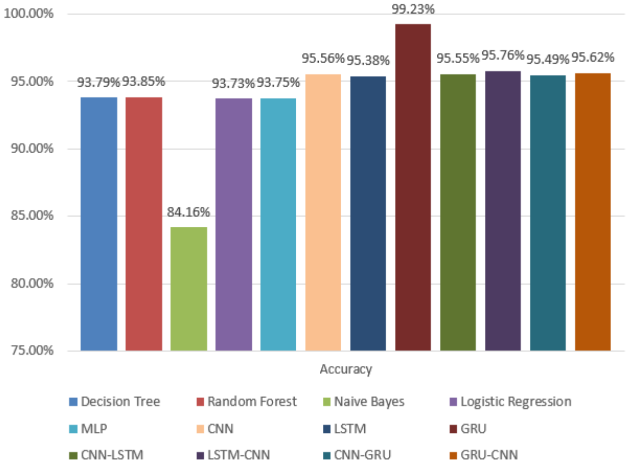


**Fig. 10.** Accuracy results of the models.

In the model established with the Random Forest algorithm, maxDept= 20 and maxBins=50. Maximum depth expresses the frequency of the tree, and as the depth increases, the chance of capturing more information about the data increases. Similarly, in the model established with the Decision Tree algorithm, it is specified as maxDept= 20 and maxBins=50. For the Naive Bayes algorithm, the smoothing value is 1.0 and the model type is Multinomial Naive Bayes parameters. All parameters used in the Logistic Regression model are used by default. The model established with MLP has four layers. Several neurons equal to the number of features used in the data sets were used in the input layer. There are 5 and 4 nerve cells in the hidden layers. In the study, the output layer took the value of 4. In addition, the maximum number of iterations for the MLP model is 100 and the block size is 128.

Models installed with Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have three layers. For both algorithms, 128 neurons are used in the input and hidden layers and 150 neurons are used in the output layer. In the first and second convolution layers of the model established with the Convolutional Neural Network (CNN), filter 32 and kernel size 3 are taken. In the output layer, the filter is configured as 100. The CNN-LSTM model has a convolution layer and an LSTM layer. In the first convolution layer, filter 32 and kernel size 3 are taken. In pooling layers, the stride parameter is 2. There are 128 neurons in the LSTM layer. A Dense layer of 100 neurons was then used. The LSTM-CNN model, on the other hand, has an LSTM and a convolution layer, similar to the CNN-LSTM model. The same parameters used in the CNN-LSTM model were used in the LSTM-CNN model. The CNN-GRU model has a convolution layer and a GRU layer. In the first convolution layer, filter 32 and kernel size 3 are taken. In pooling layers, the stride parameter is 2. There are 128 neurons in the GRU layer. A Dense layer of 100 neurons was then used. The GRU-CNN model, on the other hand, has a GRU and a convolution layer, similar to the CNN-GRU model. The same parameters used in the CNN-GRU model were used in the GRU-CNN model.

The ReLu activation function is used in the input and hidden layers in LSTM, GRU, CNN-1D and CNN-LSTM models built using the Keras library. The output layer has taken the values of 4. In the output layer, the activation function is determined as Softmax. The loss function used in the model is used as categorical cross-entropy. The optimization algorithm is Adam. Models were run for 30 epochs.

When the literature is examined, [8] and [17] studies where data sets were produced also showed the validity of the data sets with the same techniques. MLP, CNN, LSTM and GRU, which is widely preferred in many classification problems, have been used as deep learning algorithms. In addition, CNN-LSTM, LSTM-CNN, CNN-GRU and GRU-CNN hybrid deep learning techniques were also used in this study. As machine learning algorithms, basic machine learning algorithms, which are widely used in the literature, have been used. These algorithms are Random Forest, Decision Tree, Naive Bayes and Logistic Regression. The result obtained for the GRU algorithm was obtained by using the hyper parameters shown in Table 7 as a result of the tuning process.

TABLE VII
HYPERPARAMETERS USED WITH THE GRU ALGORITHM

| **Hyper parameters** | **Values** |
|---|---|
| Activation function | Relu, Softmax |
| Number of Epoch | 30 |
| Learning rate | 0.01 |
| Output size | 150 |
| Drop rate | 0.01 |
| Units | 128 |
| Optimizer | Adam |
| Loss | Categorical entropy |
| Hidden layer | 1 |

When the results of these parameters are examined, the proposed method shows the best results for each parameter. A comparison of the performance of classification algorithms according to accuracy, precision, recall and F1-Score parameters is presented in Figure 9 and Table 8.

TABLE VIII
ACCURACY, PRECISION, RECALL VE F-SCORE RESULTS

| Algorithms | Precision | Recall | F-Score | Accuracy |
|---|---|---|---|---|
| Decision Tree | 0.920382 | 0.937973 | 0.910834 | 0.937973 |
| Random Forest | 0.878815 | 0.938451 | 0.907187 | 0.938451 |
| Naive Bayes | 0.708250 | 0.841576 | 0.769178 | 0.841576 |
| Logistic Regression | 0.878815 | 0.937292 | 0.907112 | 0.937292 |
| MLP | 0.878815 | 0.937451 | 0.907187 | 0.937451 |
| CNN | 0.951428 | 0.955641 | 0.948956 | 0.955641 |
| LSTM | 0.946327 | 0.953759 | 0.946692 | 0.953759 |
| CNN-LSTM | 0.954104 | 0.955503 | 0.954695 | 0.955503 |
| LSTM-CNN | 0.953982 | 0.957598 | 0.955238 | 0.957598 |
| CNN-GRU | 0.951045 | 0.954940 | 0.952519 | 0.954940 |
| GRU-CNN | 0.951019 | 0.956195 | 0.951624 | 0.956195 |
| GRU | 0.990003 | 0.990182 | 0.989900 | 0.990182 |

In addition, detailed classification results according to the evaluation parameters of the GRU deep learning algorithm are presented in Table 9.

TABLE IX
CLASSIFICATION PERFORMANCE OF THE GRU ALGORITHM

| **Parameters** | **Blackhole** | **Flooding** | **Selective Forwarding** | **Normal** |
|---|---|---|---|---|
| Accuracy | 0.82158150 | 0.99800288 | 0.97728229 | 0.99714336 |
| Precision | 0.95392774 | 0.99085702 | 0.95636064 | 0.99251412 |
| Recall | 0.82158150 | 0.99800288 | 0.97728229 | 0.99714336 |
| F1-Score | 0.88282209 | 0.99441711 | 0.96670828 | 0.99482336 |

From the results obtained, the GRU deep learning algorithm achieved 82.16%, 99.80%, 97.73%, and 99.71% classification accuracy according to Blackhole, Flooding, Selective Forwarding attack traffics and Normal traffic classes, respectively. The classification results of the proposed algorithm for each class are shown in Figure 11 on the Confusion Matrix.

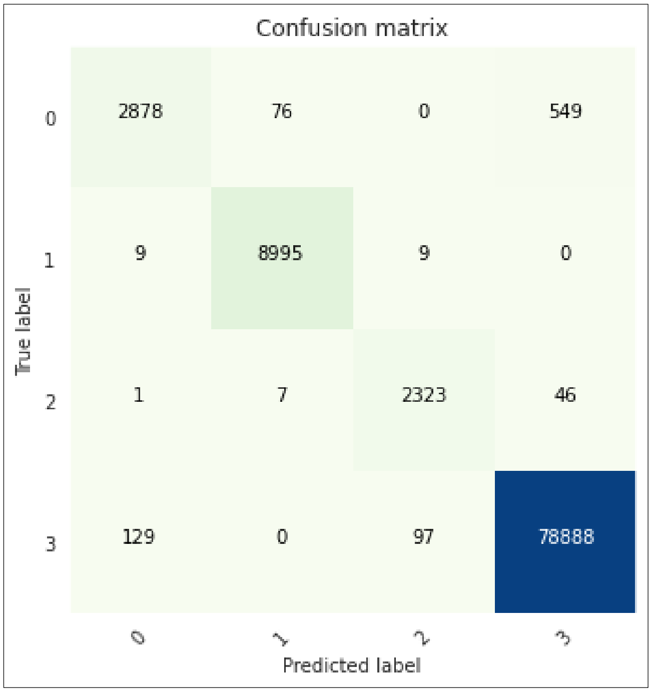


**Fig. 11.** Confusion Matrix.

It is seen from Confusion Matrix that the proposed algorithm is successful for all classes. Conversion and class numbers shown numerically in the figures are shown in Table 6. Figure 11 shows the actual distribution of the data set according to traffic classes and the distribution classified by the models. The vertical axis shows how many samples the data set contains from which traffic. On the horizontal axis, there exists the predictions of the models. Besides, ROC and Precision-Recall curves used for accurate comparison of classification performances in imbalanced datasets are shown in Figure 12 and Figure 13, respectively.

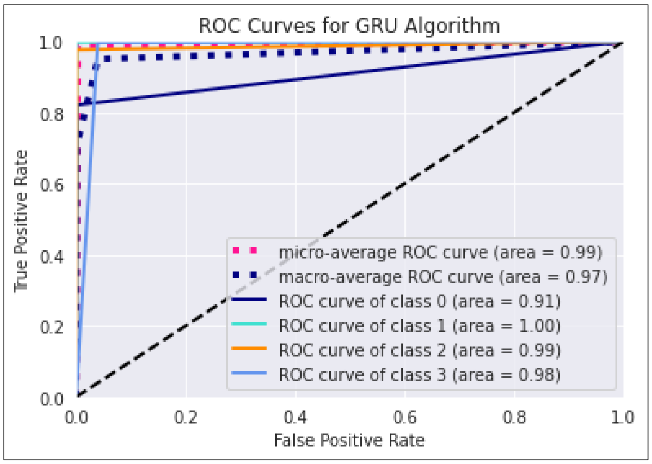


**Fig. 12.** ROC curve.

Another parameter useful in measuring the quality of classification success in the study is the ROC curves obtained in Figure 12. In ROC curves, the X axis shows the False

Positive ratio, while the Y axis shows the True Positive ratio. Therefore, results close to the upper left corner are considered ideal results. When the results obtained in the study are evaluated, it is seen that the curves for the GRU algorithm are close to the ideal point, and quite successful results are obtained for all classes.

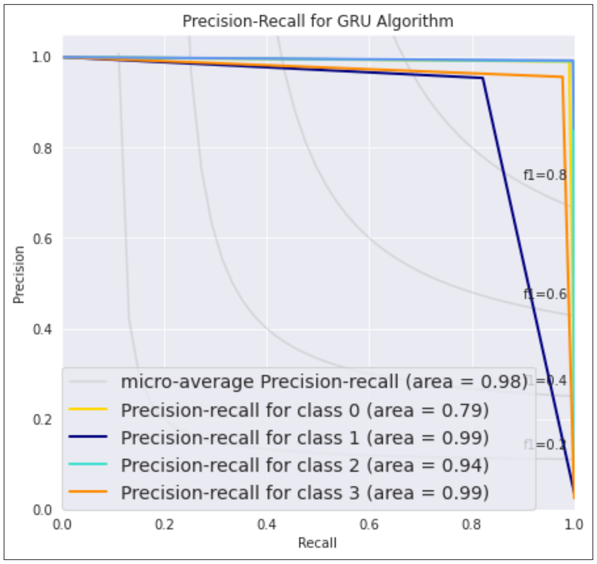


**Fig. 13.** Precision-Recall curve.

Precision-Recall curves are a very useful parameter for measuring prediction success when classes are very unstable. This curve shows the balance between precision and recall. A high area under the curve represents both high recall and high precision. When the Precision-Recall curves obtained in Figure 13 are examined, it shows that the area under the curves for the proposed GRU algorithm is close to 1, which is the maximum value, and very successful results are obtained for all classes. In Table 10, a comparison of classification algorithms according to accuracy parameters for each class is presented.

TABLE X
ACCURACY VALUE PERFORMANCE OF ALGORITHMS ACCORDING TO TRAFFIC CLASSES

| **Algorithms** | **Blackhole** | **Flooding** | **Selective Forwarding** | **Normal** |
|---|---|---|---|---|
| Decision Tree | 0.01 | 0.99 | 0.01 | 1.00 |
| Random Forest | 0.04 | 1.00 | 0.01 | 1.00 |
| Naive Bayes | 0.01 | 0.01 | 0.01 | 1.00 |
| Logistic Regression | 0.01 | 1.00 | 0.01 | 1.00 |
| MLP | 0.01 | 1.00 | 0.01 | 1.00 |
| CNN | 0.42 | 0.98 | 0.41 | 0.99 |
| LSTM | 0.35 | 0.98 | 0.65 | 0.99 |
| CNN-LSTM | 0.59 | 0.99 | 0.61 | 0.98 |
| LSTM-CNN | 0.61 | 0.98 | 0.62 | 0.99 |
| CNN-GRU | 0.59 | 0.99 | 0.61 | 0.98 |
| GRU-CNN | 0.62 | 0.99 | 0.62 | 0.99 |
| GRU | 0.82 | 0.99 | 0.97 | 0.99 |

From the results obtained, it was observed that deep learning algorithms achieved better results than traditional machine learning algorithms. The naive Bayes algorithm showed the lowest performance. The Multinomial Naive Bayes algorithm was used in this study. The naive Bayes algorithm showed the lowest performance because it could not detect the data of attack classes at a high rate. When Table 10 is examined, it is seen that machine learning algorithms cannot correctly classify the attack traffic classes that contain a small number of data. MLP, CNN and LSTM deep learning algorithms cannot successfully classify data belonging to Blackhole and Selective Forwarding classes. This is because the dataset is created in an imbalanced manner in accordance with reality. In real WSN environments, the amount of normal traffic passing through the network is considerably higher than the amount of attack traffic.

In this study, basic Random Forest, Decision Tree, Naïve Bayes and Logistic Regression machine learning algorithms, whose performance has been evaluated in many intrusion detection system studies, and CNN, LSTM and GRU deep learning algorithms and their hybrid structures are used. CNN algorithm is mostly used in image processing and classification of two-dimensional data. However, as in this study, it can also be used for one-dimensional data and performs quite well. Convolutional neural networks are the regulated version of MLP algorithms. In MLP networks, each neuron in one layer is connected to all neurons in the next layer. This increases the possibility of data overfitting in MLP networks. To overcome this situation, CNN carries out various regulatory processes. The LSTM algorithm is frequently used in natural language processing and video processing applications, thanks to its ability to store sequential data in memory. However, as in this study, the LSTM algorithm offers very successful results in cases where the historical data of WSN-specific features such as Time, Rest Energy and Src_Seq_Num are also important in DoS attack detection. Similarly, the GRU algorithm plays an important role in detecting DoS attacks by keeping historical data. Although deep learning algorithms are more complex than machine learning algorithms, they produce better results in terms of performance. As a result of this situation, it has become increasingly popular in many classification studies and has achieved significant success. In this study, it has been confirmed by the obtained results that deep learning algorithms exhibit successful results in detecting DoS attacks in wireless sensor networks.

In our study [43], we also carried out studies on the WSN-DS dataset for intrusion detection in wireless sensor networks. While detecting Grayhole, Blackhole, TDMA and Flooding attacks in the WSN-DS dataset, a detection study was carried out against Blackhole, Flooding and Selective Forwarding attacks in the proposed WSN-BFSF dataset. Machine learning and deep learning algorithms similar to those in this study were used in the WSN-DS dataset. When the results in the WSN-BFSF dataset produced in this study and the WSN-DS dataset are compared, deep learning algorithms have achieved successful results in both datasets. From these results, it is seen that the proposed WSN-BFSF dataset can be used by researchers to detect DoS attacks in wireless sensor networks. Unlike the WSN-DS dataset, it is considered that it will contribute to the studies of researchers especially for Selective forwarding attacks.

## VII. CONCLUSION

In this study, the most commonly used network layer attacks in WSN namely Blackhole, Flooding, and Selective Forwarding were carried out in the NS-2 simulation environment. In the same network, 4 different traffic types as being three different attack traffic and normal traffic were created. Since there will be a need for the normal traffic while using the WSN-BFSF data set, during the attacks normal traffic continued to work. While running different traffics on the same network in the NS-2 simulation environment, error messages are frequently generated due to factors such as traffic conflict, excessive traffic density, and unbalanced packet size, and as a result the platform stops working. Paying attention to these details is crucial to be able to get the intended attack traffic. After the network traffic was completed, the raw form of the WSN-BFSF dataset was taken from the simulation platform and preprocessed incrementally. The data set which is ready for use after the preprocessing consist of 16 features and 312 106 rows. WSN-BFSF data set consists of 84% normal traffic, 10% Flooding attacks traffic, 4% Blackhole attack traffic and 2% Selective Forwarding attacks. WSN-BFSF data set was examined with 4 different machine learning models as being Random Forest, Decision Tree, Naive Bayes and Logistic Regression and 8 different deep learning models as being MLP, CNN, LSTM, GRU, CNN-LSTM, LSTM-CNN, CNN-GRU and GRU-CNN with a total of 12 different models in total. In addition to the classification of attack and normal traffics, attack traffic is also classified by attack type. While classifying the examined data set, models are combined with combining hyper parameters such as learning rate, epoch, drop rate optimizer and loss function to get the highest accuracy percentage. The highest accuracy rate is obtained with GRU deep learning model with 99.02% percent rate. While reaching this rate, the GRU model, ReLu and Softmax activation functions, 30 epoch values of 0.01 learning rate, categorical entropy loss function and Adam optimizer algorithm were used.

## REFERENCES

[1] M. Abazeed, et al., “A review of secure routing approaches for current and next- generation wireless multimedia sensor networks, *International Journal of Distributed Sensor Networks*,11(10),1-22, 2015.

[2] V. Ekong, and U. Ekong, “A survey of security vulnerabilities in wireless sensor networks,” *Nigerian Journal of Technology*, 35(2), 392, 2016.

[3] C. Karlof, and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” *Ad Hoc Networks*, 1 (2-3), 293-315, 2003.

[4] D. Deif, and Y. Gadallah, “An ant colony optimization approach for the deployment of reliable wireless sensor networks,” *IEEE Access*, 5 (1), 10744-10756, 2017.

[5] Z. Sheng, C. Mahapatra, C. Zhu, and V. Leung, “Recent advances in ındustrial wireless sensor networks toward efficient management in IoT,” *IEEE Access*, 3 (1), 622-637, 2015.

[6] M. Selvi, K. Thangaramya, S. Ganapathy, K. Kulothungan, H. Khannah Nehemiah and A. Kannan, “An energy aware trust based secure routing algorithm for effective communication in wireless sensor networks,” *Wireless Personal Communications*, 105 (4), 1475-1490, 2019.

[7] M. Dener, “Security analysis in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, 10 (10),303-501, 2014.

[8] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, “WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks,” *Journal of Sensors*, 1–16, 2016.

[9] S.Ifzarne, H. Tabbaa, I. Hafidi, and N. Lamghari, “Anomaly detection using machine learning techniques in wireless sensor networks,” *Journal of Physics: Conference Series*, 1743 (1), 012-021, 2021

[10] V. Garcia-Font, C. Garrigues and H. Rifà-Pous, “A comparative study of anomaly detection techniques for smart city wireless sensor networks,” *Sensors*, 16(6), 868, 2016.

[11] N. Alrajeh, S. Khan, J. Lloret, and Jo. Loo, “Artificial neural network based detection of energy exhaustion attacks in wireless sensor networks capable of energy harvesting,” *Ad-Hoc and Sensor Wireless Networks*, 22, 109-133, 2014.

[12] S. Otoum, B. Kantarci and H. Mouftah, “On the feasibility of deep learning in sensor network intrusion detection,” *IEEE Networking Letters*, 1(2), 68-71, 2019.

[13] R. Alshinina, and K. Elleithy, “A highly accurate deep learning based approach for developing wireless sensor network middleware,” *IEEE Access*, 6, 29885-29898, 2018.

[14] M. Pawar, and J. Anuradha, “Detection and prevention of black-hole and wormhole attacks in wireless sensor network using optimized LSTM,” *International Journal of Pervasive Computing and Communications*, 88-95, 2021.

[15] H. Bahsi, S. Nomm, and F. B La Torre, “Dimensionality reduction for machine learning based ıot botnet detection,” 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 1857-1862, 2018.

[16] S. A. Sokolov, T. B. Iliev, and I. S. Stoyanov, “Analysis of cybersecurity threats in cloud applications using deep learning techniques,” 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Astana, 441-446, 2019.

[17] S. Nomm, H. Bahsi, “Unsupervised anomaly based botnet detection in ıot networks,” 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, 1048-1053, 2018.

[18] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and Burnap, P, “A supervised ıntrusion detection system for smart home IoT devices,” *IEEE Internet of Things Journal*, 6(5), 9042-9053, 2019.

[19] A. Kumar, T. J. Lim, “EDIMA: Early detection of IoT malware network activity using machine learning techniques,” IEEE 5th World Forum on Internet of Things (WF-IoT), Singapur, 289-294, 2019.

[20] I. P. Possebon, A. S. Silva, L. Z., Granville, A. Schaeffer-Filho and A. Marnerides, “Improved Network Traffic Classification Using Ensemble,” *IEEE Symposium on Computers and Communications* (ISCC),1-6, 2019.

[21] Y. Meidan, M. Bohadana, et. al, “Profiliot: a machine learning approach for IoT device identification based on network traffic analysis,” Symposium on Applied Computing (SAC), 506–509, 2017.

[22] H. Bai, X. Zhang, and F. Liu, “Intrusion detection algorithm based on change rates of multiple attributes for WSN,” *Wireless Communications and Mobile Computing*, 1-16, 2020.

[23] E. S. Karanja, S. Masupe, M. Jeffrey, “Analysis of Internet of things malware using ımage texture features and machine learning techniques,” *Internet of Things*, 9,100-153, 2020.

[24] I. Cvitić, D. Perakovic, M. Periša, and M. Botica, “Smart home IoT traffic characteristics as a basis for DDOS traffic detection,” 3rd EAI International Conference on Management of Manufacturing Systems, Dubrovnik, 112-121, 2018.

[25] G. Liu, X. Wang, X. Li, J. Hao, and Z. Feng, “ESRQ: an efficient secure routing method in wireless sensor networks based on Q-Learning,”17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). 149–155, 2018.

[26] M. Anwer, U. Muhammad K Farooq, F. Waseemullah, “Attack detection in IoT using machine learning,” *Engineering, Technology & Applied Science Research*,11 (3), 7273-727, 2021.

[27] C. Okur, M. Dener, “Detection of dos attacks using machine learning methods in wireless sensor networks,” *El-Cezerî Journal of Science and Engineering*, 8(3), 1550-1564, 2021.

[28] C. Okur, M. Dener, “M. Detecting IoT botnet attacks using machine learning methods,” 13. International Conference on Information Security and Cryptology, Ankara, 21-25, 2020.

[29] C. Okur, A. Orman, M. Dener, (2022). DDOS intrusion detection with machine learning models: N-BaIoT data set. 4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2022), İstanbul, 772-776, 2022.

[30] R. Wazirali, R. Ahmad, “Machine Learning Approaches to Detect DoS and Their Effect on WSNs Lifetime,” *Computers, Materials & Continua*, 70(3), 2021.

[31] A. Branitskiy, and I. Kotenko. "Hybridization of computational intelligence methods for attack detection in computer networks," *Journal of Computational Science*, 23, 145-156, 2017.

[32] A. Fatani, A. Dahou, M. A. Al-qaness, S. Lu, and M. A. Abd Elaziz, "Advanced feature extraction and selection approach using Deep Learning and Aquila Optimizer for IOT Intrusion Detection System," Sensors, vol. 22, no. 1, p. 140, 2021

[33] M. Otair, O. T. Ibrahim, L. Abualigah, M. Altalhi, and P. Sumari, "An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks," Wireless Networks, vol. 28, no. 2, pp. 721–744, 2022

[34] J. Ahmad, S. A. Shah, S. Latif, F. Ahmed, Z. Zou, and N. Pitropakis, "DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 8112–8121, 2022

[35] Ahmed, A. Dahou, S. A. Chelloug, M. A. Al-qaness, and M. A. Elaziz, "Feature selection model based on Gorilla Troops Optimizer for intrusion detection systems," Journal of Sensors, vol. 2022, pp. 1–12, 2022.

[36] Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization," Expert Systems with Applications, vol. 204, p. 117597, 2022.

[37] A. Dahou, M. Abd Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. Al-Betar, M. A. Al-qaness, and A. Forestiero, "Intrusion detection system for IOT based on Deep Learning and modified reptile search algorithm," Computational Intelligence and Neuroscience, vol. 2022, pp. 1–15, 2022.

[38] C. Okur, M. Dener, "Performance comparison of AODV and DSR routing protocols," V. International Scientific and Vocational Studies Congress Engineering, Ankara, 201-206, 2020.

[39] M.Islam A. Fahmin, M. Hossain and M. Atiquzzaman, "Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques," *Wireless Personal Communications*, 116(3), 1993-2021, 2020.

[40] H. C. Chaudhari, and L. U. Kadam, "Wireless sensor networks: Security, attacks and challenges,"*International Journal of Networks*, 1(1), 4-16, 2011.

[41] V. P., Singh, S., Jain, J. Singhai, "Hello flood attack and its countermeasures in wireless sensor networks," *International Journal of Computer Sciences*,7(3), 23, 2010.

[42] S.Al, M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment", Computers & Security, 110, 1-21, 2021.

[43] M.Dener, S.Al, A.Orman, ""STLGBM-DDS: An Efficient Data Balanced DoS Detecion System for Wireless Sensor Networks on Big Data Environment", IEEE Access, Vol 10, 92931-92945, 2022.

**Murat Dener** works as a faculty member at Gazi University. Dener, who received the title of associate professor from Computer Science and Engineering, is also the head of the Information Security Engineering Department. He has been working in the field of Internet of Things, Information Security and Smart Cities for nearly 15 years. Dener has more than 120 published international and national studies.

**Celil Okur** is an MSc student in Information Security Engineering Department. He has been working in the field of Big Data Analytics, Wireless Sensor Networks and Information Security.

**Samed Al** is a PhD student in Information Security Engineering Department. He has been working in the field of Big Data Analytics, Explainable Artificial Intelligence and Information Security.

**Abdullah Orman** works as a faculty member at Ankara Yıldırım Beyazıt University. He has been working in the field of Computer Networks, Database and Data Structures, Artificial Intelligence and Information Security.