Enhancing Security in Wireless Sensor Networks: A Machine Learning-based DoS Attack Detection

Ghadeer Al Sukkar
Department of Computer Science, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan | Al-Hussein Technical University, Amman, Jordan
gdy9224472@ju.edu.jo

Saleh Al-Sharaeh
Department of Computer Science, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan
ssharaeh@ju.edu.jo (corresponding author)

Received: 3 March 2024 | Revised: 11 October 2024 and 21 October 2024 | Accepted: 4 December 2024

 DOI: https://doi.org/10.48084/etasr.7191

ABSTRACT

The Internet of Things (IoT) is based on Wireless Sensor Networks (WSNs), which are essential for many applications. Denial of Service (DoS) attacks are a major risk for WSNs due to their open architecture and limited resources. This paper investigates how different Machine Learning (ML) methods can be used to identify DoS attacks and mitigate their effects. The predictions from several models were combined using the ensemble method to increase overall accuracy, while explainable Artificial Intelligence (AI) techniques were also deployed to enhance transparency and understanding. To compare the performance of both hard and soft ensemble methods, the WSN Dataset (WSN-DS) and the WSN Blackhole, Flooding, and Selective Forwarding (WSN-BFSF) dataset were utilized. The ensemble techniques aggregated predictions from multiple models to improve overall accuracy, while both showed high accuracy for both datasets. With an accuracy of 98.12%, the soft ensemble technique slightly outperformed the hard ensemble technique for the WSN-DS dataset, which had an accuracy of 97.97%. For the WSN-BFSF dataset, the hard ensemble technique achieved an accuracy of 99.967%, while the soft ensemble technique achieved an excellent accuracy of 100%.

Keywords-security; wireless sensor networks; machine learning; DoS attack detection

I. INTRODUCTION

WSNs are geographically distributed networks of sensors that communicate wirelessly to track various environmental or physical parameters. Small, inexpensive sensor nodes with sensing, processing, and communication capabilities make up most of these networks. With their ability to collect and transmit data in real time without the need for a pre-existing communication infrastructure, WSNs have found applications across a wide range of fields. They can be detected in a variety of industries, supporting increased productivity, improved monitoring, and informed decision-making [1]. As a key component of contemporary technology, WSNs enable the real-time collection and transmission of data for a wide range of applications including healthcare, industrial automation, agriculture, and environmental monitoring [2]. In the healthcare industry, they are used to manage medical equipment, track vital signs, and monitor patients, enabling remote and customized healthcare solutions. They support industrial automation by facilitating machinery monitoring, process optimization, and predictive maintenance, thereby increasing overall productivity [3]. WSNs are essential for environmental monitoring, providing real-time data on climate parameters, water quality, and air quality, which helps in disaster management and ecological preservation [4]. However, due to their widespread use and crucial nature, WSNs are vulnerable to constantly changing security threats, with one of the most dangerous being the DoS attacks [5]. In WSNs, a DoS attack is an attempt by malicious actors to disrupt normal network operations, compromise the integrity of data transmission, or limit the resources of certain sensor nodes [6]. DoS attacks on WSNs can have a serious impact on the reliability and efficiency of critical applications of these networks. DoS attacks are a type of cyber-attack that disrupts computer systems, networks, or the normal operation of online services by rendering them inaccessible to users for either a period of time or permanently [7]. They exploit security vulnerabilities and cause DoS to authorized users by flooding the target with traffic or depleting system resources. Flooding attacks flood the network with excessive traffic, causing

congestion and even resource exhaustion. Jamming attacks cause disruption and increased energy consumption by interfering with wireless transmissions [8]. DoS attacks pose a significant risk to WSNs, affecting their reliability and seamless operation. Attackers exploit the unique vulnerabilities of these dynamic networks to disrupt communications, compromise data integrity, and consume vital resources [9]. Sinkhole attacks, Sybil attacks, and selective forwarding manipulate the data flow, compromising the reliability and accuracy of the information collected by the network. Physical layer attacks pose another level of vulnerability by destroying or manipulating the sensor nodes [10].

This study aims to address this security challenge by investigating how ML techniques can be used in WSNs to identify and minimize DoS attacks. The security of WSNs is challenging due to their inherent characteristics, which include many networked sensor nodes operating with limited resources [11]. Traditional security protocols designed for traditional networks are inadequate to address the unique problems of WSNs [12, 13]. Although several research techniques have been developed to address the problems of detecting DoS attacks in WSNs, it is still difficult to identify an effective detection solution [14]. This study presents ML as a dynamic and adaptable method to strengthen the security measures of WSNs in response to these problems. ML methods, which are known for their ability to detect patterns and anomalies across huge datasets, provide the ability to detect indicators of DoS attacks. This study aims to develop an effective DoS detection system specific to WSNs by utilizing supervised learning methods, such as the Logistic Regression (LR), Decision Tree (DT) classifier, K-Nearest Neighbors (KNN) algorithm, Gradient Boosting (GB) classifier, Support Vector Machines (SVM), and the Voting classifier. The investigation of ML based DoS detection takes place in a structured manner and includes key steps, such as data preprocessing, splitting the data into training and test data, and applying multiple ML algorithms. The present study focuses on addressing the critical security issues that WSNs are facing, by reducing the risk of DoS attacks.

II. LITERATURE SURVEY

Authors in [15] proposed the Weighted Score Selector (WSS), a novel ensemble-based method, to detect attacks in WSNs. Several supervised ML classifiers, including Naive Bayes (NB), SVM, LightGBM, DT, Random Forest (RF), and KNN were used to implement the proposed method and increase the detection accuracy. The proposed method was applied on the WSN-DS with promising results. Authors in [5] presented a simple ML detection technique for DoS attacks in WSNs, which combines the Gini feature selection method with a DT algorithm and was applied to a modified WSN-DS dataset. In comparison to the RF, the proposed method demonstrated good performance by achieving a high accuracy rate with low overhead. To improve the detection of DoS attacks, authors in [16] presented a WSN intrusion detection model that combines the KNN with the arithmetic optimization algorithm. The experimental results demonstrated the usefulness of the proposed detection model on the WSN-DS dataset. Authors in [14] introduced the CH_Rotations algorithm, a revolutionary clustering technique to improve the detection accuracy of the DoS attacks. Furthermore, the use of feature selection techniques along with ML algorithms in examining WSN node traffic and the effect of these techniques on the lifetime of WSNs was evaluated. The evaluation results using the WSN-DS dataset showed that the Water Cycle (WC) feature selection performed better on the WSN-DS dataset than other optimization methods. In addition, the proposed approach showed good accuracy with only one feature. Authors in [9] proposed the Deep Learning-based Defense Mechanism (DLDM), a novel lightweight DoS detection technique to detect and isolate the attacks in the Data Forwarding Phase (DFP), including flooding, homing, jamming, and exhaustion. Deep Learning (DL) was combined with the proposed detection approach to provide flexible and cost-effective cluster-based measures for effective detection and isolation. The outcome of the proposed method demonstrated that it can achieve high accuracy and low energy consumption. Authors in [17] proposed an NB classifier and a round-trip time-based method to detect black hole and wormhole attacks. The main advantage of the proposed method was the reduction of communication overhead. The current study aims to address the emerging security threats that WSNs are facing, with a particular emphasis on reducing the risk of DoS attacks.

III. DATASETS AND METHODOLOGY

A. Datasets

In this study, two datasets are used to train and test the ML methods and techniques:

The WSN-DS dataset by employing the LEACH approach [18]. The dataset contains numerous attack scenarios along with WSN features for more precise detection and categorization of four different types of DoS attacks: Flooding, Scheduling (TDMA), Blackhole, and Grayhole. It has a total of 374661 records and 19 attributes, as evidenced inTable I.

The WSN-BFSF dataset was introduced in [19]. Three categories of network layer attacks for WSNs were developed by the WSN-BFSF dataset consisting of 312106 rows. It additionally contains information from the selective forwarding attack, in contrast to the WSN-DS dataset. Table II shows the attributes of the WSN-BFSF dataset.

B. Preprocessing of Datasets

The data are considered imbalanced when there is a non-uniform or unequal distribution of class labels in a classification task. That is, the minority class has much fewer instances than the majority class. This imbalance in the distribution of classes can pose a challenge to ML models, particularly those that aim to maximize overall accuracy, as it can lead to a bias toward predicting the majority class [20]. In ML, class imbalance is a common problem, especially in classification problems where one class greatly dominates the other. Downsampling is a preprocessing approach used to create a more balanced dataset by randomly deleting samples from the majority class to address this problem [21]. The

process normally starts with understanding the distribution of classes and then dividing the data into training and testing sets while maintaining class proportions. Next, the imbalance ratio is computed [20]. After determining the optimal class balance, downsampling is performed utilizing a randomly selected subset of the majority class samples to match the size of the minority class. To solve the imbalance problem in the two datasets, downsampling is applied to them.

TABLE I. WSN-DS ATTRIBUTES

No	Attribute Name	Attribute Description
1	ID	A unique identification number to identify the sensor node at any time and in any round
2	Time	The node's current simulation time
3	Is CH	A flag to distinguish CH from a normal node
4	Who CH	CH identifier in the current round
5	Dist_To_CH	How far the node is from its CH
6	Energy Consumption	The amount of energy consumed in the last round
7	ADV_CH send	The number of advertized CH's broadcast messages sent to the nodes
8	ADV_CH receives	The number of advertized CH messages received from CHs
9	Join_REQ send	The number of join request messages the nodes send to the CH
10	Join_REQ receives	The number of join request messages received by the CH from the nodes
11	ADV_SCH send	The number of advertized TDMA schedule broadcast messages sent to the nodes.
12	ADV_SCH receives	The number of TDMA schedule messages received from CHs
13	Rank	The order of this node within the TDMA schedule
14	Data S	The number of data packets sent from a sensor to its CH
15	Data R	The number of data packets received from CH
16	Data sent to BS	The number of data packets sent to the BS
17	Dist- CH to BS	The distance between the CH and the BS
18	Send Code	The cluster sending code
19	Attack Type	The type of the node

TABLE II. WSN-BFSF ATTRIBUTES

No	Attribute Name	Attribute Description
1	Event	It contains information about the operation performed on the traffic. It is represented by the value 1 for sending, 2 for receiving, 3 for forwarding, 4 for dropping, and 5 for Energy information
2	Time	The time of the event performed in the row
3	S_Node	The source node's number
4	Node_id	The number of the relevant node
5	Rest_Energy	The remaining energy of the relevant node
6	Mac_Type_Pck	The MAC type of the packet
7	Source_IP_Port	The port number of the source node
8	Packet_Size	The size of the forwarded packet
9	TTL	The lifetime of the forwarded traffic in the network
10	Hop_Count	The number of passed nodes.
11	Broadcast_ID	The ID number of the broadcast packets
12	Dest_Node_Num	The ID of the target node
13	Dest_Seq_Num	The sequence number of the traffic forwarded to the destination
14	Src_Node_ID	The ID number of the source node
15	Src_Seq_Num	The source sequence number of the traffic forwarded to the destination
16	Class	The type of classified network traffic

C. Methodology

First, the dataset undergoes a preprocessing phase, including balancing to ensure that the classes are evenly distributed within the dataset, and standardizing the features to normalize the data range. The data are then divided into subsets for testing and training. Once the data are prepared, they enter the model training phase, where multiple ML algorithms are applied. These include LR, KNN, SVM, DT, and GB. Each algorithm makes predictions denoted as Pred LR, Pred KNN, Pred SVM, Pred DT, and Pred GB, respectively. The predictions from these models are then fed into two types of ensemble techniques, hard and soft voting. Hard voting takes the most frequent prediction from the classifiers as the final prediction, while soft voting considers the confidence level of the predictions from each classifier. Finally, the ensemble predictions are evaluated to assess the performance of the combined model. The outcome of this evaluation is then used in conjunction with explainable AI techniques to provide insights and explanations for the predictions made by the model, thereby enhancing transparency and the understanding of the model's decision process. Techniques like feature importance analysis, partial dependence graphs, and model-agnostic approaches are frequently used in the explainable AI methodology. These techniques aim to preserve interpretability and transparency while clarifying the decision-making process of the AI algorithms. Figure 1 depicts the flowchart of the methodology.

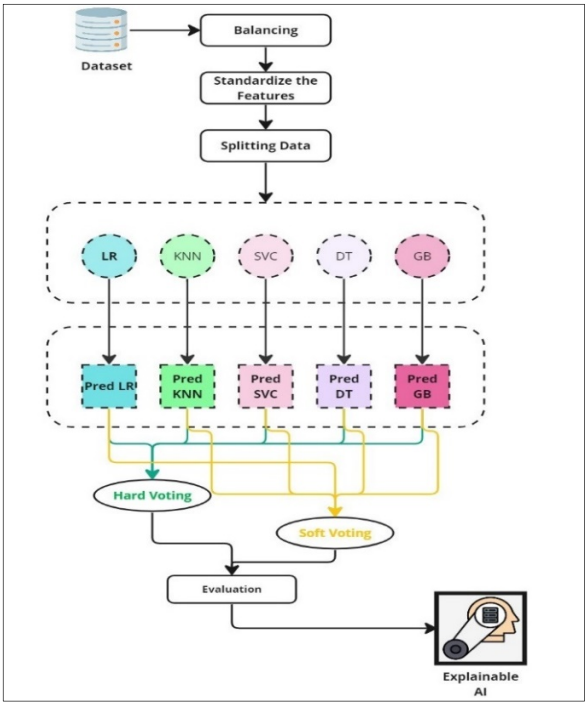

Fig. 1. The flowchart of the methodology.

IV. RESULTS

The results presented here span across five different ML models applied to a dataset for classification tasks, with each model yielding different performance metrics. Starting with the

WSN-DS dataset, the LR achieved an accuracy of approximately 90.1%. The precision and recall metrics varied across different classes. For instance, the LR was highly precise and completely recall-efficient in identifying the 'Flooding' class, while it had lower precision in identifying the 'Blackhole' class, albeit with perfect recall. The F1-score, which balances precision and recall, was high for the 'Normal' and 'TDMA' classes, indicating good overall performance for these categories. The confusion matrix for the LR indicated perfect classification for 'Flooding' but there was some confusion in distinguishing 'Grayhole' from the other classes. The KNN model exhibited an improvement in accuracy of approximately 95.3%. This model had high precision and recall across all classes, particularly excelling at classifying 'Flooding'. The F1-scores were consistently high, demonstrating a balanced performance in both the precision and recall aspects of classification. The confusion matrix for the KNN showed very few misclassifications, with most errors coming from mislabeling within the 'Blackhole' and 'Grayhole' classes. The SVM came in with an accuracy close to that of the LR, around 90.3%. Its precision and recall metrics were similar to those of the LR, doing exceptionally well with the 'Flooding' class and somewhat less so with the 'Grayhole' class. The F1-scores were comparable to those of the LR, indicating a similar balance of performance. The confusion matrix for the SVM displayed some misclassification in the 'Grayhole' and 'TDMA' classes, but excellent performance in 'Flooding'. The DT classifier marked a significant increase in accuracy of about 96.7%, showing high precision and recall across all classes. This model was particularly effective at identifying the 'Blackhole' class, with nearly perfect metrics. However, there were some misclassifications in the 'Normal' and 'TDMA' classes, as reflected in the confusion matrix, where 'Normal' was sometimes mistaken for 'TDMA'. Lastly, GB achieved the highest accuracy among all models at approximately 97.6%. It demonstrated very high precision and recall across all classes, nearly perfect in many cases. The 'Flooding' class was identified with 100% accuracy. The F1-scores for GB were the highest, indicating robust performance. The confusion matrix revealed very few errors, with 'Grayhole' being the most challenging class but still with very high accuracy. In summary, while all models performed well, the GB and DT stood out in terms of accuracy and other metrics. The models varied in their ability to accurately predict different classes, with 'Flooding' being the easiest to predict across all models, and 'Grayhole' being the most challenging. The high F1-scores for GB and DT suggest that these models were able to effectively balance precision and recall, making them potentially more reliable for this classification task. Each model had its strengths and weaknesses, but the ensemble techniques, when used, were able to potentially leverage them to further improve classification performance. Both ensemble methods, soft and hard voting, were deployed to integrate the predictions from multiple models to improve the overall classification accuracy. Soft voting considers the classifiers' predicted probabilities, giving more weight to votes with higher confidence. In hard voting, only the predicted class labels are considered, and the most frequent label is chosen as the final prediction. For the soft voting ensemble, the achieved accuracy is approximately 98%, reflecting a sophisticated blend of the individual classifiers' strengths. Precision, recall, and F1-scores are exceptionally high across all classes, nearly perfect for the 'Blackhole' and 'Flooding' classes. The 'Grayhole' class also displays excellent recall, indicating that the ensemble is very effective at detecting this category. The 'Normal' and 'TDMA' classes exhibit slightly lower precision, which could indicate occasional misclassifications. However, the F1-scores are still quite high, suggesting a balance between precision and recall. The confusion matrix for soft voting shows very few misclassifications across the board, confirming the robustness of this approach. The hard voting ensemble performs slightly better than the soft voting ensemble, with an accuracy of just over 98.1%. However, it matches the soft voting ensemble's high precision and recall, with near-perfect scores for the 'Flooding' and 'Blackhole' classes. The F1-scores for the 'Grayhole', 'Normal', and 'TDMA' classes are marginally higher than those of the soft voting ensemble, indicating an even better balance between precision and recall. This suggests that the hard voting ensemble is slightly more effective at class consensus when it comes to a majority rule approach. The confusion matrix for hard voting confirms this, showing minimal misclassifications and a high level of accuracy in class predictions. When comparing the two ensemble techniques, it is evident that they perform exceptionally well, with the hard voting ensemble having a slight edge. The increased precision and recall rates across the classes with both ensemble methods demonstrate the power of combining multiple ML models. By leveraging the diverse strengths of individual classifiers, ensemble methods can provide a more accurate and reliable classification system. This is particularly beneficial for complex or imbalanced datasets where single-model predictions might falter. The consistency of high F1-scores across the classes also indicates that the ensemble is stable and robust, which is critical for practical applications where high reliability is essential. Figure 2 illustrates the accuracy of the applied ML models on the WSN-DS dataset.

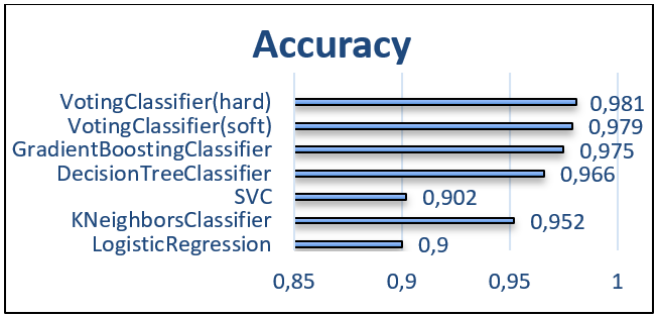

Fig. 2. Accuracy of the applied ML models on the WSN-DS dataset.

Figure 3 portrays an output of the model's prediction probabilities for the multi-class classification task, along with the feature importance ranking for the class labeled 'Normal'. On the left side, the prediction probabilities are displayed, indicating the model's confidence in assigning the data point to each of the potential classes: 'Normal', 'Grayhole', 'Blackhole', 'TDMA', and 'Flooding'. In this instance, the model predicts with high confidence, 0.99 probability, that the data point belongs to the 'Normal' class, while the probabilities for the

other classes are 0.00, suggesting that the model is quite certain of this classification. The feature importance values for predicting the 'Normal' class are listed on the right side. The model indicates that features like 'Data_R' with a high positive value are more indicative of the 'Normal' class, whereas features with negative values, such as 'Expanded Energy', 'Time', and ADV_S,' are less indicative of the 'Normal' class. Each feature has a value that indicates how important it is to the model's decision-making process; positive values help the model forecast something to be 'Normal,' while negative values make it less so. This type of analysis is crucial for understanding what drives the decisions of the ML models, particularly in the field of explainable AI, where it is important to make the model's decision-making process as transparent and understandable as possible. It also helps validate the model by ensuring that the predictions are based on sensible data-driven insights.

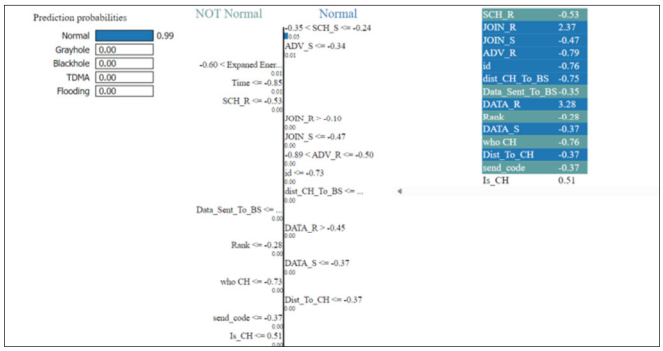

Fig. 3. Feature importance ranking for the 'Normal' class.

Figure 4 illustrates the model's prediction output for a single instance, focusing on the 'Grayhole' classification. The prediction probabilities on the left side indicate that the model has predicted with absolute certainty, meaning with a probability of 1.00, that the instance belongs to the 'Grayhole' class, assigning zero probability to all other classes. On the right side of the Figure, a list of features is provided, along with their corresponding values that have contributed to this classification. These features have varying levels of importance, as suggested by their values. Features, such as 'ADV_S', with a high positive value are strongly indicative of the 'Grayhole' class, whereas others, like 'SCH_S', with a negative value are less indicative of this class. This list is crucial for understanding which attributes the model considers important when determining that an instance is a 'Grayhole'. Furthermore, some features are highlighted in orange, likely indicating their particular significance in the classification decision for this instance. For example, 'ADV_S' and 'SCH_R' are quite high in their respective positive and negative values, implying a strong influence on the 'Grayhole' prediction. This visualization of feature importance is a key component of explainable AI, shedding light on the model's decision-making process and providing insight into the predictive relationships within the data.

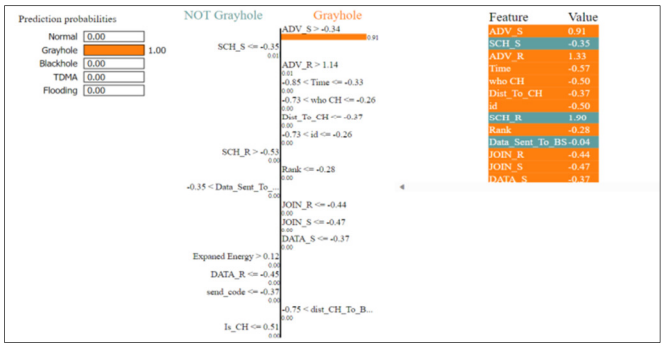

Fig. 4. Feature importance ranking for the 'Grayhole' class.

For the WSN-BFSF dataset, there is a range of results regarding the performance of the ML models. The LR shows a decent performance with an accuracy of roughly 78.9%. It is particularly effective in identifying the 'Flooding' class, with high precision and perfect recall, indicating that it almost always identifies this class correctly. However, it struggles with the 'Normal' class, as seen by the lower recall, suggesting that it often misclassifies 'Normal' instances as something else, which is further evidenced by the confusion matrix, where a substantial number of 'Normal' instances are classified as 'Blackhole'. The KNN improves upon LR, achieving an accuracy of about 87.7%. The KNN performs notably better across all classes, with impressive precision and recall, especially for 'Forwarding', indicating a strong ability to accurately identify this class. Nonetheless, it also presents a difficulty with the 'Normal' class, but to a lesser extent than the LR. The confusion matrix supports this, with fewer misclassified 'normal' instances compared to the LR. The SVM presents a further improvement in precision and recall for most classes, achieving an accuracy of approximately 83.3%. It performs exceptionally well with the 'Flooding' class, much like the LR, but also shows a balanced performance across other classes. The recall for 'Normal' is higher than that of the LR but still not as high as for other classes, suggesting some difficulty in accurately identifying 'Normal' instances. The DT and GB models exhibit nearly perfect accuracy, precision, recall, and F1-scores, with the DT slightly outperforming the GB. These models display an almost flawless classification of the dataset with negligible misclassification, as evidenced by their confusion matrices. The performance of the DT and GB indicates a very strong fit to the data, which might suggest overfitting given their near perfect metrics. Across all models, 'Flooding' is consistently the easiest class to predict, while 'Normal' proves to be the most challenging. The high accuracy of the DT and GB indicates that they have learned the decision boundaries for this dataset exceptionally well, but such near-perfect performance raises questions about their generalizability to unseen data. In contrast, the LR and SVM, with their respective modest accuracies, might offer more generalizable models, albeit at the cost of some misclassification. The KNN serves as a middle ground with good accuracy and better generalization than the DT and GB, based on its performance metrics. Ultimately, the selection of a model would depend on the requirements of the given task, weighing the importance of generalization and the risk of overfitting. The ensemble methods applied to the second

dataset achieved remarkable performance, with both soft and hard ensemble techniques nearing or reaching perfect classification metrics. The soft ensemble method attains an accuracy close to 100%, with precision, recall, and F1-scores hitting the maximum for all classes. This ensemble method effectively integrates the individual models' predictions, likely through a weighted average based on prediction confidence, which compensates for the weaknesses of each individual model. However, the confusion matrix reveals a slight imperfection: two instances of the 'Blackhole' class were misclassified. Despite this, the overall predictive power of the soft ensemble method remains exceptionally high. The hard ensemble method, on the other hand, achieves absolute perfection, with an accuracy of 100%. Each class has precision, recall, and F1-scores of 1.00, indicating flawless classification with no misclassified instances. In the hard voting ensemble, the final class prediction is determined by the mode of the predictions of the individual models. This means that the hard ensemble method has consistently produced a unanimous prediction across all models for each instance in the dataset, a strong indicator that the models agree on the decision boundaries within the data. The impeccable performance of the hard ensemble method could raise some concerns about overfitting, especially if the dataset lacks diversity or if the models have been exposed to all possible variations within the data. Although these results are impressive, they would need to be validated on a separate, unseen test set to ensure that the models generalize well to new data. Figure 5 shows the accuracy of the applied ML models on the WSN-BFSF dataset.

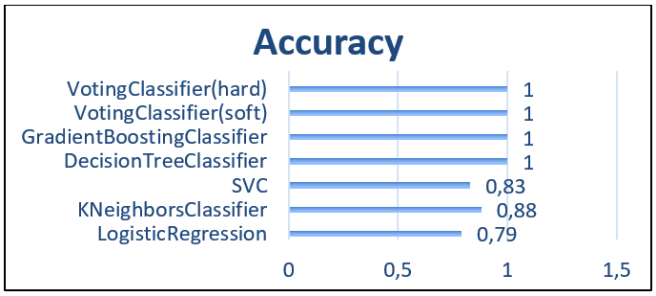

Fig. 5. Accuracy of applied ML models on the WSN-BFSF dataset.

Figure 6 illustrates the prediction probabilities and feature importance for a model's classification decision regarding network traffic, specifically whether the traffic involves 'Forwarding'. The prediction probabilities demonstrate that the model is highly confident (0. 96) that the current instance is related to the 'Forwarding' activity, with very low probabilities being assigned to the 'Normal', 'Flooding', and 'Blackhole' activities. On the right side, the feature importance chart lists the attributes that the model has used to make its decision, ranked by their impact. The positive values indicate features that support the 'Forwarding' classification, while the negative values suggest features that are against it. For instance, 'Hop_Count' with a high positive value strongly supports the 'Forwarding' class, whereas 'Rest_Energy' with a negative value suggests that the instance is less likely to be classified as 'Forwarding'. The 'Dest_Node_Num' feature has the most positive influence, while 'TTL' has the most negative influence, according to the model. This information is valuable for understanding the model's reasoning and can be used to interpret the model's decisions in a real-world context, such as in a network security analysis or in traffic management. It provides transparency into the classification process, which is essential for building trust in automated systems.

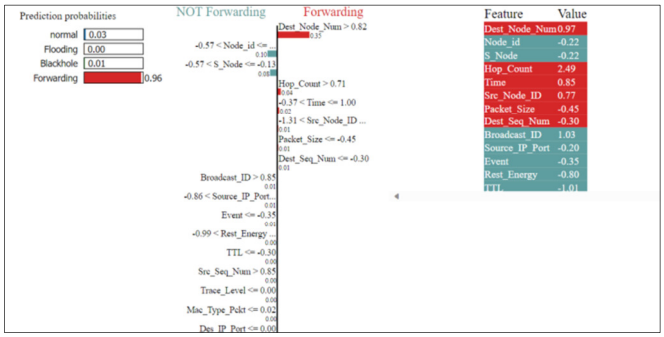

Fig. 6. Feature importance ranking for the 'Forwarding' class.

Figure 7 depicts the model's prediction probabilities regarding network traffic classification. The model predicts with perfect certainty that the network traffic is "Normal.". In addition, the feature importance for the 'Normal' classification is displayed on the right side of the Figure.

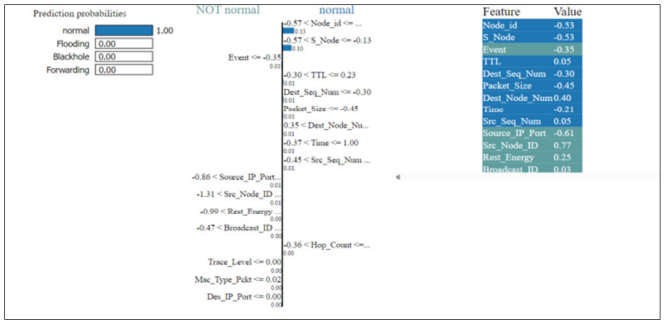

Fig. 7. Feature importance ranking for the 'Normal' class.

V. COMPARISON

Table III compares the performance of the hard and soft ensemble methods on WSN-DS and WSN-BFSF datasets. With an accuracy of 98.12%, the soft ensemble technique slightly outperformed the hard ensemble technique for the WSN-DS dataset, which yielded an accuracy of 97.97%. The small difference indicates that while both ensemble techniques are very successful with this dataset, the soft ensemble technique might be more adept at managing the subtleties of the data. When there is diversity in the predictability of the outcomes, the soft ensemble usually takes into account the confidence of the predictions from various models, which occasionally produces more accurate results. In contrast, for the WSN-BFSF dataset, the hard ensemble method reached an accuracy of 99.967%, with the soft ensemble method achieving a perfect accuracy of 100%. The hard ensemble method, which likely uses a majority voting scheme, was almost perfect, indicating that the models that make up the ensemble were mostly in

agreement on/regarding the predictions. However, the soft ensemble's perfect score implies that when the models' confidence levels were considered, the ensemble was able to correct any remaining errors, achieving flawless classification. The fact that both datasets have such high accuracy with both ensemble methods indicates that the models are very well suited to the data, and the ensemble methods are effectively leveraging their collective predictive power. However, such high accuracies, especially the 100% accuracy on the WSN-BFSF dataset, might raise questions about the complexity of the dataset, the possibility of overfitting, or whether the ensemble methods would maintain such high performance on unseen data. It is essential to ensure that the models do not simply memorize the data, but can generalize from the patterns they have learned when they encounter new, unseen data. This can be tested through a cross-validation or by evaluating the models on a separate test set that was not used during the training process. Furthermore, Table IV presents the performance of the proposed method in comparison to two different studies that have deployed the WSN-DS dataset.

TABLE III. RESULTS OF THE PROPOSED METHOD ON WSN-DS AND WSN-BFSF DATASETS

Dataset	Hard Ensemble	Soft Ensemble
WSN-DS dataset	97.97%	98.12%
WSN-BFSF dataset	99.967%	100%

TABLE IV. TWO STUDIES USING WSN-DS DATASET COMPARED TO THE PROPOSED METHOD

Method	Description	Results	Limitations
[22]	Homogeneous ensemble with Hoeffding Adaptive Tree (HAT) algorithm and heterogeneous ensemble consisting of an Adaptive Random Forest (ARF) combined with the HAT algorithm	Homogeneous: 96.84% and heterogeneous: 97.2 %	Higher computational load and lower accuracy than the proposed method
[23]	Various DL models	98.79%	DL causes more overhead, which is not suitable for working on WSNs
This study	Proposed method	97.97% and 98.12%	For future work, DL combined with ensemble techniques may be used

VI. CONCLUSION

Wireless Sensor Networks (WSNs) are increasingly becoming an essential component in many applications. This study focused on the integration of Machine Learning (ML) into the essential field of WSN security to reduce the impact of Denial of Service (DoS) attacks. The studies of various ML techniques for DOS attack detection have shown promise in significantly improving the WSN security posture. This study investigates the detection and mitigation of DoS attacks employing several ML methodologies. The WSN Dataset (WSN-DS) and WSN Blackhole, Flooding, and Selective Forwarding (WSN-BFSF) dataset were used as two distinct datasets to test the efficacy of hard and soft ensemble techniques, and the results demonstrated that their overall accuracy is increased by combining predictions from different models. With an accuracy of 98.12%, the WSN-DS dataset was slightly more accurate with the soft ensemble technique than with the hard ensemble technique, which achieved an accuracy of 97.97%. For the WSN-BFSF dataset, the soft ensemble technique achieved 100% perfect accuracy, while the hard ensemble technique achieved 99.967% accuracy.

REFERENCES

[1] K. Gulati, R. S. Kumar Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in Internet of Things (IoT)," *Materials Today: Proceedings*, vol. 51, no. 1, pp. 161–165, Jan. 2022, https://doi.org/10.1016/j.matpr.2021.05.067.

[2] I. Ali, I. Ahmedy, A. Gani, M. U. Munir, and M. H. Anisi, "Data Collection in Studies on Internet of Things (IoT), Wireless Sensor Networks (WSNs), and Sensor Cloud (SC): Similarities and Differences," *IEEE Access*, vol. 10, pp. 33909–33931, 2022, https://doi.org/10.1109/ACCESS.2022.3161929.

[3] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of Wireless Sensor Networks: An Up-to-Date Survey," *Applied System Innovation*, vol. 3, no. 1, Mar. 2020, Art. no. 14, https://doi.org/10.3390/asi3010014.

[4] F. Alawad and F. A. Kraemer, "Value of Information in Wireless Sensor Network Applications and the IoT: A Review," *IEEE Sensors Journal*, vol. 22, no. 10, pp. 9228–9245, May 2022, https://doi.org/10.1109/JSEN.2022.3165946.

[5] M. A. Elsadig, "Detection of Denial-of-Service Attack in Wireless Sensor Networks: A Lightweight Machine Learning Approach," *IEEE Access*, vol. 11, pp. 83537–83552, Aug. 2023, https://doi.org/10.1109/ACCESS.2023.3303113.

[6] M. N. U. Islam, A. Fahmin, Md. S. Hossain, and M. Atiquzzaman, "Denial-of-Service Attacks on Wireless Sensor Network and Defense Techniques," *Wireless Personal Communications*, vol. 116, no. 3, pp. 1993–2021, Feb. 2021, https://doi.org/10.1007/s11277-020-07776-3.

[7] A. Cetinkaya, H. Ishii, and T. Hayakawa, "An Overview on Denial-of-Service Attacks in Control Systems: Attack Models and Security Analyses," *Entropy*, vol. 21, no. 2, Feb. 2019, Art. no. 210, https://doi.org/10.3390/e21020210.

[8] S. Balaji and T. Sasilatha, "Detection of denial of service attacks by domination graph application in wireless sensor networks," *Cluster Computing*, vol. 22, no. 6, pp. 15121–15126, Nov. 2019, https://doi.org/10.1007/s10586-018-2504-5.

[9] M. Premkumar and T. V. P. Sundararajan, "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks," *Microprocessors and Microsystems*, vol. 79, Nov. 2020, Art. no. 103278, https://doi.org/10.1016/j.micpro.2020.103278.

[10] A. Huseinović, S. Mrdović, K. Bicakci, and S. Uludag, "A Survey of Denial-of-Service Attacks and Solutions in the Smart Grid," *IEEE Access*, vol. 8, pp. 177447–177470, Sep. 2020, https://doi.org/10.1109/ACCESS.2020.3026923.

[11] E. Suryaprabha and N. M. Saravana Kumar, "Enhancement of security using optimized DoS (denial-of-service) detection algorithm for wireless sensor network," *Soft Computing*, vol. 24, no. 14, pp. 10681–10691, Jul. 2020, https://doi.org/10.1007/s00500-019-04573-4.

[12] M. A. Elsadig, A. Altigani, and M. Abuelaila, "Security Issues and Challenges on Wireless Sensor Networks," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 4, pp. 1551–1559, Aug. 2019, https://doi.org/10.30534/ijatcse/2019/78842019.

[13] M. B. Apsara, P. Dayananda, and C. N. Sowmyarani, "A Review on Secure Group Key Management Schemes for Data Gathering in Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 1, pp. 5108–5112, Feb. 2020, https://doi.org/10.48084/etasr.3213.

[14] R. Wazirali and R. Ahmad, "Machine Learning Approaches to Detect DoS and Their Effect on WSNs Lifetime," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4922–4946, Oct. 2021, https://doi.org/10.32604/cmc.2022.020044.

[15] S. Ismail, Z. El Mrabet, and H. Reza, "An Ensemble-Based Machine Learning Approach for Cyber-Attacks Detection in Wireless Sensor Networks," *Applied Sciences*, vol. 13, no. 1, Jan. 2023, Art. no. 30, https://doi.org/10.3390/app13010030.

[16] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, "An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs," *Sensors*, vol. 22, no. 4, Feb. 2022, Art. no. 1407, https://doi.org/10.3390/s22041407.

[17] K. Lakshmi Narayanan, R. Santhana Krishnan, E. Golden Julie, Y. Harold Robinson, and V. Shanmuganathan, "Machine Learning Based Detection and a Novel EC-BRTT Algorithm Based Prevention of DoS Attacks in Wireless Sensor Networks," *Wireless Personal Communications*, vol. 127, no. 1, pp. 479–503, Nov. 2022, https://doi.org/10.1007/s11277-021-08277-7.

[18] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks," *Journal of Sensors*, vol. 2016, no. 1, 2016, Art. no. 4731953, https://doi.org/10.1155/2016/4731953.

[19] M. Dener, C. Okur, S. Al, and A. Orman, "WSN-BFSF: A New Data Set for Attacks Detection in Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2109–2125, Jan. 2024, https://doi.org/10.1109/JIOT.2023.3292209.

[20] T. M. Barros, P. A. Souza Neto, I. Silva, and L. A. Guedes, "Predictive Models for Imbalanced Data: A School Dropout Perspective," *Education Sciences*, vol. 9, no. 4, Nov. 2019, Art. no. 275, https://doi.org/10.3390/educsci9040275.

[21] D.-X. Zhou, "Theory of deep convolutional neural networks: Downsampling," *Neural Networks*, vol. 124, pp. 319–327, Apr. 2020, https://doi.org/10.1016/j.neunet.2020.01.018.

[22] H. Tabbaa, S. Ifzarne, and I. Hafidi, "An Online Ensemble Learning Model for Detecting Attacks in Wireless Sensor Networks," *Computing and Informatics*, vol. 42, no. 4, pp. 1013–1036, Dec. 2023, https://doi.org/10.31577/cai_2023_4_1013.

[23] S. Salmi and L. Oughdir, "Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network," *Journal of Big Data*, vol. 10, no. 1, Feb. 2023, Art. no. 17, https://doi.org/10.1186/s40537-023-00692-w.