IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 September 2023, accepted 2 October 2023, date of publication 10 October 2023, date of current version 23 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323573

RESEARCH ARTICLE

An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things

MOUSA'B MOHAMMAD SHTAYAT[1], MOHAMMAD KAMRUL HASAN[1], (Senior Member, IEEE), ROSSILAWATI SULAIMAN[1], SHAYLA ISLAM[2], (Senior Member, IEEE), AND ATTA UR REHMAN KHAN[3], (Senior Member, IEEE)

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia
[2]Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur 56000, Malaysia
[3]College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates

Corresponding authors: Mohammad Kamrul Hasan (hasankamrul@ieee.org) and Shayla Islam (shayla@ucsiuniversity.edu.my)

This work was supported by the Ministry of Higher Education Malaysia, under the Fundamental Research under Grant FRGS/1/2020/ICT03/UKM/02/6.

ABSTRACT Ensuring the security of critical Industrial Internet of Things (IIoT) systems is of utmost importance, with a primary focus on identifying cyber-attacks using Intrusion Detection Systems (IDS). Deep learning (DL) techniques are frequently utilized in the anomaly detection components of IDSs. However, these models often generate high false-positive rates, and their decision-making rationale remains opaque, even to experts. Gaining insights into the reasons behind an IDS's decision to block a specific packet can aid cybersecurity professionals in assessing the system's effectiveness and creating more cyber-resilient solutions. In this paper, we offer an explainable ensemble DL-based IDS to improve the transparency and robustness of DL-based IDSs in IIoT networks. The framework incorporates Shapley additive explanations (SHAP) and Local comprehensible-independent Clarifications (LIME) methods to elucidate the decisions made by DL-based IDSs, providing valuable insights to experts responsible for maintaining IIoT network security and developing more cyber-resilient systems. The ToN_IoT dataset was used to evaluate the efficacy of the suggested framework. As a baseline intrusion detection system, the extreme learning machines (ELM) model was implemented and compared with other models. Experiments show the effectiveness of ensemble learning to improve the results.

INDEX TERMS Explanation AI (XAI), intrusion detection systems (IDS), SHapley additive explanations (SHAP), local comprehensible model-independent clarifications (LIME), ensemble learning, CNN.

I. INTRODUCTION

An emerging technology called the Industrial Internet of Things (IIoT) is being connected more and more into our daily lives [1], [2]. This technology is transforming cities around the globe into smart cities, and a recent study conducted in 2021 revealed a significant increase in the quantity of IIoT-connected devices. Specifically, the number of such devices was recorded at 8.6 billion in 2019, which rose to 9.76 billion in 2020 and reached 11.28 billion in 2021. The study also projected a substantial surge in the coming years, estimating a staggering 29.42 billion IoT-connected devices by 2030 [3]. By leveraging communication technologies, IIoT aims to connect and deploy billions of devices in order to support various applications across different industries, such as agriculture [4], healthcare [5], factories [6], and transportation [7]. The IIOT has the potential to enhance productivity and efficiency through smart, remote management. However, it also brings an increased vulnerability to cyberattacks due to constant

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

connectivity, data sharing, and the IIoT networks' resource-constrained design [8], [9]. Thus, it is essential to develop different security mechanisms [10], [11], [12] to address a variety of cyberattacks on the IIOT ranging from different attacks. In this regard, IIoT networks may be protected against a variety of attacks using intrusion detection systems (IDS), which is a potential approach. In order to protect systems and networks from malicious actions that can bypass security boundaries, Intrusion Detection Systems (IDSs) are frequently utilized as a supplementary layer of security [13]. IDSs are designed to monitor system and network events for any suspicious activity and provide an early warning of potential threats [14]. This proactive security approach helps mitigate potential damages and minimize the impact of any successful intrusion attempts. By providing an extra layer of defense, IDSs are critical in ensuring the security and integrity of systems and networks.

In addition to IDSs, alternative strategies and techniques exist for enhancing IIoT systems' security. Ullah et al. introduced a novel approach to encrypting and digitally authenticating data within a network of interconnected devices, obviating the need for traditional certificates. This technique leverages a specialized curve design to expedite and fortify both encryption and authentication processes [15]. Conversely, Shahzad et al. unveiled an innovative mechanism for verifying the legitimacy of devices within a sensor and machine network, bypassing the reliance on passwords or certificates. This methodology employs a confidential key and a randomized value to validate the devices' identities [16].

The current IDSs for IIoT encounter limitations in anomaly detection, with prevailing methods yielding unsatisfactory outcomes. This deficiency often leads to elevated rates of inaccurate results, compromised detection performance, and substantial losses [17], [18], [19], [20], [21], [22]. These challenges underscore the pressing need for enhanced anomaly detection techniques tailored to the intricate IIoT environment.

Intelligent IDS that utilize artificial intelligence (AI) algorithms are gaining popularity as an effective means of detecting IIoT-related intrusions. Researchers have used a variety of machine learning (ML) approaches in recent years to classify network attacks without detailed knowledge of their specific traits. Traditional ML approaches, however, struggle to offer unique feature descriptors for attack detection due to their limited model complexity. Lately, a significant advancement in ML has been achieved through the simulation of the human brain using neural network structures, known as DL methods. These methods employ a deep-layered architecture to tackle complex problems, including convolutional neural networks (CNNs) [23]. By learning the distinctive characteristics of each form of IIoT attack, DL-based IDSs can rapidly and effectively identify and anticipate system intrusions. DL-based IDSs offer a more effective and dependable method of protecting IIoT networks by maximizing the detection rate of IIoT-related intrusions [24]. DL models are often considered black-box models, with decisions made by these models provided to users without any explanations or interpretations on how or why such decisions were made [25]. This lack of transparency and interpretability means that users are unable to understand or trust the decisions made by DL models [26] and are unable to base their choices on the results of DL models. To address these limitations, the emerging paradigm of XAI provides a range of techniques to interpret and understand predictions made by DL models [27], [28]. By using XAI, cybersecurity experts can explain the decisions made by DL models and make them more interpretable [24]. As a result, professionals are better able to trust and modify these models and, eventually, make more informed judgments based on the models' outputs [29]. This research paper introduces a new framework that combines DL architecture with XAI techniques for binary and multi-class classification. Consequently, the following are the study's primary contributions:

- □ Propose an innovative ensemble DL-based architecture that uses three CNN models and an extreme learning machines (ELM) model to secure IIoT networks.
- □ Improve the intrusion detection performance for the IIoT network by evaluating the proposed model using one of the recent datasets, the TON_IoT dataset [30], and overcoming the state-of-the-art IDS's performance in the same field in binary and multi-class classification.
- □ Improve the interpretability of DL-based decisions by providing explanations. The project utilizes two specific techniques, namely Local Interpretable Model-agnostic Explanations (LIME) [31] and SHapley Additive explanations (SHAP) [32], in order to achieve this goal. By effectively explaining the decision-making process of the IDS, this project aims to enhance the transparency and reliability of the cybersecurity model and ultimately help prevent cyber threats.

The remainder of this paper is divided as follows. Section II introduces an overview of the concepts and techniques used in our research, including CNNs, ELM, ensemble methods, and XAI. In Section III, we present a literature review of related work in the field of ID, focusing on the use of these techniques. Section III describes the methodology used in our study, including details on the dataset, data preparation techniques, and implementation of the five models (3 CNN models, ELM, and Ensemble). Section IV presents and analyzes the results obtained from the five models, including accuracy, precision, recall, F1 score, and the application of XAI methods (SHAP and LIME). In Section V, we provide a summary of our key findings and discuss potential future work that could build on our research. We conclude our paper in Section VI."

II. BACKGROUND

This section provides an overview of the key concepts and techniques employed in our research. Our study utilizes CNNs, extreme learning machines (ELM), ensemble methods, and XAI techniques to tackle the challenges at hand.

A. CONVOLUTION NEURAL NETWORK

An advanced kind of neural network architecture known as a CNN is created especially for processing grid-like data, such as images and time-series data. Convolution, pooling, and fully connected layers make up the three basic layers that make up the core architecture of a CNN [33]. By employing convolutional layers, CNNs can automatically learn to extract relevant features from input data, reducing the need for manual feature engineering [34]. This ability to learn hierarchical feature representations has made CNNs popular for various classification and pattern recognition tasks.

B. EXTREME LEARNING MACHINE (ELM)

ELM is a single-hidden-layer feedforward neural network (SLFN) training algorithm that aims to overcome some of the limitations of traditional gradient-based learning methods, such as slow convergence and getting stuck in local minima [35]. ELM offers a fast learning speed and good generalization performance by randomly assigning input weights and analytically determining the output weights.

C. ENSEMBLE LEARNING

Ensemble learning, a prominent research area in the data mining field, addresses the challenges of complex data types by unifying data fusion, modeling, and mining within a single framework [36], [37], [38]. Ensemble methods are considered the state-of-the-art solution for many ML challenges, as they enhance the predictive performance of a single model by training multiple models and combining their predictions [39], [40]. Initially, ensemble learning extracts diverse features through various transformations and then employs multiple learning algorithms to generate weak predictive outcomes based on these features [41]. Ultimately, the informative knowledge from these preliminary results is fused to attain knowledge discovery and improved predictive performance using adaptive voting schemes [36]. Ensemble methods, which include bagging, boosting, and stacking, are beneficial for reducing overfitting and increasing model robustness [42].

D. EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

XAI is a field of study that focuses on developing algorithms and techniques that enable AI models to provide human-understandable explanations for their decisions and predictions [39]. XAI aims to enhance transparency, accountability, and trust in AI systems by allowing humans to understand how the models arrived at their conclusions [29]. In the context of Cybersecurity, XAI is crucial in detecting and preventing cyber threats [43]. Cybersecurity experts use AI models to analyze large volumes of data and identify potential threats. However, the black-box nature of these models makes it difficult to understand how they make decisions, leaving experts unsure of how to respond to identified threats [44]. With XAI, cybersecurity experts can better understand the reasoning behind the AI model's predictions, enabling them to make more informed decisions about how to respond to cyber threats [29].

III. RELATED WORK

Many researchers have been interested in using ML and DL methods for network security and attack detection. Different algorithms were used: XGBoost [45], Decision Tree [43], random forest (RF) [46], and support vector machine (SVM) [47]. [48] used three benchmark datasets, NSL-KDD, CIC-IDS2018, and TON IoT, to offer a three-tiered DL-based technique for identifying abnormal network intrusion behaviors. The proposed framework combines K-means clustering, GANomaly, and CNN techniques. An Ensemble-based Network IDS with Bayesian CNN was implemented and evaluated using the NSL-KDD and UNSW-NB15 datasets [49]. A hybrid CNN-based DL approach has been proposed for classifying flow traffic as an attack or not [34].

A novel model by [50] evaluated two feature sets (NetFlow and CICFlowMeter) across different network environments and attacks. NetFlow demonstrates superior detection accuracy, enhanced by Shapley Additive explanations (SHAP) for explaining ML model decisions and feature influence. An explainable deep learning-based intrusion detection framework was presented by [51] for improving transparency and resiliency in IoT networks. The framework utilizes Shapley additive explanations (SHAP) to interpret decisions made by the deep learning-based IDS. Experimental results using the ToN_IoT dataset demonstrate high performance with 99.15% accuracy and 98.83% F1 score, showcasing its effectiveness in protecting IoV networks against sophisticated cyber-attacks.

A real-time deep neural network-based intrusion detection system was proposed using benchmark Netflow-based datasets. It incorporates a packet capturing and detecting algorithm for accurate attack detection, showcasing its effectiveness [52]. Another recent model implemented by [53] achieved good performance for the Internet of Medical Things (IoMT).

The present study focuses on the interpretable aspect of classification algorithms employed in IDS by utilizing an ensemble model based on three CNN models. Ensemble models are well-known for generating accurate results by choosing the best outcome. However, our research prioritizes accuracy and emphasizes the importance of interpretability in IDS models. This approach allows for a better understanding of the model's reasoning and helps to identify any potential issues or errors in the DL model. By combining accuracy and interpretability in our approach, we aim to develop an IDS model that is effective in detecting intrusions and transparent and easily understandable for security professionals.

IV. METHODOLOGY

The methodology employed in this study aimed to develop a robust IDS using the ToN-IoT dataset since it is considered an IIoT dataset, one of the recent IIoT datasets, and has nine types of attacks. Most state-of-the-art- IDSs are evaluated

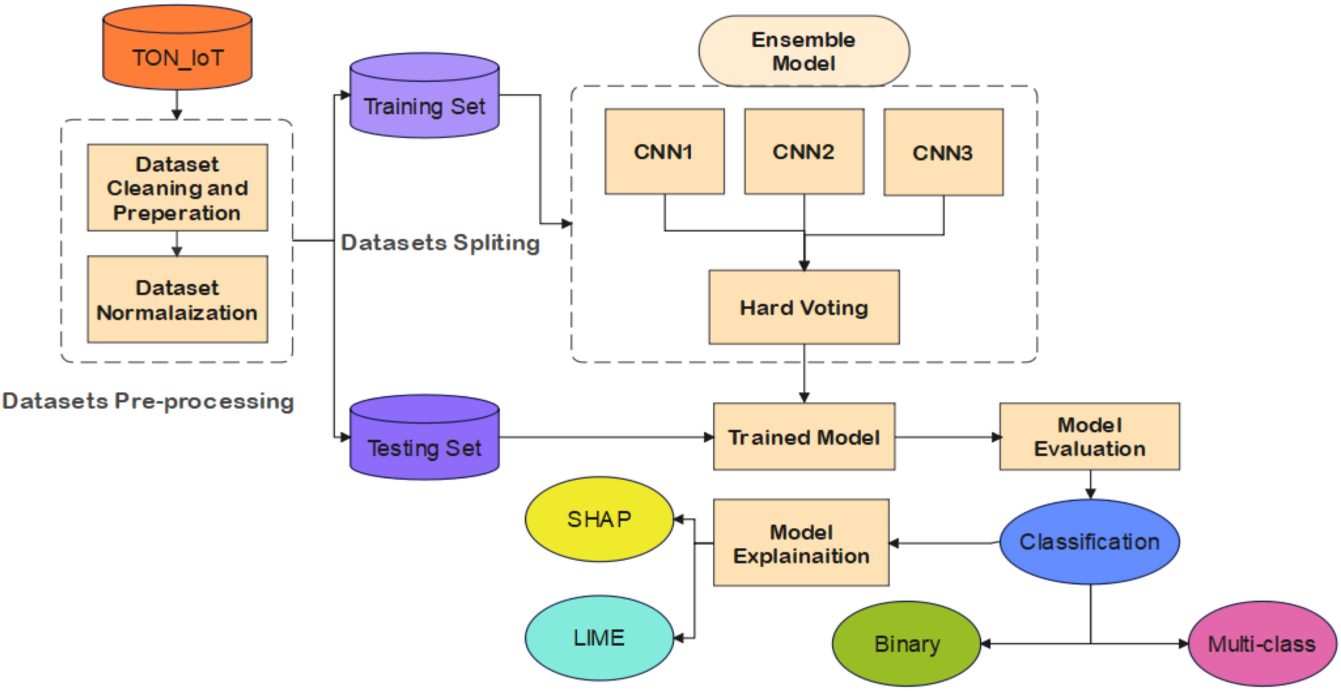

FIGURE 1. Proposed IIoT intrusion detection model architecture.

using non-recent datasets, which are limited in their ability to address the latest attack categories. Alternative intrusion detection systems (IDSs) were assessed using recent datasets encompassing only a restricted range of attack types or classifications. For instance, LATAM-DDOD-IOT focuses solely on two categories: Denial of Service (DoS) and Distributed Denial of Service (DDoS). On the other hand, CIC IoT 2023 incorporates seven distinct attack categories. Still, most instances within this dataset pertain to DDoS and DoS attacks, leaving a comparatively lower number of instances for other attack categories. Notably, the authors of this evaluation did not clarify whether this dataset can be categorized as IIoT data. The data preparation stage involved normalizing and scaling the data using min-max scaling to ensure all features were in the same range. Five distinct models were employed in the creation of the IDS. Three different CNN models were utilized, with varying numbers of layers and nodes, to evaluate their performance. An ensemble model was also developed, which combined the three CNN models using a hard voting mechanism. This approach allowed for the selection of the most accurate prediction among the three CNN models. Several metrics were computed to assess how well the models performed. Additionally, XAI methods such as SHAP and LIME provided insights into the feature importance and the decision-making process of the models. Model architecture is shown in Figure 1.

A. DATASET PRE-PROCESSING

In our study, we leverage the TON-IoT dataset, which consists of diverse data sources collected from an entire IIoT (Industrial Internet of Things) system. The dataset includes telemetry information from connected devices, operating system logs from both Linux and Windows systems, and network traffic related to the IIoT system. This mixed data was collected from a medium-sized IoT network developed by the UNSW Canberra IoT Labs and the Cyber Range. The ToN-IoT dataset is available through the ToN-IoT repository [30] and is presented in CSV format with labeled columns indicating either normal behavior or an attack, along with a sub-category specifying the type of attack.

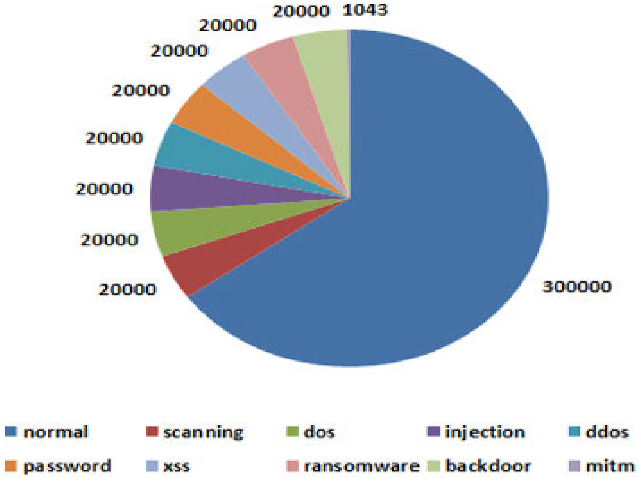

FIGURE 2. Type distribution in TON_IoT attacks.

The TON-IoT network dataset identifies nine primary categories of attacks targeting IoT networks, as explained in Figure 2. These are as follows:

- □ Scanning: Attackers scan IoT systems to gather information, such as available services and open ports, to identify potential vulnerabilities and plan further attacks.
- □ DoS (Denial of Service): This attack aims to overwhelm an IoT system with malicious requests, rendering its services unavailable.
- □ Injection: In this attack, the perpetrator attempts to insert malicious data or software into an IoT system, potentially disrupting its normal operation and control mechanisms.
- □ DDoS (Distributed Denial of Service): Similar to DoS, but executed using a network of compromised systems (botnets) to flood IoT resources with multiple connections, exhausting the resources.
- □ Password (Password Cracking Attack): Attackers use various password-cracking techniques (e.g., dictionary attacks or brute force) to bypass authentication methods and gain control over IoT devices.
- □ XSS (Cross-Site Scripting): IoT applications' web servers can be vulnerable to malicious software like XSS, which can compromise the authentication processes and information used by IoT systems.
- □ Ransomware: This sophisticated malware denies legitimate users access to devices or services, demanding payment for the decryption key needed to regain access. IoT applications and devices are attractive targets due to their critical functions.
- □ Backdoor: In this type of attack, backdoor malware allows an attacker to gain unauthorized remote access to compromised IIoT systems. The attacker can then use these systems and botnets to launch DDoS attacks.
- □ MITM (Man-In-The-Middle): This common network attack involves intercepting data flow within an IoT network, allowing the attacker to steal sensitive information or manipulate data transmission.

Data cleaning and preparation are crucial to ensure high accuracy and expedite the learning process when employing ML methods. This process typically involves removing irrelevant features that could negatively impact performance, converting non-numerical features, and addressing missing values. Data preparation primarily consists of two stages: data pre-processing and data normalization.

Data pre-processing: Initially, the' label' and' type' features are dropped from the dataset, as they will serve as target variables in the approaches under consideration. Categorical features with nominal values are converted to numeric values to facilitate the application of ML methods. Label encoding is applied to any remaining categorical features in the dataset to maintain compatibility with the chosen ML algorithms.

Network Data

Data Normalization: In some cases, certain features may have significantly larger values than others, potentially leading to biased model outcomes. Data normalization helps to mitigate this issue by scaling features within a range of [0, 1] without altering the overall data behavior. The min-max normalization method is employed to scale feature values within the [0, 1] range using the formula (1). This approach ensures that features with smaller values are not overshadowed by those with larger values, promoting a more balanced analysis. After that, the dataset was divided into a training set (80%) and a testing set (20%).

$$z = (z - MIN)\big/(MAX - MIN) \tag{1}$$

where z represents the value of an attribute, MIN refers to the minimum value observed for that attribute, and MAX represents the maximum value observed.

B. MODEL IMPLEMENTATION

We implemented five different models: three CNNs, an Extreme Learning Machine (ELM), and an ensemble model. For the ensemble model, we will combine the predictions from the three CNN models to improve the performance.

- □ Convolution Neural Networks: For binary and multi-class classification tasks, we developed and tested three distinct CNN models in our research.
- □ The first model (Figure 3), referred to as CNN 1, is composed of a single 1D convolutional layer featuring 32 filters, a kernel size of 2, and the same padding applied to the input data. A ReLU activation function, a max pooling layer with a pool size of 2, reduces the input data's spatial dimensions. The network then employs a flattened layer, two fully connected layers with 50 and 25 neurons (both with ReLU activation), and an output layer with either 2 or 10 neurons, depending on the classification task. A softmax activation function is used for binary or multi-class classification.
- □ The second model (Figure 4), CNN 2, begins with a 1D convolutional input layer containing 64 filters, a kernel size of 3, and ReLU activation applied to input data. A 1D max pooling layer with a pool size of 2 follows. The model then incorporates another 1D convolutional layer with 128 filters, a kernel size of 3, and ReLU activation, followed by another 1D max pooling layer with a pool size of 2. After flattening the input data, a fully connected layer with 64 neurons and ReLU activation is utilized. The output layer consists of 2 or 10 neurons and a softmax activation function for binary or multi-class classification.
- □ Our third model (Figure 5), CNN 3, starts with a 1D convolutional input layer featuring 32 filters, a kernel size of 3, and ReLU activation applied to input data. A 1D max pooling layer with a pool size of 2 is incorporated, followed by another 1D convolutional layer with 64 filters, a kernel size of 3, and ReLU activation. Another 1D max pooling layer with a pool size of 2 is employed, succeeded by a third 1D convolutional layer with 128 filters, a kernel size of 3,

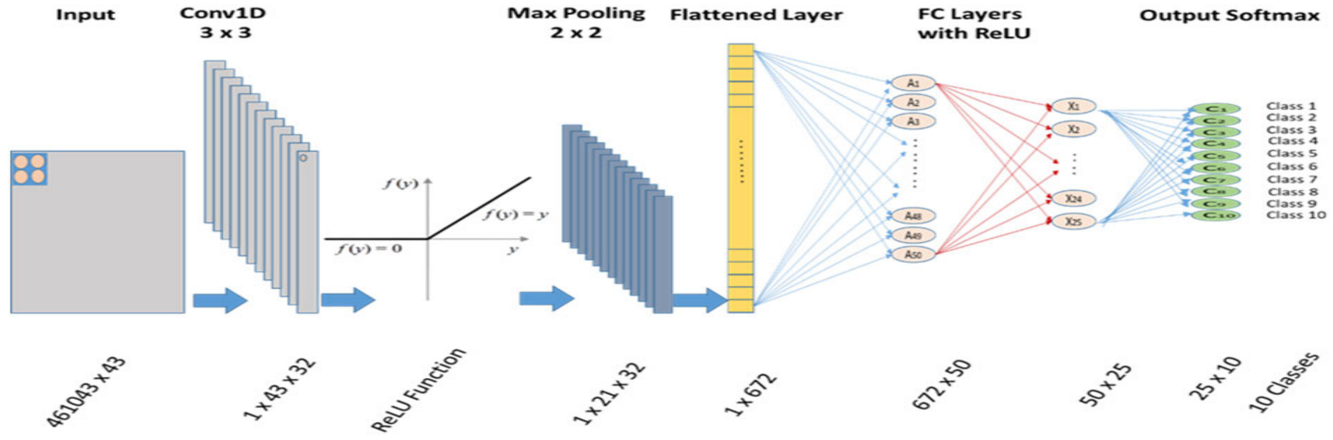

FIGURE 3. CNN1 model architecture.

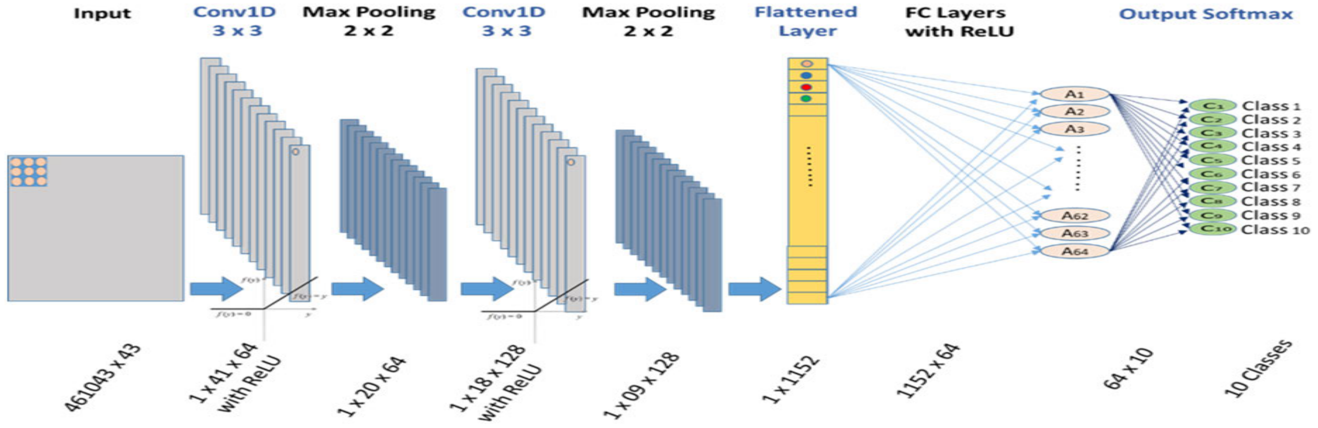

FIGURE 4. CNN2 model architecture.

and ReLU activation. A final 1D max pooling layer with a pool size of 2 precedes the flattening layer. The network features a fully connected layer with 128 neurons and ReLU activation, a dropout layer with a rate of 0.5 to prevent overfitting during training, and an output layer with either 2 or 10 neurons and a softmax activation function for binary or multi-class classification.

Each of the three CNN models mentioned follows a similar architecture, including an input layer, convolutional layers, pooling layers, flattened layers, fully connected layers, and an output layer. However, the specific structure of each model differs, with variations in the number of convolutional layers, pooling layers, flattened layers, and dense layers. Please refer to Figures 3, 4, and 5 for a visual representation of these model structures. If we consider the input feature map of the CNN as M_j, the convolution process can be represented as follows:

$$M_j = ReLU(conv1D\left(M_{j-1}, W_j\right) + b_j) \tag{2}$$

where W_j is the convolution kernel weight vector of the j layer, $ReLU$is the activation function, and b_j is the bias of the j layer.

The convolutional layer captures various features from the previous data M_{j-1} using different window values and convolution kernels. By applying the same convolution kernel, the weights and bias are shared. This significantly reduces the number of parameters in the overall neural network. The pooling layer performs sampling on the feature map obtained from the convolutional layer based on various sampling rules. If M_j represents the input to the pooling layer and M_{j+1} represents the output of the pooling layer. All models use the maximum pooling technique with varying pool sizes. The maximum pooling operation can be described as follows:

$$M_{j+1} = \max\left(M_j, poolsize\right) \tag{3}$$

Next is the flattening layer: The flattening layer in the network plays a crucial role in converting the output of convolutional and pooling layers into a single, elongated feature vector. The input data is transformed by collapsing its spatial dimensions into the channel dimension to achieve this flattening process. Specifically, if the input consists of M feature maps, each with a dimension of $Fin \times Fin$, the flattened output, $Fout$, is obtained by multiplying the input dimensions by the number of maps as described in the

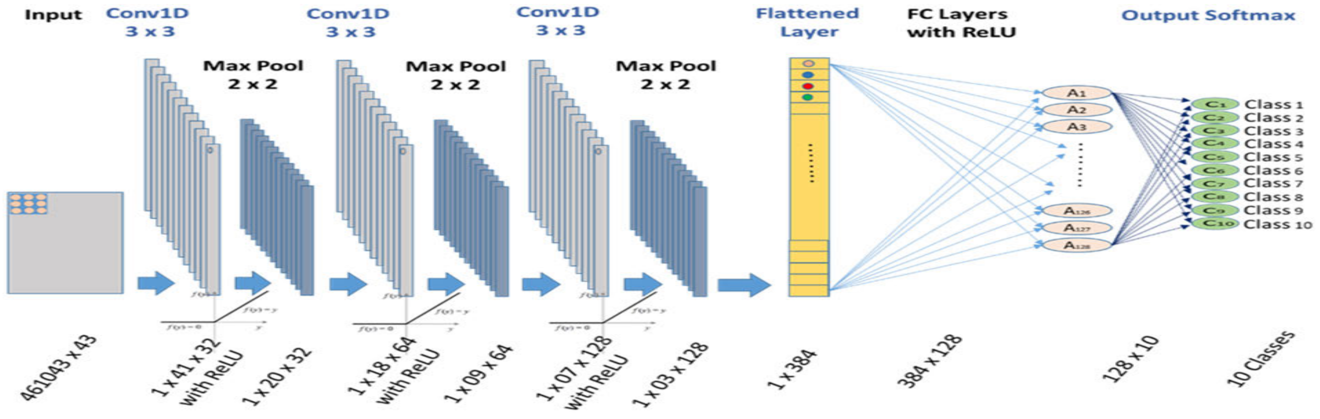

FIGURE 5. CNN3 model architecture.

following formula:

$$F_{out} = M * F_{in} * F_{in} \tag{4}$$

Fully connected (FC) layers with the ReLU function are layers in which all the inputs from one layer are connected to every activation unit in the next layer. Their main role is to aggregate the high-level features extracted by the preceding layers (such as convolutional and pooling layers) into a condensed representation of low-level features. The classifier in the network's output layer utilizes this compressed representation of features to generate class probabilities and carry out classification tasks.

The output layer of the neural network employs the softmax function to predict the correct classification for each sample record in the IoT attacks dataset. This module performs two types of classification: binary classification, which determines whether the sample is normal or anomaly, and multi-class classification, which assigns one of several attack types (Backdoor, DoS, DDoS, Injection, MITM, Scanning, Ransomware, Password, XSS) to the sample. The softmax function ensures that the predicted probabilities for each class sum up to 1, allowing for a meaningful classification decision to be made.

The final layer of the neural network is a softmax layer, having the same number of nodes as the output layer. Its purpose is to normalize the output and convert it into a probability distribution across the classes. Softmax assigns numerical probability values to each class at the output layer, ensuring that these probabilities add up to 1.0, thus conforming to a valid probability distribution. This allows us to interpret the output as the likelihood of the input belonging to each class. The softmax function σ is defined as follows:

$$\sigma(x)_i = \frac{e^{xi}}{\sum_{K=1}^{K} e^{xk}} for\ i = 1, 2, 3, \ldots, K \tag{5}$$

The models are compiled using the Adam optimizer and sparse categorical cross-entropy as the loss function, with accuracy as the evaluation metric. By testing these three models, we aimed to explore the trade-offs between model complexity and performance in our classification tasks.

2) Ensemble model: In addition to the individual CNN models, we also implement an ensemble model that combines the predictions of the three CNNs. The ensemble model is constructed using the Voting-Classifier method from the sci-kit-learn library, which aggregates the outcomes of the three CNN models through hard voting. The ensemble model combines the strengths of the individual models, potentially leading to improved classification performance compared to each CNN model separately.

Hard voting is an approach used in classification problems, where an ensemble of classifiers is created, and each classifier predicts a given input. The final prediction is determined by a majority vote of the individual classifiers' predictions. In other words, the class that receives the most votes from the classifiers becomes the final predicted class. Equation 6 explains the hard voting technique.

$$y = modeC1\{(x)\,, C2\,(x)\,, C3\,(x)\} \tag{6}$$

where y is the final predicted class, C1 is the output class of CNN1, C2 is the output class of CNN2, and C3 is the output class of CNN3.

3) Extreme Learning Machine: We also implement an ELM model for binary and multi-class classification in our study. For Binary classification, the ELM model is initialized with an input layer corresponding to the number of input features and an output layer with a single output neuron. Following this, we add two hidden layers to the ELM model, each with 1000 neurons: The first layer consists of 1000 neurons with a sigmoid activation function, and the second layer contains 1000 neurons with a hyperbolic tangent (tanh) activation function. For multi-class classification, the ELM model is initialized with an input layer corresponding to the number of input features, an output layer with a single output neuron, and a batch size of 100 for training. The classification parameter is set to" multi-class" to handle multiple attack types.

C. MODEL EXPLAINABLE

Two techniques, Shapley Additive explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), were implemented to enhance transparency and comprehensibility in decision-making. SHAP assesses the influence of individual features on the model's predictions, providing insights at a feature level. Conversely, LIME constructs surrogate models that approximate the behavior of the underlying model in the vicinity of a specific instance, aiming to explain individual predictions.

1) SHARPLY ADDITIVE EXPLANATION (SHARP)

SHAP is a well-established, comprehensive framework for interpreting various models. It explains the predictions for a specific instance by calculating each feature's impact on the final decision, whether positive or negative. Unlike linear models, SHAP is applicable to any model or classifier. Instead of concentrating just on local interpretations, SHAP considers global interpretations by averaging each feature independently and adding up the input values of the features. The explanation for an instance x using SHAP is derived as follows:

$$g(s) = v_0 + \sum\nolimits_{i=1}^{N} v_i s_i \tag{7}$$

where g is the explanation model, N is the maximum size of the feature vector, v_i is the Shapley value for the feature I, s is the simplified features (coalition factor), and $s \epsilon \{0,1\}^{N}$, the 1 in s means the features in the new data are the same as those of the original data, while the 0 means the features in the new data are different from those of the original data.

The Shapley value represents the contribution of each feature to the model's prediction, with higher values indicating a larger contribution. To identify the most important features, we can use the importance factor equation:

$$IF_j = \sum\nolimits_{i=1}^{n} |v_j(xi)| \tag{8}$$

where n is the total number of instances in the dataset, and IFj represents the average absolute value of the Shapley value for feature j.

By computing IFj for all features, we can identify the most influential features in the model's predictions.

D. LOCAL COMPREHENSIBLE MODEL INDEPENDENT CLARIFICATION (LIME)

LIME aims to create an understandable model utilizing an easily understood representation while maintaining local fidelity to the original classifier. Given an instance with its original representation, $x \in \mathbb{R}^d$, and an explanation model, $g\epsilon G$, where G represents a set of visually expressible, interpretable models (e.g., a linear model), LIME's explanation can be determined as follows:

$$\varphi(x) \underset{g\in G}{\text{argmin}} = [L(f, g, \omega x) + \Omega(g)] \tag{9}$$

where $\varphi(x)$ The interpretation model, f is the classification model, ωx is a similarity metric between the original and new instances (with higher values signifying greater similarity), L is the loss function that gauges the closeness of the predictions between the explanation and original models, and Ω(g) measures the complexity of model g.

LIME aims to develop a model that is both locally focused and interpretable by minimizing the function L(f, g, ωx)+Ω(g), where f is the original model, g is the locally derived interpretation model, and ωx is a weight vector for instance x. The regularization term Ω(g) helps prevent overfitting the interpretation model.

After minimizing the objective function, LIME explains a specific instance using the locally derived interpretation model $\varphi(x)$. The interpretation model $\varphi(x)$ is designed to be simple and transparent, making it easier for humans to understand the reasoning behind a particular prediction. By using a locally focused and interpretable model, LIME provides insights into the decision-making process of complex models.

V. MODEL TRAINING AND EVALUATION

Section IV-A mentions that the TON-IoT dataset is stored in a CSV file format. Initially, we split the dataset into two parts: training and testing. 80% of the data is allocated for training and evaluating the chosen ML methods, while the remaining 20% is reserved for testing the models with unseen data. The 80-20 split, as recommended in [54], is considered the optimal ratio to prevent overfitting, where a model memorizes the data instead of learning from it. The suggested framework examines the use of explanation approaches to determine the most instructive elements and look at how they affect the predictions made by the final model.

To evaluate the effectiveness of our CNN-based IDS models trained on the ToN_IoT dataset, we employ four widely recognized metrics: Accuracy, Precision, Recall, and F1-score. To compare the outcomes of our IDS model with the Ground Truth, we employ terms such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These metrics and terminologies allow us to measure the model's performance and evaluate its ability to classify instances and distinguish between different classes correctly.

□ Accuracy (ACC) is a metric utilized to assess the overall performance of a model. It is determined by calculating the ratio of correct predictions to the total number of predictions made. In the binary classification IDS model context, ACC is computed based on the counts of true positives, false positives, true negatives, and false negatives. This metric indicates how accurately the model can classify instances and make correct predictions.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

□ Precision (P) is a metric used to evaluate the performance of our IDS system. Specifically, it measures the ratio of correctly predicted attack instances to the number of instances classified as attacks. This can be

TABLE 1. Binary classification results.

Model	Accuracy	Precision	Recall	F-Score	Error	Sensitivity	Specificity
CNN-1	99.5 %	99.5 %	100 %	99.7 %	0.0044	99.57 %	99.57 %
CNN-2	99.64 %	**100 %**	99.62 %	**100 %**	0.0033	99.62 %	99.62 %
CNN-3	99.46 %	99.52 %	99.41 %	99.52 %	0.0052	99.41 %	99.41 %
Ensemble	**99.69 %**	**100 %**	**100 %**	**100 %**	**0.0030**	**99.68 %**	**99.63 %**
ELM	98.77 %	98.5 %	98.5 %	98.5 %	0.012	98.97 %	98.97 %
Resilient IDS [51]	99.15 %	99.1 %	99.15 %	98.83 %			
DIDS [52]	99.48 %	99.48 %	99.48 %	99.48 %			
IoMT ensemble and Fog cloud [57]	96.35%	90.54%	99.98%	95.03%	0.365		
IDS-Netflow dataset [58]	97.86%	NULL	97.86%	99%	NULL		
Ensemble Framework [59]	98.63%	98.2%	98.6%	98.61%	NULL		
Feature analysis-IIoT [60]	97.49%	NULL	97.51%	99%	NULL		
IDS –Cost-Sensitive, Fog [61]	NULL	98.1%	97.5%	97.7%	0.0216		
(DFF)[50]	94.74 %	NULL	NULL	NULL	NULL		

mathematically represented as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

□ Recall (R): also known as sensitivity or true positive rate, is a metric used to evaluate the performance of our IDS system. It measures the ratio of correctly predicted attack instances to the total number of actual attack instances. Mathematically, it can be expressed as follows:

$$R = \frac{TP}{TP + FN} \tag{12}$$

□ The F1 score is a metric that combines both precision and recall to provide an overall measure of the model's performance. It is calculated as the harmonic mean of precision and recall. Mathematically, it can be expressed as follows:

$$F1score = 2 * \frac{P * R}{P + R} \tag{13}$$

In our experiments, we used Google Colab, a collaborative Jupyter notebook environment, to conduct deep learning research. We employed Keras (version 2.12.0), an open-source Python deep learning library that runs on top of Google's open-source data flow software [55], and TensorFlow [56] (version 2.12.0), as a backend engine. The available RAM for our experiments in the Google Colab environment was 13.62 GB. This setup provided a robust and accessible platform for implementing and testing our models. It comprises the two methods, LIME and SHAP.

A. BINARY CLASSIFICATION RESULTS

Our study focused on developing and testing ML models on the TON-IoT dataset to classify different types of network traffic, including normal and malicious traffic. The' label' feature was used as the target variable during training and evaluation. To quantitatively evaluate the performance of the candidate ML methods, we measured various metrics such as accuracy, error rate, recall, precision, and F-score. Additionally, sensitivity and specificity were calculated for each model. Table 1 and Fig 6, 7, and 8 show that all models achieved high accuracy rates, with Ensemble having the highest accuracy rate of 99.69% and ELM having the lowest accuracy rate of 98.77%. The CNN models achieved higher accuracy rates than the ELM model. CNN 2 and Ensemble achieved 100% precision, while CNN 1, CNN 3, and ELM had precision rates of 99.5%, 99.52%, and 98.5%, respectively. CNN 1 and Ensemble achieved 100% recall rates. CNN 2, CNN 3, and ELM have a recall rate of 99.62%, 99.41%, and 99%, respectively. CNN 2 and ensemble models have the highest F-S rate of 100%, and ELM has the lowest F-S rate of 98.5%. All models achieved high sensitivity and specificity rates, with Ensemble having the highest rates of 100% and 99.68%, respectively.

The TON-IoT dataset was analyzed using the SHAP method for the CNN 2 binary classification task. The results are displayed in Fig. 9. Local explanations of the CNN 2 DL-based IDS using LIME on the TON-IoT dataset are presented in Fig. 9. LIME was applied to provide insights into how the model makes predictions for individual instances within the dataset.

B. MULTI-CLASS CLASSIFICATION RESULTS

Next, we train and evaluate our models using the' type' feature as the target variable. This approach will enable us to better understand the specific aspects of IoT network traffic associated with each category.

When comparing the results, we can see that all models achieved high accuracy rates, with Ensemble having the highest accuracy rate of 99.63% and ELM having the lowest

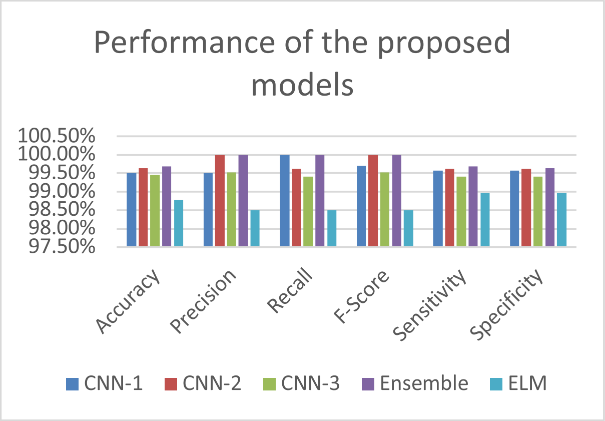

FIGURE 6. Performance comparison between the proposed models for binary classification.

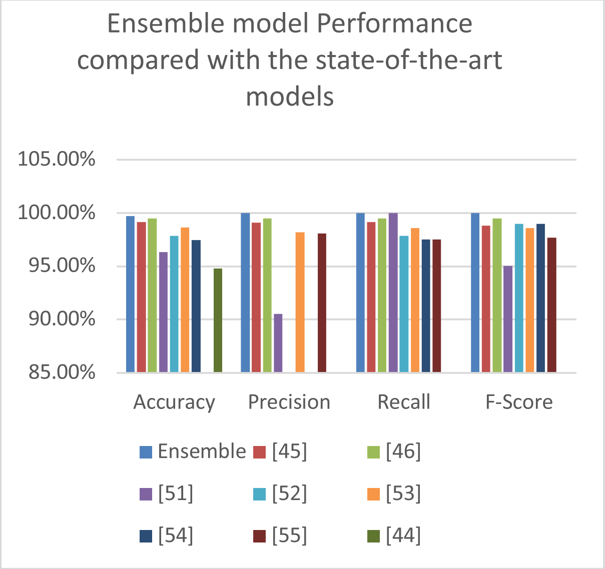

FIGURE 7. Performance comparison between the ensemble model and the current models for Binary classification.

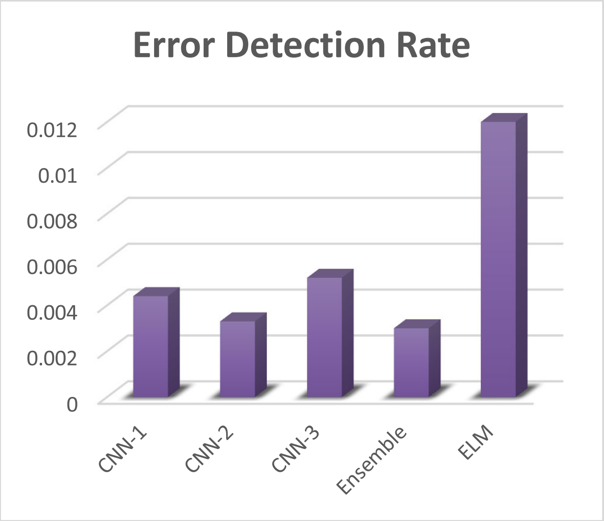

FIGURE 8. Error detection rate for the proposed models for binary classification.

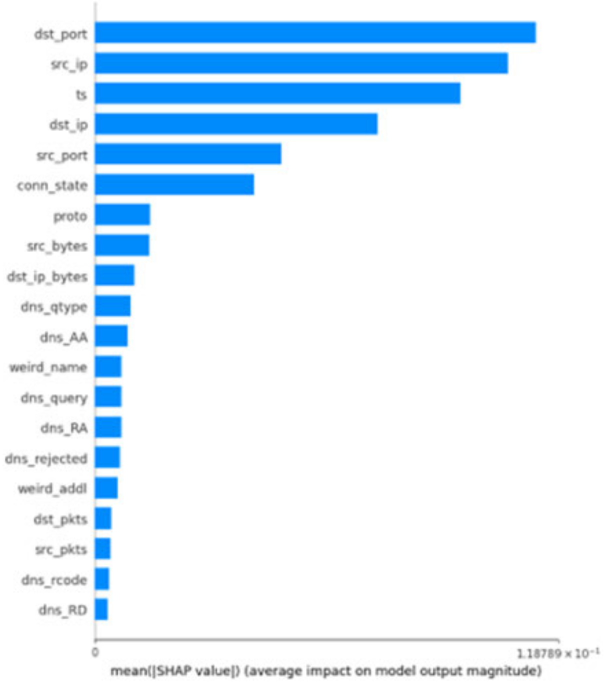

FIGURE 9. Feature importance scores using SHAP techniques for binary classification.

accuracy rate of 88.23%. The ensemble has the highest precision rate of 99.8%, and ELM has the lowest precision rate of 74.7%. CNN 1 achieved the highest recall rate of 99%, followed by CNN 3 and Ensemble, with recall rates of 99% and 99.2%, respectively. CNN 2 and ELM had lower recall rates of 98.7% and 78.8%, respectively. We can also see that all models, except for ELM, achieved high F-score rates, with Ensemble having the highest F-score rate of 99.5%. ELM had a lower F-score rate of 71.7%. In addition, all models achieved high sensitivity and specificity rates, with Ensemble having the highest sensitivity rate of 99.18% and the highest specificity rate of 99.92%. All results are explained in Table 2 and illustrated in Fig. 11, 12, and 13.

Using the SHAP method, the TON-IoT dataset was analyzed for multi-class classification of the CNN 2 DL-based IDS, and the results are shown in Fig. 14. Local explanations of the CNN 2 DL-based IDS using LIME on the TON-IoT dataset are presented in Fig. 15. LIME was applied to provide insights into how the model makes predictions for individual instances within the dataset.

VI. DISCUSSION

Performance of CNN Models: The CNN models (CNN-1, CNN-2, and CNN-3) performed remarkably well across multiple performance metrics, including accuracy, precision, recall, F-Score, error rate, sensitivity, and specificity. These models consistently achieved accuracy above 99% and

TABLE 2. Multi-class classification results.

Model	Accuracy	Precision	Recall	F-Score	Error	Sensitivity	Specificity
CNN-1	99.19 %	**99.98 %**	99 %	99 %	0.008	99.41 %	99.87 %
CNN-2	99.42 %	99.2 %	98.7 %	98.9 %	0.0057	98.7 %	99.9 %
CNN-3	99.5 %	99.5 %	99 %	99.1 %	0.0049	98.91 %	99.89 %
Ensemble	**99.63 %**	**99.8 %**	**99.2 %**	**99.5 %**	**0.0036**	**99.18%**	**99.92 %**
ELM	88.23 %	74.7%	83.82 %	71.7 %	0.11	78.72 %	98.65 %
Resilient IDS [51]	90.15 %	90.55 %	90.22 %	90.75%	NULL		
DIDS [52]	69.53 %	69.53 %	61.96 %	56.84 %	NULL		
IDS-Netflow dataset [58]	NULL	NULL	84.61%	87%	NULL		
IDS –Cost-Sensitive, Fog [61]	NULL	96.1%	97.3%	96.6%	0.033		
An Ensemble RNN [62]	NULL	94%	96.4%	94.9%	NULL		
Micro-service Oriented IDS [63]	NULL	95.2%	96.8%	95.7%	NULL		
XSRU-IoMT [53]	99.38%	99.39%	98.99%	98.83%	NULL		

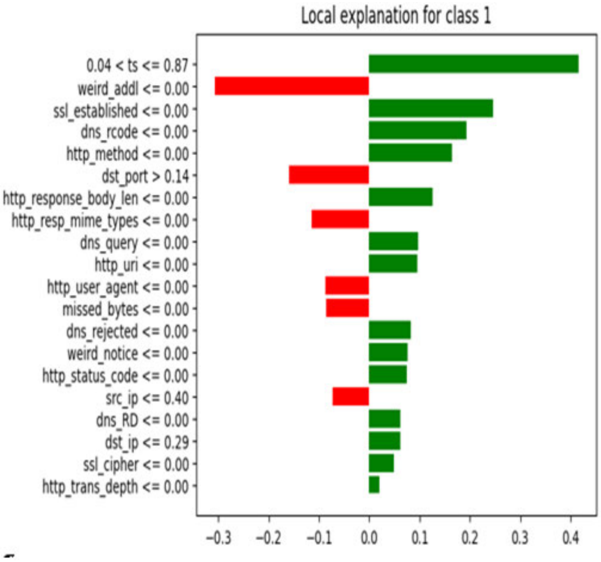

FIGURE 10. Feature importance scores using LIME technique for binary classification.

demonstrated balanced precision and recall scores, indicating their ability to classify positive and negative instances correctly. The low error rates suggest that these models made a few incorrect predictions, further highlighting their effectiveness.

Ensemble Model: The Ensemble model achieved the highest accuracy among the listed models, reaching an impressive 99.69% for binary classification and 99.63% for Multi-class classification. This indicates that combining the predictions of multiple models resulted in improved performance. The model exhibited perfect precision, recall, and F-Score, showcasing its ability to classify instances accurately. The sensitivity and specificity scores were also high, suggesting effective identification of positive and negative instances.

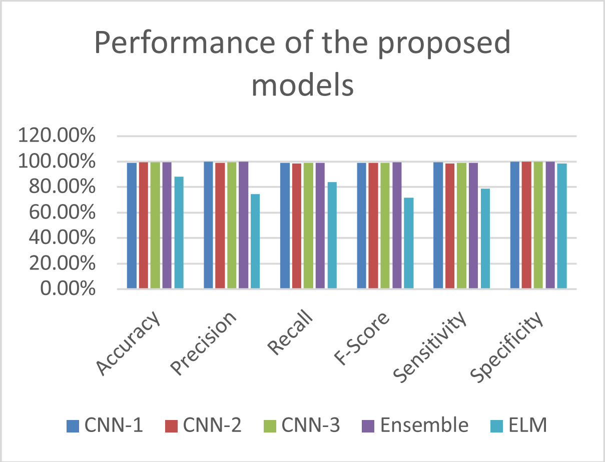

FIGURE 11. Feature importance scores using SHAP techniques for multi-class classification.

In binary classification, The (DFF) model achieved an accuracy of 94.74%, which is comparatively lower than the other models. However, limited information is available regarding its performance, as additional performance metrics are not provided. However, in multi-class classicization, the DIDS model achieved a lower accuracy and demonstrated lower precision, recall, and F-Score values. This suggests a relatively weaker classification performance compared to the other models.

Based on the insights derived from Fig. 9 for the binary classification, the SHAP results emphasize the significance of certain features with notable scores. These include the destination port number (dst port), which plays a crucial role in identifying specific applications or services operating on a device, and the source IP address (src ip), which aids in locating the device and detecting any suspicious or malicious network activity. Furthermore, the timestamp (ts) associated with the network packet emerged as a key feature, enabling

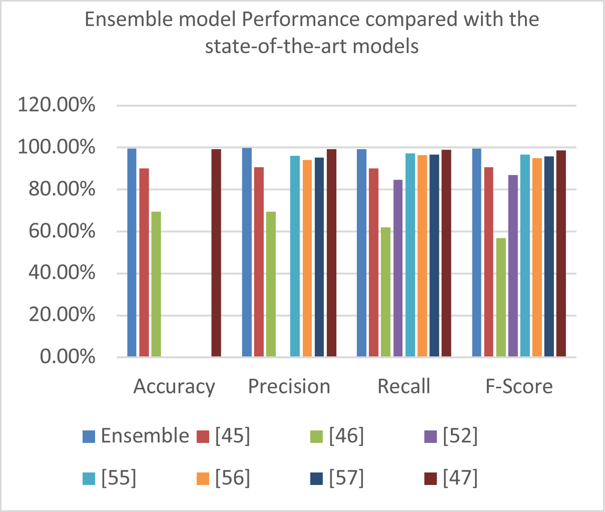

FIGURE 12. Feature importance scores using SHAP techniques for multi-class classification.

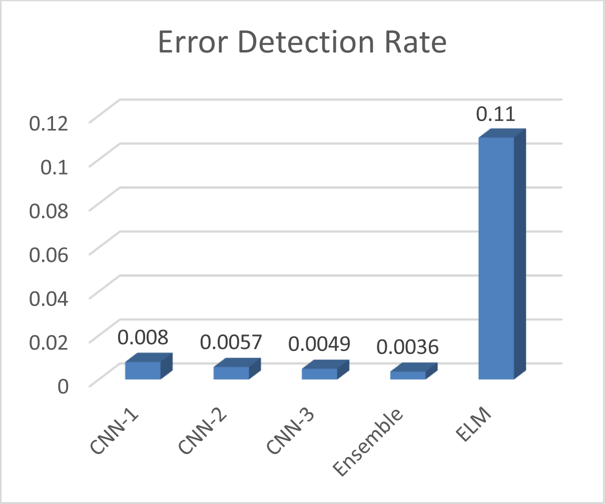

FIGURE 13. Feature importance scores using SHAP techniques for multi-class classification.

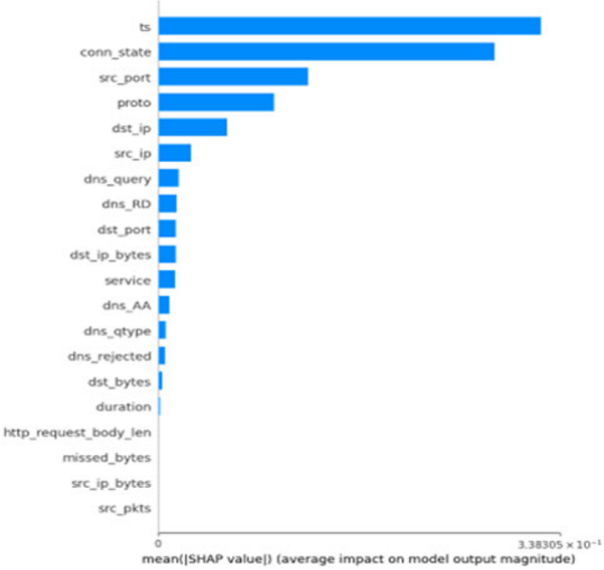

FIGURE 14. Feature importance scores using SHAP techniques for multi-class classification.

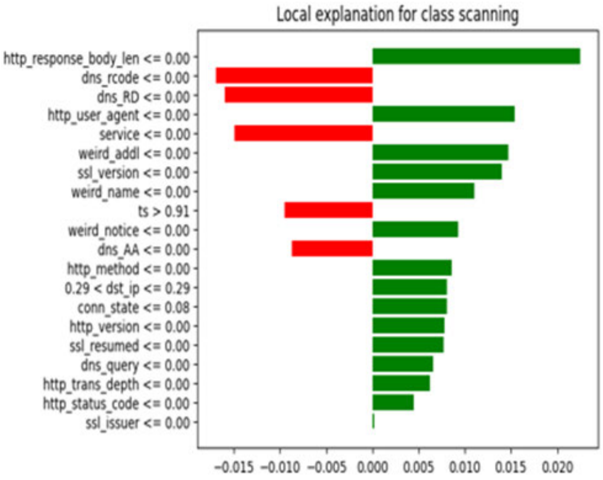

FIGURE 15. Feature importance scores using LIME technique for multi-class classification.

the examination of network traffic patterns, identification of anomalies or attacks, and the correlation of events across multiple systems.

The LIME diagram analysis reveals that certain features significantly impact the prediction. So, for the binary classification and from Fig. 9, we can extract that instances with "ts" values between 0.04 and 0.87 have a positive impact on the prediction, while instances with "weird_addl" values less than or equal to 0.00 have a negative impact. Features such as "ssl_established," "dns_rcode," "http_method," "http_response_body_len," "dns_query," "http_uri," "dns_rejected," "weird_notice," "http_status_code," "dns_RD," "dst_ip," "ssl_cipher," and "http_trans_depth" have values that indicate a positive impact. Conversely, instances with "dst_port" values greater than 0.14 and features like "http_resp_mime_types," "http_user_agent," "missed_bytes," and "src_ip" have values suggesting a negative impact. Understanding the specific values associated with these features helps interpret the model's predictions and make informed decisions based on their contributions.

Based on the analysis of Fig. 14 for multi-class classification, the SHAP results reveal significant features for prediction. One notable feature with a high score is "conn-state," which informs about the current connection state between network devices. This feature holds value in identifying anomalies and potential security threats. Additionally, the "ts" feature, representing the timestamp of network packets, is deemed necessary. It plays a crucial role in detecting patterns and attacks and establishing correlations among events across multiple systems.

Then from fig 15 explains the results of LIME for multi-class classification; features such as "http_response_body_len," "http_user_agent," "weird_add!," "ssl_version," "weird_name," "weird_notice," "http_method," "0.29< dst_ip <= 0.29," "Conn_state," "http_version," "ssl_resumed," "dns_query," "http_trans_depth," "http_status_code," and "ssl_issuer" have values indicating a positive impact on the prediction. These features are associated with the color green and positioned to the right. Conversely, features like "dns_rcode," "dnts_RD," "service," "ts > 0.91," and "dnsAA" have values suggesting a negative impact, represented by the color red and positioned to the left. Understanding the specific values associated with these features aids in interpreting the model's predictions and making informed decisions based on their contributions.

VII. CONCLUSION AND FUTURE WORK

This research introduces an inclusive and efficient framework for an IDS explicitly designed for the intricate context of IIoT networks. The methodology employs the meticulous utilization of the ToN-IoT dataset, which encompasses a wide range of real-world IIoT scenarios, effectively addressing the ever-evolving challenges presented by numerous types of attacks. The core strength of this approach lies in its intelligent fusion of advanced techniques. By harnessing the capabilities of CNN models combined with a novel ensemble strategy, the system achieves accuracy rates exceeding 99%. This outcome underscores the system's exceptional capacity to distinguish between normal and anomalous network behaviors, thus showcasing its robustness and effectiveness. Beyond its predictive capabilities, the methodology significantly enhances transparency and interpretability by incorporating eXplainable AI (XAI) techniques. The strategic integration of SHAP and LIME empowers stakeholders to comprehend the decision-making process and the significance of individual features. This transparency fosters trust and facilitates ongoing improvement of the system's performance and adaptability. Furthermore, this study addresses a significant gap in the field by utilizing a contemporary dataset, the ToN-IoT dataset, which covers a broad spectrum of attacks and mirrors the evolving landscape of IIoT. Departing from conventional datasets equips the methodology with enhanced potential to counter emerging threats and novel attack methods. Looking ahead, refining model architecture, exploring diverse CNN setups, and incorporating industry insights could enhance accuracy. Advanced techniques like autoencoders and real-time monitoring hold potential against sophisticated attacks. Enhanced interpretability through feature selection, testing in varied IIoT environments, and blending CNNs with other methods could improve model understanding. Advancements in XAI methods aid non-technical users in grasping model decisions. To stay effective, continuous dataset updates and collaborative research efforts are essential. Alternative ensemble models such as soft voting or stacking and XAI methods such as LRP or CAM offer deeper insights. This ensures the proposed IDS evolves to secure evolving IIoT networks.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[2] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in Internet of Things: A review," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5482–5494, Mar. 2022.

[3] L. S. Vailshery. (2023). *Number of Internet of Things (IoT) Connected Devices Worldwide From 2019 to 2021, With Forecasts From 2022 to 2030*. Accessed: Mar. 25, 2023. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[4] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of IoT in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156237–156271, 2019.

[5] S. Selvaraj and S. Sundaravaradhan, "Challenges and opportunities in IoT healthcare systems: A systematic review," *Social Netw. Appl. Sci.*, vol. 2, no. 1, p. 139, Jan. 2020.

[6] P. Suler, L. Palmer, and S. Bilan, "Internet of Things sensing networks, digitized mass production, and sustainable organizational performance in cyber-physical system-based smart factories," *J. Self-Governance Manag. Econ.*, vol. 9, no. 2, pp. 42–51, 2021.

[7] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[8] M. K. Hasan, A. K. M. A. Habib, S. Islam, N. Safie, S. N. H. S. Abdullah, and B. Pandey, "DDoS: Distributed denial of service attack in communication standard vulnerabilities in smart grid applications and cyber security with recent developments," *Energy Rep.*, vol. 9, pp. 1318–1326, Oct. 2023.

[9] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1761–1804, 3rd Quart., 2020.

[10] I. Memon, R. A. Shaikh, M. K. Hasan, R. Hassan, A. U. Haq, and K. A. Zainol, "Protect mobile travelers information in sensitive region based on fuzzy logic in IoT technology," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Nov. 2020.

[11] M. K. Hasan, S. Islam, I. Memon, A. F. Ismail, S. Abdullah, A. K. Budati, and N. S. Nafi, "A novel resource oriented DMA framework for Internet of Medical Things devices in 5G network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8895–8904, Dec. 2022.

[12] M. K. Hasan, M. Shafiq, S. Islam, B. Pandey, Y. A. B. El-Ebiary, N. S. Nafi, R. C. Rodriguez, and D. E. Vargas, "Lightweight cryptographic algorithms for guessing attack protection in complex Internet of Things applications," *Complexity*, vol. 2021, pp. 1–13, Apr. 2021.

[13] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.

[14] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021.

[15] I. Ullah, N. U. Amin, M. Zareei, A. Zeb, H. Khattak, A. Khan, and S. Goudarzi, "A lightweight and provable secured certificateless signcryption approach for crowdsourced IIoT applications," *Symmetry*, vol. 11, no. 11, p. 1386, Nov. 2019.

[16] K. Shahzad, M. Alam, N. Javaid, A. Waheed, S. A. Chaudhry, N. Mansoor, and M. Zareei, "SF-LAP: Secure M2M communication in IIoT with a single-factor lightweight authentication protocol," *J. Sensors*, vol. 2022, pp. 1–16, Nov. 2022.

[17] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1–27, Mar. 2021.

[18] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: A review," *J. Big Data*, vol. 6, no. 1, pp. 1–21, Dec. 2019.

[19] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry, "IoT architecture challenges and issues: Lack of standardization," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 731–738.

[20] D. Job and V. Paul, "Challenges, security mechanisms, and research areas in IoT and IIoT," in *Internet of Things and its Applications* (EAI/Springer Innovations in Communication and Computing). 2022, pp. 523–538.

[21] S. F. Tan and A. Samsudin, "Recent technologies, security countermeasure and ongoing challenges of industrial Internet of Things (IIoT): A survey," *Sensors*, vol. 21, no. 19, p. 6647, Oct. 2021.

[22] L. Zhou and H. Guo, "Anomaly detection methods for IIoT networks," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat. (SOLI)*, Jul. 2018, pp. 214–219.

[23] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 113, pp. 418–427, Dec. 2020.

[24] Z. A. E. Houda, B. Brik, and L. Khoukhi, "'Why should I trust your IDS?': An explainable deep learning framework for intrusion detection systems in Internet of Things networks," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1164–1176, 2022.

[25] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[26] W. J. von Eschenbach, "Transparency and the black box problem: Why we do not trust AI," *Philosophy Technol.*, vol. 34, no. 4, pp. 1607–1622, Dec. 2021.

[27] S. Wang, M. Atif Qureshi, L. Miralles-Pechuan, T. Reddy Gadekallu, and M. Liyanage, "Explainable AI for B5G/6G: Technical aspects, use cases, and research challenges," arXiv e-prints, p. arXiv-2112, 2021.

[28] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney, "Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: A systematic review," *Appl. Sci.*, vol. 11, no. 11, p. 5088, May 2021.

[29] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020.

[30] N. Moustafa. (2023). *ToN_IoT Dataset*. [Online]. Available: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.

[32] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[33] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[34] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103160.

[35] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 41611–41660, 2022.

[36] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 241–258, 2020.

[37] O. S. M. B. H. Almazrouei, P. Magalingam, M. K. Hasan, and M. Shanmugam, "A review on attack graph analysis for IoT vulnerability assessment: Challenges, open issues, and future directions," *IEEE Access*, vol. 11, pp. 44350–44376, 2023.

[38] M. K. Hasan, T. M. Ghazal, A. Alkhalifah, K. A. A. Bakar, A. Omidvar, N. S. Nafi, and J. I. Agbinya, "Fischer linear discrimination and quadratic discrimination analysis–based data mining technique for Internet of Things framework for healthcare," *Frontiers Public Health*, vol. 9, Oct. 2021, Art. no. 737149.

[39] G. Alicioglu and B. Sun, "A survey of visual analytics for explainable artificial intelligence methods," *Comput. Graph.*, vol. 102, pp. 502–520, Feb. 2022.

[40] M. D. Oskouei and S. N. Razavi, "An ensemble feature selection method to detect web spam," *Asia–Pacific J. Inf. Technol. Multimedia*, vol. 7, no. 2, pp. 99–113, Dec. 2018.

[41] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, Jul. 2018, Art. no. e1249.

[42] J. Lee, J. Kim, and W. Ko, "Day-ahead electric load forecasting for the residential building with a small-size dataset based on a self-organizing map and a stacking ensemble learning method," *Appl. Sci.*, vol. 9, no. 6, p. 1231, Mar. 2019.

[43] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, pp. 1–11, Jan. 2021.

[44] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding DQNs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1899–1908.

[45] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021.

[46] A. K. Balyan, S. Ahuja, U. K. Lilhore, S. K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar, "A hybrid intrusion detection model using EGA-PSO and improved random forest method," *Sensors*, vol. 22, no. 16, p. 5986, Aug. 2022.

[47] M. A. Almaiah, O. Almomani, A. Alsaaidah, S. Al-Otaibi, N. Bani-Hani, A. K. A. Hwaitat, A. Al-Zahrani, A. Lutfi, A. B. Awad, and T. H. H. Aldhyani, "Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels," *Electronics*, vol. 11, no. 21, p. 3571, Nov. 2022.

[48] R. Kale, Z. Lu, K. W. Fok, and V. L. L. Thing, "A hybrid deep learning anomaly detection framework for intrusion detection," in *Proc. IEEE IEEE 8th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2022, pp. 137–142.

[49] J. Zhang, F. Li, and F. Ye, "An ensemble-based network intrusion detection scheme with Bayesian deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[50] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," *Big Data Res.*, vol. 30, Nov. 2022, Art. no. 100359.

[51] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov, "An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1000–1014, Jan. 2023.

[52] M. Vishwakarma and N. Kesswani, "DIDS: A deep neural network based real-time intrusion detection system for IoT," *Decis. Anal. J.*, vol. 5, Dec. 2022, Art. no. 100142.

[53] I. A. Khan, N. Moustafa, I. Razzak, M. Tanveer, D. Pi, Y. Pan, and B. S. Ali, "XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks," *Future Gener. Comput. Syst.*, vol. 127, pp. 181–193, Feb. 2022.

[54] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, 2022.

[55] F. Chollet, "Keras," GitHub, Tech. Rep., 2015. [Online]. Available: https://github.com/fchollet/keras

[56] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[57] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Comput. Commun.*, vol. 166, pp. 110–124, Jan. 2021.

[58] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in *Proc. 10th EAI Int. Conf. BDTA, 13th EAI Int. Conf. Wireless Internet (WiCON)*, 2021, pp. 117–135.

[59] Y. Alotaibi and M. Ilyas, "Ensemble-learning framework for intrusion detection to enhance Internet of Things' devices security," *Sensors*, vol. 23, no. 12, p. 5568, Jun. 2023.

[60] M. Sarhan, S. Layeghy, and M. Portmann, "Feature analysis for ML-based IIoT intrusion detection," 2021, *arXiv:2108.12732*.

[61] A. Telikani, J. Shen, J. Yang, and P. Wang, "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 23260–23271, Nov. 2022.

[62] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020.

[63] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5087–5095, Aug. 2022.

MOUSA'B MOHAMMAD SHTAYAT received the bachelor's degree in electrical and computer engineering from The Hashemite University, Jordan, in June 2003, and the master's degree in information technology management (ITM) from The University of Sunderland, U.K., in July 2009. He is currently pursuing the Ph.D. degree in cybersecurity with UKM, Malaysia.

He has been working in "Royal Commission for Jubail and Yanbu," as a Lecturer, at the Computer Science and Engineering Department, Yanbu University College (the new name is Yanbu Industrial College), since September 2012. He is teaching many courses, such as computer networks, network security, digital logic, and electrical circuits. In addition, he was a course development member for network courses and digital logic course materials and labs. He is a member of different committees, such as Quality Management System, ISO Auditing as an Internal Auditor, and a COOP Coordinator as a member of the Academic Support Committee.

MOHAMMAD KAMRUL HASAN (Senior Member, IEEE) received the Ph.D. degree in electrical and communication engineering from the Faculty of Engineering, International Islamic University, Malaysia, in 2016. He is currently an Associate Professor and the Head of the Network and Communication Technology Laboratory, Faculty of Information Science and Technology, Center for Cyber Security, Universiti Kebangsaan Malaysia (UKM). He is specialized in elements pertaining to cutting-edge information centric networks, computer networks, data communication and security, mobile network and privacy protection, cyber-physical systems, the Industrial IoT, transparent AI, and electric vehicles networks. He has published more than 230 indexed papers in ranked journals and conference proceedings. He is a member of the Institution of Engineering and Technology and the Internet Society. He is a certified Professional Technologist (P.Tech /Ts.), Malaysia. He served as the Chair for the IEEE Student Branch, from 2014 to 2016. He has actively participated in many events/workshops/trainings for IEEE and IEEE Humanity programs, Malaysia. He is the general chair, the co-chair, and a speaker for conferences and workshops for the shake of society and academy knowledge building, sharing, and learning. He is an editorial member in many prestigious high-impact journals, such as IEEE, IET, Elsevier, Frontier, and MDPI. He has been contributing and working as a volunteer for underprivileged people for the welfare of society.

ROSSILAWATI SULAIMAN received the B.Sc. degree in computer science from Universiti Kebangsaan Malaysia, in 2000, the M.Sc. degree in computer science from the University of Essex, U.K., in 2003, and the Ph.D. degree from the University of Canberra, in 2011.

She is currently a Senior Lecturer with Universiti Kebangsaan Malaysia. Her work has been published in *Journal Theoretical and Applied Information Technology*, *International Journal of Advanced Computer Science and Applications*, and *Journal of Computer Science*. Her research interests include steganography and applied cryptography.

SHAYLA ISLAM (Senior Member, IEEE) received the B.Sc. degree in computer science and engineering from International Islamic University Chittagong, Bangladesh, the M.Sc. degree from the Department of Electrical and Computer Engineering, International Islamic University Malaysia (IIUM), in 2012, and the Ph.D. degree in engineering from the Electrical and Computer Engineering (ECE) Department, IIUM, in 2016, under the Malaysian International Scholarship.

She is currently an Assistant Professor with UCSI University, Malaysia. She was awarded the Silver Medal for her research work with International Islamic University Malaysia. In consequence, she was awarded the Young Scientist Award for the contribution of a research paper at the second International Conference on Green Computing and Engineering Technologies 2016 (ICGCET'16), organized by the Department of Energy Technology, Aalborg University, Esbjerg, Denmark.

ATTA UR REHMAN KHAN (Senior Member, IEEE) is currently an Associate Professor with the College of Engineering and Information Technology, Ajman University, United Arab Emirates. In the past, he was a Postgraduate Program Coordinator with Sohar University; the Director of the National Cybercrime Forensics Laboratory, Pakistan; and the Head of the Cybersecurity Center, Air University. His research interests include cybersecurity, mobile cloud computing, ad hoc networks, and the IoT. He serves as a domain expert for multiple international research funding bodies, and has received multiple awards, fellowships, and research grants. He is a Senior Member of ACM and a Steering Committee Member/track chair/Technical Program Committee (TPC) member of over 85 international conferences. He is serving as an Associate Editor for IEEE Access (Elsevier) and *Journal of Network and Computer Applications*; an Associate Technical Editor for *IEEE Communications Magazine*; and an Editor for *Cluster Computing* (Springer), *The Computer Journal* (Oxford), IEEE SDN Newsletter, *KSII Transactions on Internet and Information Systems*, and *Ad hoc Sensor Wireless Networks*. For more updated information, visit his website at www.attaurrehman.com.