International Journal of Computers and Informatics

Journal Homepage: https://www.ijci.zu.edu.eg

Int. j. Comp. Info. Vol. 2 (2024) 39–52

Paper Type: Original Article

Hybrid LSTM-Random Forest for Intrusion Detection in Wireless Sensor Networks: Enhanced DoS Classification with Explainable AI

Ahmed Tolba [1,*]

[1] Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt; a.tolba24@fci.zu.edu.eg.

Received: 01 Dec 2023 **Revised**: 01 Feb 2024 **Accepted**: 26 Feb 2024 **Published**: 28 Feb 2024

Abstract

Wireless Sensor Networks (WSN) have emerged as one of the most active study topics in computer science due to their vast range of applications, which include crucial military and civilian uses. To ensure the security and dependability of WSN services, an Intrusion Detection System (IDS) should be implemented. This IDS must be compatible with the features of WSNs and capable of identifying the greatest number of security risks. Using the WSN dataset, this research proposes a new hybrid model that combines LSTM and Random Forest to help detect and categorize four forms of Denial of Service (DoS) attacks: blackhole, grayhole, flooding, and scheduling. The "proposed model" surpasses LSTM, GRU, RNN, CNN, CNN-LSTM, Random Forest, GaussianNB, and Decision Tree in attack detection, as indicated by the highest accuracy, precision, recall, F1-Score, and AUC accuracy score of 0.996, 0.98, 0.98, 0.98, and 0.99, respectively. By offering insights into the decision-making process and facilitating a better comprehension of the feature contributions to attack detection, the application of Explainable Artificial Intelligence (XAI) approaches to the Random Forest model analysis enhanced the interpretability of the results.

Keywords: Wireless Sensor Networks; Intrusion Detection; XAI; Machine Learning; Deep Learning.

1 | Introduction

The rise of ubiquitous computing has increased reliance on advanced technologies in a variety of industries, including people, huge organizations, and government agencies. This greater dependence has increased Internet transactions and information exchange. As a result, maintaining this ever-changing landscape presents substantial issues for intrusion detection systems (IDS). As malicious attacks become more complex, detecting unknown and obfuscated malware becomes increasingly challenging, as malware developers try to avoid detection and reverse engineering by IDS [1].

Wireless Sensor Networks (WSN) have emerged as one of the most active study topics in computer science due to their vast range of applications, which include crucial military and civilian uses. Such applications have introduced a variety of security risks, particularly in unsupervised contexts. To ensure the security and dependability of WSN services, an Intrusion Detection System (IDS) should be implemented. This IDS must be compatible with the features of WSNs and capable of identifying the greatest number of security risks [2].

Corresponding Author: a.tolba24@fci.zu.edu.eg

An Intrusion Detection System (IDS) is a system that analyzes network traffic for unusual activity and alerts managers immediately when it is identified. It is a software application that searches networks or systems for potential security threats or policy breaches. When such activities are discovered, they are typically reported to a system administrator or combined into a centralized Security Information and Event Management (SIEM) system. The SIEM integrates data from many sources and uses alert filtering techniques to discriminate between genuine threats and false positives.

There are two types of intrusion detection systems (IDS): host-based (HIDS) and network-based (NIDS). A HIDS is deployed on a specific endpoint to protect it from both internal and external threats. This system can track incoming and outgoing network traffic, inspect running processes, and examine system logs. While a HIDS is only visible to its host system, restricting the context accessible for decision-making, it gives detailed information about the host's internal workings. A NIDS, on the other hand, is intended to manage the entire protected network by monitoring all traffic and making choices based on packet information and content. This broader perspective provides more information and improves the identification of common threats; yet, NIDS lacks visibility into the internal activities of the endpoints they defend [3].

Several strategies exist for detecting attacks, including Signature-based Intrusion Detection Systems (SIDS) [4]. SIDS uses matching techniques to recognize previously recorded intrusions. Essentially, when a detected intrusion signature matches one from the existing signature database, an alarm is raised. In a study by Díaz-Verdejo [5], three SIDS were evaluated for their ability to identify online attacks with high precision. The data revealed that SIDS's greatest detection rate is insufficient for effective protection and falls short of expectations for known threats. The efficiency of each detector is substantially determined by the established settings, which affect both detection capabilities and the incidence of false alarms. The report recommends the usage of open-source SIDS with default configurations to protect against web threats. Traditional IDS, which are knowledge- or signature-based, struggle to cope with quickly expanding networks and fail to control threats as their volume, complexity, and diversity increase [1].

Furthermore, the firewall prevents illicit access to the entire network. It has been proved that the firewall and its versions can be easily circumvented by attackers, for example, by utilizing a bogus source address. Furthermore, other attempts, including DoS and DDoS, went undetected. This highlighted the need to create and obtain IDS [6].

Several critical aspects influence the evolution of intrusion detection systems (IDS), including:

- Modern networked systems are becoming more complicated, resulting in vulnerabilities that attackers can exploit.
- Many existing systems have serious security flaws, making them easy targets for hackers. Despite efforts to address these issues, complete eradication is generally impossible.
- While intrusion prevention methods exist, none can provide perfect protection. IDS remains an important technology for detecting intrusions and automatically updating security measures in response to new threats.
- Attacks by authorized users offer a significant danger and are frequently more devastating than external attacks, confounding detection attempts.
- Attackers are always developing new techniques to circumvent old detection and prevention approaches, underscoring the need for continued innovation in cybersecurity.

Given these constraints, developing a dynamic, adaptable framework that combines deep learning and machine learning could considerably improve IDS efficacy. This architecture would allow for real-time threat analysis and response, increasing resilience to both known and developing attack vectors. Furthermore, building a cyber security awareness culture inside firms can assist reduce insider threats by educating employees on security measures and encouraging vigilance. Thus, the goal of this paper is to benefit from a

hybrid Long Short-Term Memory (LSTM) and Random Forest on the WSN dataset to categorize normal and four forms of Denial of Service (DoS) attacks: blackhole, grayhole, flooding, and scheduling. We evaluated our suggested model by comparing it to CNN and LSTM, GRU, CNN-LSTM - RNN - GaussianNB, Random Forest, and Decision Tree (DT).

The main contribution :

- A new hybrid DL model LSTM and Random forest is proposed.
- We have evaluated our proposed model with multiple evaluation metrics to prove the validity of the model.
- The proposed model is compared with several DL and ML models.

The remaining paper is divided as follows. Section 2 gives the background information required for this study. Section 3 describes the methodology of this investigation. Section 4 presents the proposed model. Section 5 presents the experimental outcomes. Section 6 shows the conclusion and future directions of this study.

2 |Literature Review

Shahin et al. [7] explored cybersecurity concerns in the Industrial Internet of Things (IIoT), with an emphasis on botnet assaults. They tested more than 25 Machine Learning algorithms on seven IoT devices, and several models obtained near-perfect detection performance. This study identifies high-performing botnet detection models and underlines the significance of verifying them against new datasets. Future research should look into how device type and function influence detection efficacy.

Sowmya and Mary Anita [8] examined AI-based intrusion detection mechanisms and discovered that they outperformed traditional methods in detection and classification. They emphasized the importance of feature reduction and reported that ML, DL, and ensemble approaches outperformed each other by more than 99%. The study concentrated on common threats such as DoS and R2L, advising further research to assess additional metrics and investigate hybrid approaches. They also stressed the importance of AI-based systems in addressing unforeseen assaults, while noting limitations due to the limited amount of datasets used.

Mohammed Sayeeduddin Habeeb and T. Ranga Babu [9] emphasized the growing necessity for effective Intrusion Detection Systems (IDS) to protect data from various assaults as internet usage increases. Their study provides a detailed survey of network intrusion detection systems (NIDS) that use machine learning (ML) and deep learning (DL) methodologies, analyzing classification based on detection accuracy and model complexity. They discovered that DL approaches beat ML in terms of accuracy and false alarm rates, even though DL needs more processing time in real-time implementations. The paper criticizes the use of obsolete datasets such as KDDCup99 and NSL KDD and suggests that newer datasets such as CICIDS2017 and AWID2018 provide superior performance. Overall, the research emphasizes major difficulties, such as improving detection for minority threat classes, and provides insights into prospective network security solutions.

Shraddha Mane and Dattaraj Rao [10] investigate the issues of cybersecurity, emphasizing the need for Intrusion Detection Systems (IDS) as attackers adopt new patterns. They emphasize the efficiency of machine learning and deep learning models in improving detection rates but also point out that deep neural networks are more complex and difficult to understand. To address these issues, the authors offer an explainable AI architecture that improves transparency in the machine learning pipeline by leveraging algorithms such as SHAP and LIME to describe the reasons driving cyberattack predictions. Their technique is tested on the NSL KDD dataset, confirming its effectiveness in making IDS more interpretable.

Sampath Rajapaksha et al. [11] investigate the weaknesses of the Controller Area Network (CAN), the most used in-vehicle communication protocol, due to a lack of strong security mechanisms such as message authentication and encryption. They emphasize the effectiveness of AI-based Intrusion Detection Systems

(IDSs) as a countermeasure to automobile cyberattacks. The study examines AI-based in-vehicle IDS developments from 2016 to August 2022, presenting a novel taxonomy of detection algorithms, attack kinds, characteristics, and benchmark datasets. The authors also examine the security of AI models, define the procedures required for constructing AI-based IDSs for the CAN bus, identify shortcomings in existing techniques, and suggest future research areas.

Shruti Patil et al. [12] examine the role of Intrusion Detection Systems (IDS) in cybersecurity, emphasizing their significance in reducing attacks on computer networks. The research offers a novel IDS that uses machine learning ensemble approaches to improve classification accuracy and reduce false positives, based on characteristics from the CICIDS-2017 dataset. The authors integrate multiple machine learning techniques, such as decision trees, random forests, and support vector machines (SVM), with a voting classifier to achieve an accuracy of 96.25%. Furthermore, they use the Explainable AI (XAI) algorithm LIME to improve the model's interpretability and overcome the limitations of the black-box approach. The experimental results show that LIME delivers clearer explanations and is more responsive to intrusion detection.

Ashish Rathee et al. [13] use deep learning (DL) approaches to mitigate cyberattack risks. They investigate a variety of AI models, including deep neural networks, shallow neural networks, convolutional neural networks, and attention processes, evaluating alternative designs and depths. Using a checkpoint process, they choose the models with the highest accuracy. The models are evaluated on a variety of datasets, including NSL-KDD, Kyoto, and UNSW-NB15. The report finishes with a comparative analysis that demonstrates the effectiveness of their proposed methodology in improving cybersecurity.

Patrick Vanin et al. [14] highlight the difficulty of intrusion detection as data transmission rates rise and novel attacks on data security emerge. They emphasize the importance of upgraded Intrusion Detection Systems (IDS) that improve detection accuracy and lower false alarm rates, particularly for zero-day assaults. The study describes a taxonomy of machine learning algorithms used in IDS and discusses recent implementations, including their merits, shortcomings, and datasets used. The authors finish by discussing research problems and future developments in the field of IDS.

Kumar A. Shukla et al. [15] seek to create energy-efficient machine-learning models for detecting assaults in IoT networks. They underline the importance of collecting both regular and attack data from the IoT environment for developing their model. The study assesses the possibilities of several algorithms, such as Bayesian Networks, Artificial Neural Networks (ANN), and Support Vector Machines (SVMs). They explicitly examine a standard three-layer ANN in real-world circumstances, utilizing roundtrip time and power consumption measurements.

In conclusion, major obstacles still exist even if AI-based solutions have proven successful in boosting cybersecurity. Although there are newer, underutilized datasets like CICIDS2017 and AWID2018, the model's robustness is limited by its reliance on outdated datasets like KDDCup99 and NSL-KDD. It is also difficult to detect minority threat classes because most models concentrate on majority-class threats, which may lead to the failure of uncommon but important attacks. Further research into explainable AI (XAI) is necessary since deep learning models' intricacy and interpretability make it difficult to use them in real time. Last but not least, there is still much to learn about the impact of different kinds of devices, particularly in IoT and automotive systems like CAN, on the efficacy of detection. By filling in these gaps, intrusion detection systems (IDS) could be greatly improved.

3 | Methodology

In this section, we provide some preliminaries of the hybrid DL model proposed model.

3.1 LSTM

LSTM is a form of RNN that can recognize long-term dependencies. The conventional LSTM has three gates that regulate information and pass it on to the next unit. The forgotten value either forgets all or none of the

information, depending on the value of the forget gate. The input gate contains two halves that regulate the new information needed to add the next cell state. The first component of the input gate is the sigmoid layer, which controls the output value stored in the cell state. The Tanh layer is the input gate's second component, producing a vector of new feature values that are saved in the cell state. The output gates supply the most recent cell state information. Statistics are executed selectively through the gates' structure and are transmitted through to update and store historical statistics, as well as update the cell state. The LSTM assesses previous historical values, evaluates current unknown patterns by altering itself based on the full pattern set, and forecasts future events [16]. The LSTM architecture is illustrated in Figure 1.

$$\text{LSTM Formulation} = \begin{aligned} f_t &= \sigma(\boldsymbol{W}_f \boldsymbol{x}_l + \boldsymbol{U}_f \boldsymbol{h}_{l-1} + \boldsymbol{b}_f), \\ \check{\boldsymbol{c}}_l &= \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{l-1} + \boldsymbol{b}_c), \\ \boldsymbol{i}_l &= \sigma(\boldsymbol{W}_i \boldsymbol{x}_l + \boldsymbol{U}_i \boldsymbol{h}_{l-1} + \boldsymbol{b}_i), \\ \boldsymbol{c}_l &= \boldsymbol{f}_t \odot \boldsymbol{c}_{l-1} + \boldsymbol{i}_l \odot \check{c}_t, \\ \boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{x}_l + \boldsymbol{U}_o \boldsymbol{h}_{l-1} + \boldsymbol{b}_o), \\ \boldsymbol{h}_l &= \boldsymbol{o}_l \odot \tanh(\boldsymbol{c}_l), \end{aligned} \quad (1)$$

Where: $\boldsymbol{x}_t$ Is the input at time t step , $\odot$ is the element-wise dot product, $\boldsymbol{i}_l$, $\boldsymbol{o}_t$, f_t Is the input gate, output gate and forget gate respectively, $\boldsymbol{c}_l$ is the cell state, W,U,b model Parameter

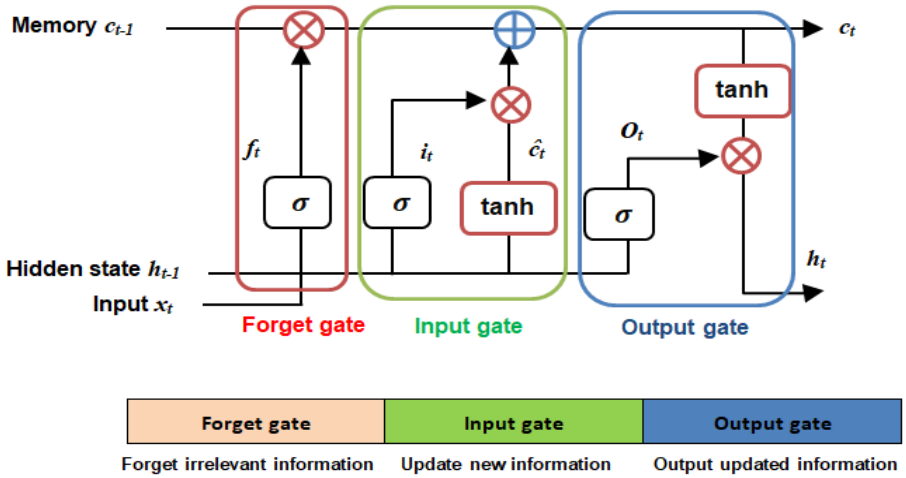


Figure 1. LSTM architecture.

3.2 Random Forest

Random Forest is an ensemble learning strategy that uses the combined power of several decision trees to improve classification accuracy. Ensemble classification is the use of many classifiers that work together to identify the class label of a new, unlabeled data point. Random Forest, for example, generates many randomized decision trees during training and then aggregates their predictions, sometimes using averaging or majority voting. This technique has sparked widespread interest in the research community due to its high accuracy, resistance to overfitting, and improved performance across multiple domains. Each decision tree in the forest is trained on a random subset of the data, and each node only considers a random subset of features for splitting. This randomization introduces variation within the trees, which is critical for increasing the model's overall resilience and generalizability [17]. The Random Forest architecture is illustrated in Figure 2.

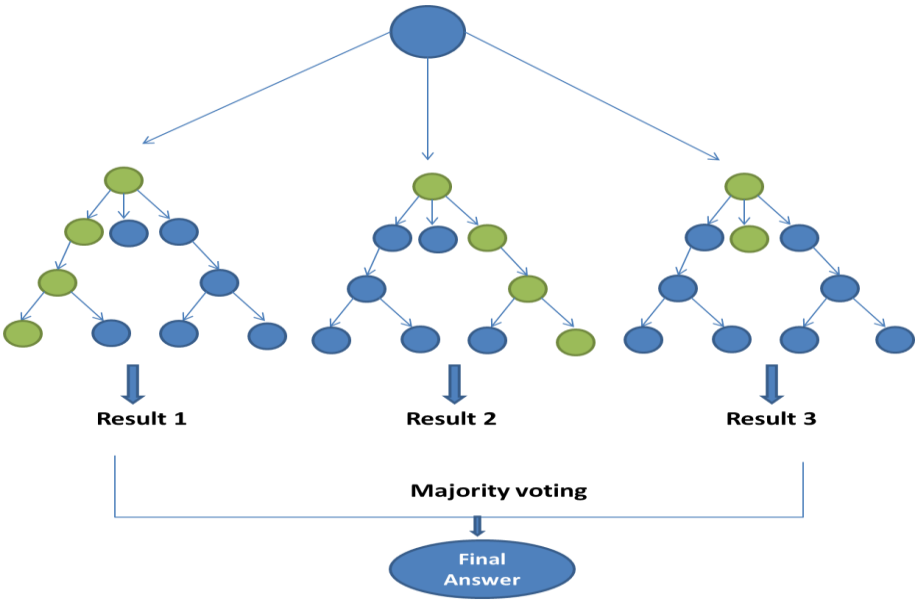


Figure 2. Random Forest architecture.

3.3 Interpretable AI

A key component of machine learning is interpretable AI, which aims to clarify how complicated models make decisions. This is especially important in high-stakes industries like cybersecurity, banking, and healthcare. Transparency and accountability are vital as these models impact important outcomes more and more. By enabling stakeholders to understand the reasoning behind model predictions, interpretable AI solutions promote confidence and aid in well-informed decision-making. When addressing any biases or inaccuracies in model outputs that could have important ramifications, this is especially pertinent [18].

One of the most effective techniques for obtaining interpretability in machine learning is SHAP (Shapley Additive explanations). Shapley values are used by SHAP, which has its roots in cooperative game theory, to fairly assign the contribution of each characteristic to a model's prediction. SHAP guarantees that feature importance is determined fairly by averaging the contributions of all conceivable feature combinations. This additive trait provides insights that are critical for stakeholder assurance and model validation by enabling a clear grasp of how individual attributes influence particular predictions [19].

SHAP values are based on game theory and assign an importance value to each feature in a model. Features with positive SHAP values positively impact the prediction, while those with negative values have a negative impact. The magnitude is a measure of how strong the effect is. Applying SHAP improves interpretability locally by providing explanations for specific predictions instead of a comprehensive comprehension of the model. SHAP values measure the influence of attributes for each occurrence, indicating positive and negative contributions to a baseline prediction. SHAP provides visualizations like dependence and summary plots. This makes SHAP an essential tool for building confidence and enabling the ethical deployment of AI systems.

4 | Proposed Model

Network security is an essential element when transmitting data over the Internet. Cyber-attacks are becoming more widespread in the IoT ecosystem as security measures are decreased. These present models have numerous drawbacks, including reduced detection accuracy, a lack of taxonomy, etc. This research study presents a novel IDS model for intrusion detection systems in wireless sensor networks that combines deep learning and machine learning approaches. The proposed hybrid model allows for automatic assault classification with high accuracy and minimal errors. The proposed hybrid model for Intrusion Detection Systems (IDS) combines deep learning and machine learning methodologies to improve cyber-attack

detection in wireless sensor networks. It uses an LSTM network for feature extraction, which captures temporal dependencies in the data. The architecture starts with an LSTM layer, followed by a dense layer and a dropout for regularization. The collected features are then fed into a Random Forest classifier, which determines the final attack categorization. This hybrid model combines the capabilities of LSTM's sequential data handling and Random Forest's robust classification, resulting in high accuracy and lower errors in detecting cyber risks. The systematic representations are shown in Figure 3.

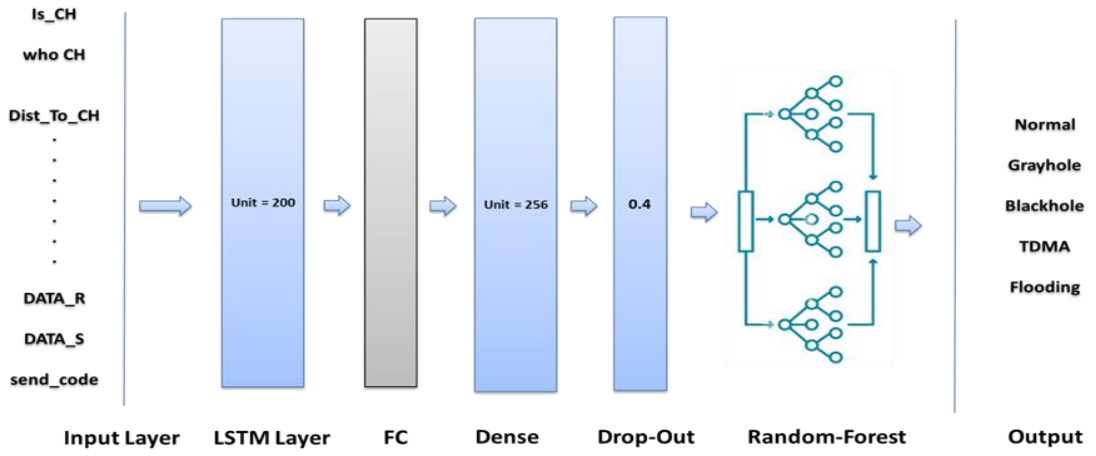


Figure 3. Proposed model architecture.

4.1 Training Deep Learning Models

Deep learning models were trained for 30 epochs. In addition, in our tests, we used the mini-batch gradient descent technique with a batch size of 500 to reduce the error estimated from the loss function (Categorical Cross Entropy) Eq. (2). The Adam optimizer is used to improve the classification accuracy of deep learning models by optimizing their weights. Each epoch divides the data into 503 batches and updates the weights in each batch. This indicates that in each epoch, the weights change 503 times, according to the number of batches.

Where y_i represent real values and $\check{y}_i$ Represent predicted values.

$$\text{Minimize: } \text{loss} = -\sum_{i=1}^{M} y_i \cdot \log \check{y}_i \quad (2)$$

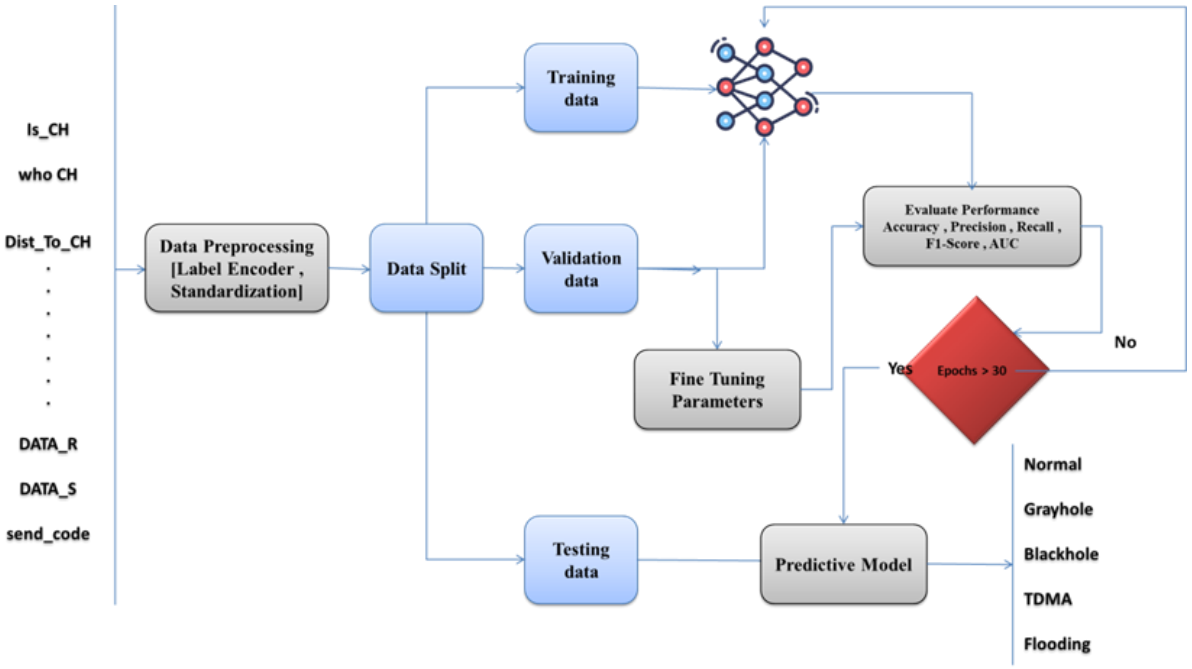


Figure 4. Deep learning pipeline for detecting attacks.

The Wireless Sensor Networks dataset is initially processed by transforming categorical data to numerical format using a label binarizer. Normalization is then accomplished using the StandardScaler, as shown in Eqs. (3-4). Where Normalizing input data to a consistent scale is critical for improving the convergence speed and reliability of the optimization procedure.

$$Mean = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3}$$

$$Std = \sqrt{\frac{\sum_{i=1}^{n} x_i - Mean}{n}} \tag{4}$$

The dataset is divided into three independent subsets: training, validation, and testing. Initially, 67% of the data is set aside for training, with the remaining 33% retained for subsequent splitting. Half of this 33% is set aside for testing, with the other half reserved for validation. This means that 67% of the data is utilized to train the model, with the remaining 16.5% used for validation and testing. The training set facilitates model learning, whereas the validation set is used to fine-tune hyperparameters and prevent overfitting. Finally, the testing set is used to assess the model's performance on previously unseen data, guaranteeing its generalizability. The overall DL pipeline in IDS is shown in Figure 4.

4.2 Training Machine Learning Models

The three machine learning algorithms—Decision Tree, Gaussian Naive Bayes, and Random Forest—were all trained using the identical Wireless Sensor Network (WSN) dataset. The dataset was separated into two parts: 70% for model training and 30% for testing, with a test split of 30%. Unlike deep learning models, which often employ validation data for adjustment, machine learning models do not require a validation set due to their simplicity. Thus, just the training and testing sets were used to assess the models' performance.

Each model was tested on its capacity to classify intrusion types, which is an important step in designing an effective Intrusion Detection System (IDS). The pipeline was created to automatically run the dataset through

each model, after which performance metrics such as accuracy, precision, recall, and F1-score were produced to evaluate their usefulness.

SHAP was used to create interpretable AI for the Random Forest model. SHAP explains the Random Forest output by assigning each feature's contribution to the final predictions, making it easier to grasp the model's underlying decision-making process. This method aids in determining which features had the greatest impact on intrusion detection, increasing the transparency and explainability of the findings.

5 | Results and Discussion

This section evaluates the performance of the proposed model on the widely used WSN-DS dataset(2), which contains 374,661 samples divided into five classes: Normal (340,066), Grayhole (14,596), Blackhole (10,049), TDMA (6,638), and Flooding (3,312). Furthermore, it is compared against a variety of deep learning models, including CNN, LSTM, GRU, RNN, CNN-LSTM, and machine learning algorithms such as Random Forest Classifier, Decision Tree, and Gaussian naive Bayes. These models are built in Python with the Kaggle platform and Keras API. The Adam optimizer was used to train the weights of the models over 30 epochs. The performance indicators used to assess the performance of the models are as follows:

- Accuracy: This metric's definition is the ratio of accurately predicted samples to all samples in a given dataset. The following equation can be used to calculate this metric:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}. \tag{5}$$

  where TP, FN, TN, and FP represent true positive, false negative, true negative, and false positive, respectively.

- Precision: A model's precision can be defined as a statistic that expresses how well it produces positive predictions. The statistic measures the proportion of correctly identified positive cases to all predicted positive cases. Using the following equation, one can calculate this metric:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{6}$$

- Recall: This measure evaluates how well the model can identify positive samples out of all the real positive samples. It is sometimes referred to as a true positive rate or sensitivity. This equation can be used to calculate this metric:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{7}$$

- F1-score: The F1 score combines precision and recall into a single metric to provide a fair assessment of model performance. This equation can be used to calculate this metric:

$$\text{F1} - \text{score} = 2 * \frac{Precision \cdot Recall}{Precision + Recal}. \tag{8}$$

- AUC: It displays the model's ability to distinguish between positive and negative examples; a higher AUC indicates better performance. This equation can be used to calculate this metric:

$$\text{AUC} = \frac{1 + \frac{\text{TP}}{\text{TP+TN}} - \frac{\text{FP}}{\text{FP+TN}}}{2}$$

- Confusion Matrix: A confusion matrix is a visual aid that offers an illustration of a machine learning or deep learning model's performance on a certain dataset. Based on the model's predictions, it shows the quantity of accurate and inaccurate occurrences. Classification model efficacy is typically evaluated using matrices like F1Score, Accuracy, Permission, and Recall that were previously discussed. The model operates at its best when all of the true values are as large as possible.

- ROC Curve: The receiver operating characteristic (ROC) curve displays a model's classification performance. It displays the specificity (1 - false positive rate) against the sensitivity (true positive rate) at different classification thresholds. Better performance is indicated by a larger ROC curve, which is used to evaluate how well the model can distinguish between positive and negative cases.

5.1 Numerical Results

Table 1 and Figure 5 provide key insights into the classification model's performance, demonstrating its ability to correctly categorize instances into each class and indicate any probable misclassifications. Figure 6 illustrates the performance evaluation of the proposed model using the accuracy, loss, and ROC curves.

Based on the data presented in Table 1 and Figure 5, Long Short-Term Memory, or LSTM, is the best deep learning model. Whereas Random Forest is one of the top machine-learning models. The hybrid LSTM-Random Forest design that is being proposed combines these two potent models to create a superior model that optimally utilizes their respective strengths. The "Proposed model" outperforms all other models in detecting attacks, as evidenced by its greatest accuracy, precision, recall, F1-Score, and AUC accuracy score of 0.996, 0.98, 0.98, 0.98, and 0.99, respectively.

The confusion matrix in Table 2 shows excellent classification performance across all classes, with high recall and precision values for the majority of them. The Normal and TDMA classes achieve near-perfect accuracy, while the Flooding class has a slightly lower precision (0.92%). The model accurately distinguishes between different attack types with minimal misclassification.

Table 1. Comparison between the proposed model and others in terms of various performance indicators.

Model Name	Accuracy	Precision macro avg	Recall macro avg	F1-score macro avg	AUC Macro avg	# Parameters
CNN-LSTM	0.9886	0.92	0.93	0.93	0.99	56,587
CNN	0.9920	0.94	0.96	0.95	0.98	27,397
LSTM	0.9928	0.95	0.96	0.95	0.98	503,141
Rnn	0.9888	0.93	0.94	0.93	0. 97	472,541
GRU	0.9891	0.93	0.93	0.93	0.96	493,241
Proposed	0.9967	0.98	0.98	0.98	0.99	503,141
Random Forest	0.97	0.98	0.98	0.98	0.98	-
Decision Tree	0.96	0.97	0.97	0.97	0.98	-
GaussianNB	0.72	0.80	0.72	0.70	0.82	-

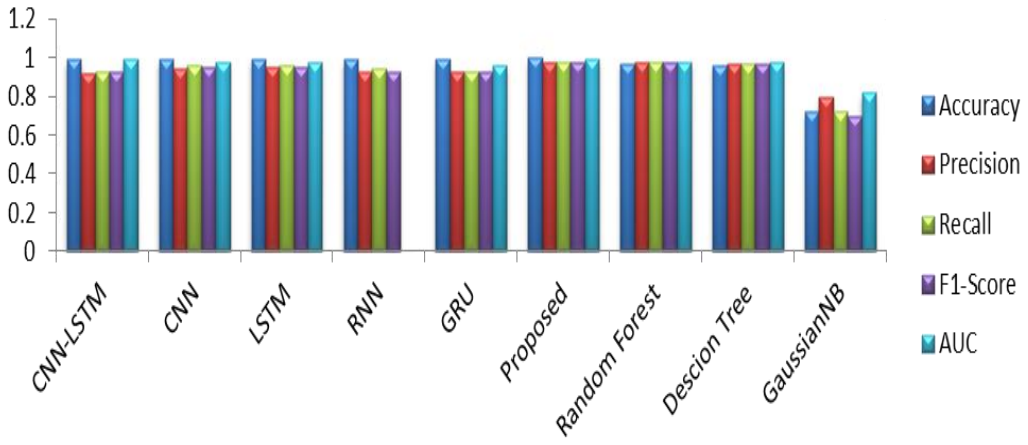


Figure 5. Comparison between the proposed model and others in terms of various performance indicators.

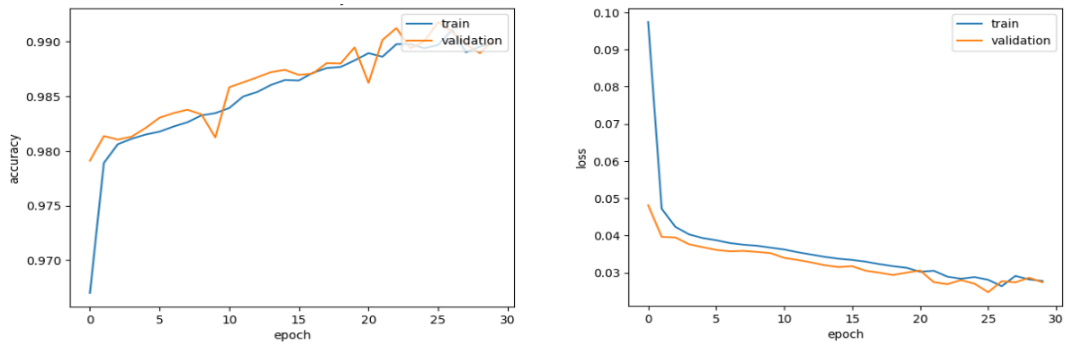
Table 2. Confusion matrix of the proposed model.

		Estimated classes					
Actual classes		Normal	Grayhole	Blackhole	TDMA	Flooding	Recall (%)
	Normal	75147	22	2	0	62	1.00 %
	Grayhole	18	3147	35	0	0	0.98 %
	Blackhole	2	18	2194	0	0	0.99 %
	TDMA	95	2	0	1381	0	0.93 %
	Flooding	11	0	0	0	702	0.98 %
Precision (%)		1.00%	0.99 %	0.98 %	1.00 %	0.92%	

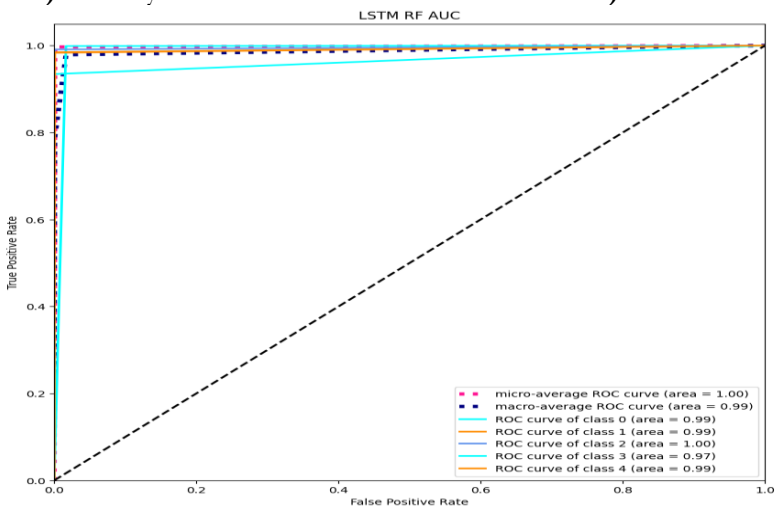
5.2 Graphical Results

According to the accuracy curve (a) and loss curve (b) in Figure 6, the learning curve depicts the evolution of the model's accuracy and loss throughout 30 epochs dedicated to training and validation. The model started with an initial training accuracy of 94.93% and a loss of 0.2094. As training progressed, the model improved continuously, obtaining a validation accuracy of 97.91% and a validation loss of 0.0481 in its first epoch. During the training process, the model constantly improved its performance. After 30 epochs, the model's training accuracy improved to 98.98%, while its loss fell to 0.0284. Simultaneously, the validation accuracy achieved 99.02%, resulting in a validation loss of 0.0274. These tendencies in the learning curves illustrate the model's ability to effectively learn and adapt to the training data. The constant improvement in accuracy and concurrent reduction in loss on both the training and validation datasets demonstrate the model's ability to achieve high levels of precision with low error, giving it a resilient and trustworthy solution for the task at hand.

The receiver operating characteristic (ROC) curve (c) of a 5-class.In Figure 6 It is seen that the macro average (AUC) attains a value of 99%, with all classes having a value greater than 97%.



a) Accuracy curve **b)** Loss curve

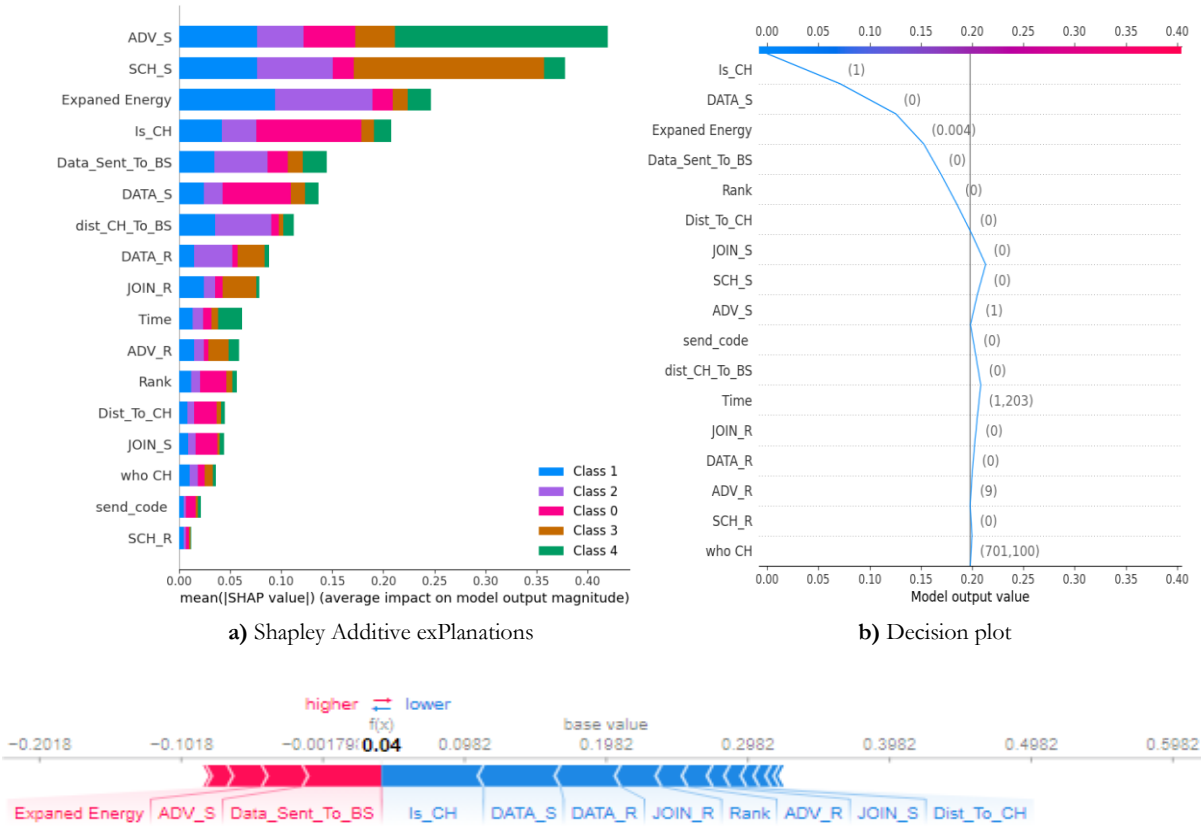


c) ROC curve

Figure 6. Performance evaluation of the proposed model under accuracy curve, loss curve and ROC curve.

5.3 XAI Results

According to Figure 7, The model's interpretability utilizing SHAP (Shapley Additive Explanations) values for five classes—Grayhole (Class 1), Blackhole (Class 2), TDMA (Class 3), Normal (Class 0), and Flooding (Class 4)—is depicted in the picture. With considerable heterogeneity in each feature's contribution to various classes, the mean SHAP values in plot (a) show that features like ADV_S and Is_CH have the most influence on the model. Using a base value as a starting point, Plot (b), or the decision plot, illustrates how these features cumulatively affect each prediction. The features that have the greatest influence on a final class are Is_CH and ADV_S. Plot (c) provides a detailed explanation of how features such as Expanded Energy and Data_Sent_To_BS influence the model's forecast towards Blackhole (Class 2) by utilizing a SHAP force plot. The contribution of each feature is represented in terms of how it affects the output. Overall, the analysis shows how essential indicators such as Is_CH, ADV_S, and Expanded Energy play a critical role in distinguishing the varied network behaviors, notably in recognizing attack patterns like Blackhole.



a) Shapley Additive exPlanations

b) Decision plot

c) SHAP Force Plot

Figure 7. SHAP Analysis for feature impact and decision interpretability.

6 | Concolusin and Future Work

This paper developed a hybrid model that combines Long Short-Term Memory (LSTM) networks and Random Forests to efficiently identify and categorize four forms of Denial of Service (DoS) assaults in Wireless Sensor Networks (WSN): blackhole, grayhole, flooding, and scheduling. The model outperformed multiple benchmark models in terms of accuracy, precision, recall, F1-Score, and AUC, with values of 0.996, 0.98, 0.98, 0.98, and 0.99. The use of Explainable Artificial Intelligence (XAI) techniques in analyzing the Random Forest model improved the results' interpretability by providing insights into the decision-making

process and allowing for a better understanding of the feature contributions to attack detection. In the future more deep learning architectures and real-time data integration should be investigated in future studies to increase the suggested hybrid model's detection accuracy and flexibility. The robustness and interpretability of the intrusion detection model will be strengthened by adding more attack scenarios to the dataset and utilizing cutting-edge XAI techniques.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Funding

This research was conducted without external funding support.

Data Availability

The implementation used in this article was in GitHub. For details, please refer to https://github.com/AhmedTolba36996/WSN-Intrusion-Detection/tree/main/Data

Conflicts of Interest

The author declares that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Silivery, A. K., Kovvur, R. M. R., Solleti, R., Kumar, L. S., & Madhu, B. (2023). A model for multi-attack classification to improve intrusion detection performance using deep learning approaches. Measurement: Sensors, 30, 100924.

[2] Almomani, I., Al-Kasasbeh, B., & Al-Akhras, M. (2016). WSN-DS: a dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors, 2016(1), 4731953.

[3] Abdulganiyu, O. H., Ait Tchakoucht, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). International journal of information security, 22(5), 1125-1162.

[4] Hajj, S., El Sibai, R., Bou Abdo, J., Demerjian, J., Makhoul, A., & Guyeux, C. (2021). Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets. Transactions on Emerging Telecommunications Technologies, 32(4), e4240.

[5] Díaz-Verdejo, J., Muñoz-Calle, J., Estepa Alonso, A., Estepa Alonso, R., & Madinabeitia, G. (2022). On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. Applied Sciences, 12(2), 852.

[6] Streun, F., Wanner, J., & Perrig, A. (2022). Evaluating susceptibility of VPN implementations to DoS attacks using adversarial testing. In Network and Distributed Systems Security Symposium 2022 (NDSS'22). Internet Society.

[7] Dash, B., Ansari, M. F., Sharma, P., & Ali, A. (2022). Threats and opportunities with AI-based cyber security intrusion detection: a review. International Journal of Software Engineering & Applications (IJSEA), 13(5).

[8] Sowmya, T., & Anita, E. M. (2023). A comprehensive review of AI based intrusion detection system. Measurement: Sensors, 28, 100827.

[9] Habeeb, M. S., & Babu, T. R. (2022). Network intrusion detection system: a survey on artificial intelligence-based techniques. Expert Systems, 39(9), e13066.

[10] Mane, S., & Rao, D. (2021). Explaining network intrusion detection system using explainable AI framework. arXiv preprint arXiv:2103.07110.

[11] Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., & Cheah, M. (2023). Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys, 55(11), 1-40.

[12] Patil, S., Varadarajan, V., Mazhar, S. M., Sahibzada, A., Ahmed, N., Sinha, O., ... & Kotecha, K. (2022). Explainable artificial intelligence for intrusion detection system. Electronics, 11(19), 3079.

[13] Rathee, A., Malik, P., & Parida, M. K. (2023, May). Network Intrusion Detection System using Deep Learning Techniques. In 2023 International Conference on Communication, Circuits, and Systems (IC3S) (pp. 1-6). IEEE.

[14] Vanin, P., Newe, T., Dhirani, L. L., O'Connell, E., O'Shea, D., Lee, B., & Rao, M. (2022). A study of network intrusion detection systems using artificial intelligence/machine learning. Applied Sciences, 12(22), 11752.

[15] Shukla, K. A., Ahamad, S., Rao, G. N., Al-Asadi, A. J., Gupta, A., & Kumbhkar, M. (2021, December). Artificial intelligence assisted IoT data intrusion detection. In 2021 4th International Conference on Computing and Communications Technologies (ICCCT) (pp. 330-335). IEEE.

[16] Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 8(1), 65.

[17] Palimkar, P., Shaw, R. N., & Ghosh, A. (2022). Machine learning technique to prognosis diabetes disease: Random forest classifier approach. In Advanced computing and intelligent technologies: proceedings of ICACIT 2021 (pp. 219-244). Springer Singapore.

[18] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., ... & Ranjan, R. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), 1-33.

[19] Hamilton, R. I., & Papadopoulos, P. N. (2023). Using SHAP values and machine learning to understand trends in the transient stability limit. IEEE Transactions on Power Systems, 39(1), 1384-1397.

[20] Arslan, Y., Lebichot, B., Allix, K., Veiber, L., Lefebvre, C., Boytsov, A., ... & Klein, J. (2022, August). Towards refined classifications driven by shap explanations. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction (pp. 68-81). Cham: Springer International Publishing.