IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 January 2024, accepted 18 March 2024, date of publication 22 March 2024, date of current version 2 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380816

RESEARCH ARTICLE

A Self-Attention-Based Deep Convolutional Neural Networks for IIoT Networks Intrusion Detection

MOHAMMED S. ALSHEHRI[1], OUMAIMA SAIDANI[2], FATMA S. ALRAYES[2], SAADULLAH FAROOQ ABBASI[3], AND JAWAD AHMAD[4], (Senior Member, IEEE)

[1]Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3]Department of Electronic, Electrical and Systems Engineering, University of Birmingham, B15 2TT Birmingham, U.K.
[4]School of Computing, Engineering and the Built Environment, Edinburgh Napier University, EH10 5DT Edinburgh, U.K.

Corresponding author: Saadullah Farooq Abbasi (s.f.abbasi@bham.ac.uk)

The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the research groups funding program grant code (NU/RG/SERC/12/3). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT The Industrial Internet of Things (IIoT) comprises a variety of systems, smart devices, and an extensive range of communication protocols. Hence, these systems face susceptibility to privacy and security challenges, making them prime targets for malicious attacks that can result in harm to the overall system. Privacy breach issues are a notable concern within the realm of IIoT. Various intrusion detection systems based on machine learning (ML) and deep learning (DL) have been introduced to detect malicious activities within these networks and identify attacks. However, traditional ML and DL models encounter significant hurdles when faced with highly imbalanced training data and repetitive patterns within network datasets, hampering their performance in distinguishing between various classes of attacks. To overcome the challenges inherent in existing systems, this paper presents a self-attention-based deep convolutional neural network (SA-DCNN) model designed for monitoring the IIoT networks and detecting malicious activities. The SA mechanism computes the significance value for each input feature, and the DCNN processes these parameters to detect IIoT network behavior. Additionally, a two-step cleaning method has been implemented to eliminate redundancy within the training data, considering both intra-class and cross-class samples. Furthermore, to tackle the issue of underfitting, we have employed a mutual information-based feature filtering method. This method ranks all the features in descending order based on their mutual information and subsequently removes the features with negative impact from the dataset. The performance of the SA-DCNN model is assessed using IoTID20 and Edge-IIoTset datasets. Moreover, the proposed study is demonstrated through a comprehensive comparison with other ML and DL models, as well as against relevant studies, showcasing the superior performance and efficacy of the proposed model.

INDEX TERMS Attention mechanism, CNN, deep learning, IIoT, intrusion detection.

I. INTRODUCTION

The Industrial Internet of Things (IIoT) is an interlinked network of smart devices, sensors, and machines employed within industrial environments for gathering, sharing, and analyzing information [1], [2]. Its primary objectives include elevating operational efficiency, facilitating predictive maintenance, optimizing processes, and enhancing overall productivity across various industries, including manufacturing, energy, transportation, and healthcare [3], [4], [5], [6]. The environment of the IIoT is characterized by a variety of systems, smart devices, and an extensive range of

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

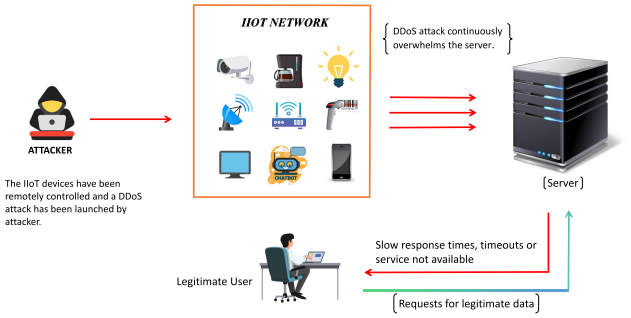


FIGURE 1. A cyber attack scenario on an IIoT network.

communication protocols [7], [8], [9]. Hence, these systems face susceptibility to privacy and security challenges, making them prime targets for malicious attacks that can result in harm to the overall system. Privacy breach issues are a notable concern within the realm of IIoT [10], [11], [12]. Figure 1 illustrates a cyberattack scenario wherein a botnet is employed to initiate a distributed denial of service (DDoS) attack on an industrial IoT network, with a specific focus on industrial servers.

For the security of the IIoT network, numerous researchers and experts have introduced various intrusion detection systems (IDS) designed to identify cyber-attacks within these networks and identify attacks [13], [14], [15], [16]. Machine learning (ML) and deep learning (DL)-based IDSs play a crucial role in identifying malicious attacks due to their generalization capabilities, enabling them to learn from network datasets and recognize previously unseen patterns [17], [18], [19], [20]. The existing ML and DL-based models demonstrate satisfactory performance for a limited number of attack identifications [21]. However, their effectiveness diminishes as the number of classes increases, especially when confronted with highly imbalanced training set data. Additionally, certain network datasets contain repetitive data, leading to inflated model performance on those specific datasets, as the model has encountered much of the test set data during training. Moreover, the ML and DL-based models decrease performance when confronted with datasets that include repetitions of similar data across various classes, where only the class labels are different.

To address these challenges, this study proposes a self-attention DL method for the prediction of intrusions in IIoT networks, along with preprocessing steps to prepare data for the model. The proposed model consists of self-attention (SA) and deep convolutional neural networks (DCNN). The SA computes the significance value for each input attribute [22], and DCNN processes these parameters to detect IIoT network behavior. The primary advantage of DCNN is its ability to converge inputs toward the most impactful parameters and reduce the overall number of parameters [23], [24]. This process enhances detection performance while minimizing time consumption. Additionally, the preprocessing steps involve cleaning, numericalization, feature filtering, and normalization. The cleaning step encompasses sub-processes. Firstly, instances with undefined and missing values are removed. Next, duplication is removed from the datasets. The dataset is scanned for duplication within an attack class and eliminated. Furthermore, the dataset is examined across all classes to identify duplicate instances where only the attack label is changed, and these duplications are removed from all the attack classes. Moreover, we employ the mutual information method for feature filtering. This method ranks features in descending order and removes those features that negatively impact the model, leading to underfitting.

The performance of the SA-DCNN model is assessed using two real-time IoT and IIoT network intrusion detection datasets, namely IoTID20 and Edge-IIoTset. Various evaluation metrics, including precision, recall, F1-score, and accuracy, are employed to assess the performance. Furthermore, to validate the proposed method's performance, it is compared with several other machine learning (ML) and deep learning (DL) models, as well as with findings from related articles. The major contributions of this article are outlined as follows:

- A novel DL-based IDS called SA-DCNN is introduced for the prediction of intrusions in IIoT networks. This model comprises of a self-attention mechanism and the DCNN model. The self-attention mechanism is utilized to compute the significance of each input value, while DCNN processes these values to detect network behaviors.
- In this study, a two-step cleaning process is implemented. The first step involves removing instances with empty and undefined values, while the second step aims to eliminate duplications from the dataset. During the removal of duplications, both intra-class and cross-class duplications are addressed in the datasets.
- A feature filtering method is employed to rank all features in descending order and eliminate those that adversely affect the model's performance, potentially leading to underfitting. Specifically, the mutual information technique is employed for feature filtering, retaining only those features that positively impact the model.
- The effectiveness of the SA-DCNN method has been validated by comparing the outcomes with other ML and DL models. The other methods were implemented under the same experimental environment as the proposed model, and the preprocessing steps were consistent for all models, including the proposed SA-DCNN.

The remainder of the paper is structured as follows: Section II provides an overview of existing works. Section III delves into a detailed presentation of the proposed model. The methodology behind the proposal is expounded upon in Section IV. Section V encompasses a comprehensive discussion of the results, accompanied by a comparison of the SA-DCNN model with other methods. Lastly, Section VI serves as the concluding section for the entire paper.

II. RELATED WORK

The rapid expansion of the IIoT in industrial sectors brings numerous benefits but also exposes vulnerabilities to malicious attackers. Many researchers and experts have been diligently working on improving security and have proposed various methods for identifying malicious attacks within these networks.

Authors in [25] present a DCNN model designed for IoT network monitoring and the identification of malicious activities. The DCNN model was applied to both category and sub-category scenarios to discern the sub-class of attacks. To evaluate the model's performance, the authors utilized the IoTID20 dataset. From the experiments, they achieved an accuracy of 77.55% in detecting malicious activities.

In [26], the authors proposed a hybrid model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for detecting intrusions in IoT network scenarios. The primary emphasis of the authors is on enhancing the model's performance, specifically concentrating on identifying sub-categories of IoT network attacks. They employed the Edge-IIoTset dataset to assess the performance of the CNN-LSTM model. The results showcase a remarkable 98.69% accuracy in classifying attacks.

In [27], the authors introduced an Extreme Learning Machine, Support Vector Machine models, and a rule-based intrusion detection system similar to SNORT for IIoT networks. The performance of the proposed model was evaluated using the KDD99, UNSW-NB15, CSE-CIC-IDS-2018, and Edge-IIoTset datasets. They achieved accuracy rates of 97.83%, 96.59%, 92.54%, and 97.27% for the respective datasets.

In [28], the authors employed the LSTM model for monitoring Software-Defined Networking (SDN)-enabled IoT networks and detecting cyberattacks. The authors specifically concentrated on enhancing the accuracy of Low-Rate Distributed Denial of Service (LDDoS) detection. They utilized the Edge-IIoTset dataset to evaluate the models. The results presented in the paper demonstrate an impressive 98.88% accuracy in the classification of multi-class sub-category classifications.

In [29], the authors introduced a hybrid model that combines a bidirectional gated recurrent unit (B-GRU) and LSTM for identifying cyber attacks in edge-envisioned smart agriculture networks. The authors focused specifically on enhancing the detection of DDoS attacks in these networks. They assessed the model using the Edge-IIoTset dataset, and the experimental outcomes revealed an impressive 98.32% accuracy.

The related studies predominantly concentrate on improving the performance of intrusion detection in IoT and IIoT networks. However, a common limitation in these studies is the oversight of data-cleaning procedures before the training phase. Specifically, there is a lack of attention to addressing redundancies within the data belonging to the same class (intra-class) and neglecting the inclusion of mixed data across different classes (inter-class). Some network datasets used in these studies exhibit repetitive data patterns, which can result in inflated model performance on those particular datasets. This is because the model may encounter much of the test set data during the training phase, leading to an overestimation of its effectiveness. Furthermore, ML and DL models tend to experience a decline in performance when confronted with datasets containing repetitions of similar data across various classes, even if only the class labels differ.

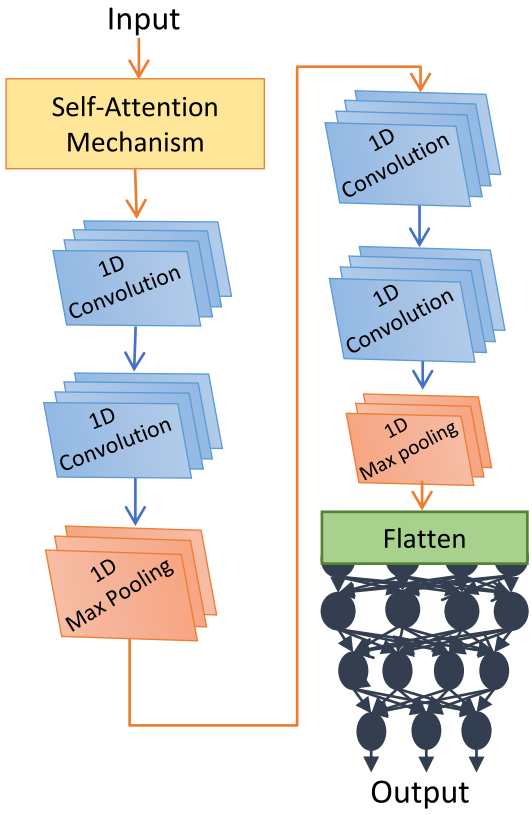


FIGURE 2. Basic architecture of SA-DCNN.

III. THE PROPOSED SA-DCNN MODEL

This study proposes a novel DL model called SA-DCNN for IIoT network traffic monitoring and detection of cyber-attacks. The SA-DCNN consists of a self-attention mechanism and deep convolutional neural networks (DCNN), as depicted in Fig 2. The self-attention mechanism computes the significance value for each input feature, and the DCNN processes these parameters to detect IIoT network behavior. The primary advantage of DCNN is its ability to converge inputs toward the most impactful parameters and reduce the overall number of parameters. This process enhances detection performance while minimizing time consumption.

In the proposed model, the self-attention mechanism is used to compute attention scores and highlight the importance of each input feature. This mechanism calculates the attention score based on queries (Q), keys (k), and values (V). Q, K, and V are computed using Eq 1, Eq 2, and Eq 3, respectively,

where X is the input and W is the learning weight.

$$Q = W_q \cdot X \tag{1}$$
$$K = W_k \cdot X \tag{2}$$
$$V = W_v \cdot X \tag{3}$$

Eq 4 is used to compute the attention score (A_S), where d_q is the length of Q. Subsequently, the attention value (A_V) is computed using Eq 5.

$$A_S = \frac{Q \cdot K^T}{\sqrt{d_q}} \tag{4}$$
$$A_V = \text{softmax}\,(A_S) \cdot V \tag{5}$$

Once the attention value for each input feature is calculated by the SA mechanism, it is then fed into the DCNN layers. The primary advantage of a CNN model lies in its capability to effectively capture the significance of input parameters. Furthermore, CNN operates with fewer parameters compared to recurrent algorithms in deep learning, resulting in improved processing speed [30]. A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers [31]. In DCNN, We used four convolutional layers, two max-pooling layers, a flattening layer, and three fully connected feedforward neural network (FFNN) layers in the proposed SA-DCNN model. The convolutional layers are utilized to emphasize each parameter using a kernel, where the size of the kernel is three. Within this layer, the ReLU activation function is used. The convolutional operation is represented in Eq 6 and 7.

$$x_k = b_k + \sum_{i=1}^{N} (P_i, w_{ik}) \tag{6}$$
$$y_k = \max(0, x_k) \tag{7}$$

where x_k denotes the input in convolutional, while P_k signifies the output of the preceding layer. w_{ik} corresponds to the kernel spanning from index i to k, and b_k denotes the bias associated with the neuron in the convolutional layer. The output of the convolutional layer is passed into the max-pooling layer which selects the most significant parameters as expressed in Eq 8, where M_k is the output of the max-pooling layer.

$$M_k = \max_{i \in \Re} y_k \tag{8}$$

The output of the max-pooling layer is forwarded to the flattening layer, which transforms it into a one-dimensional array. This array is then passed to the fully connected FFNN layers. The FFNN comprises three layers, with the first two layers being hidden layers utilizing the ReLU activation function. The final layer is dedicated to producing output probabilities, and for this purpose, the softmax activation function is employed, as expressed in Eq 9.

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{9}$$

IV. THE PROPOSED APPROACH

This section offers a thorough exploration of the implemented approach as depicted in Figure 3, which highlights its key stages. The framework initiates with an in-depth analysis of the employed dataset, covering various preprocessing stages. Subsequently, the data undergoes stratified splitting into training and testing sets. Following these stages, the model proceeds through training and testing processes.

A. DATASETS

The IoTID20 and Edge-IIoTset are widely recognized and extensively used datasets in the research community. The IoTID20 dataset has been collected from home IoT networks to facilitate the detection of cyber attacks [32]. Its primary advantage stems from the inclusion of up-to-date communication data and innovative samples, enhancing the capability to detect network intrusions [33]. The dataset comprises a total of 625,783 samples, with 40,073 classified as normal and the remaining 585,710 categorized into four types of attacks. Furthermore, these four types of attacks are subdivided into eight sub-types. The Edge-IIoTset includes samples of IoT and IIoT network traffic collected from a testbed consisting of seven layers. Comprising fourteen attacks associated with IoT and IIoT communication protocols [34], the Edge-IIoTset comprises a total of 2,219,201 samples. Among these, 1,615,643 samples are classified as normal, while the remaining 603,558 samples are related to 14 different attacks.

B. DATA PREPROCESSING

The preprocessing steps are important for readying the dataset for optimal compatibility with ML and DL models. This paper employs several preprocessing steps, encompassing data preparation, feature filtering, normalization, and the division of the dataset into train and test sets.

1) DATA PREPARATION

Data preparation is the initial step of preprocessing, involving two main methods: the first is cleaning, and the second is the conversion of categorical attributes into numerical format.

a: CLEANING

The data cleaning process consists of two sub-steps. In the first sub-step, we eliminate instances with undefined and 'Null' values from the dataset using the Pandas library in Python. In the second sub-step, we address duplication in the dataset by employing two methods. Initially, we remove duplications within the same class using the drop_duplicates function from the Pandas library. Subsequently, we eliminate duplications across different classes by considering all attributes instead of just the classification label. For this, we utilize the drop and duplicate functions of Pandas, leveraging indexes to facilitate the process.

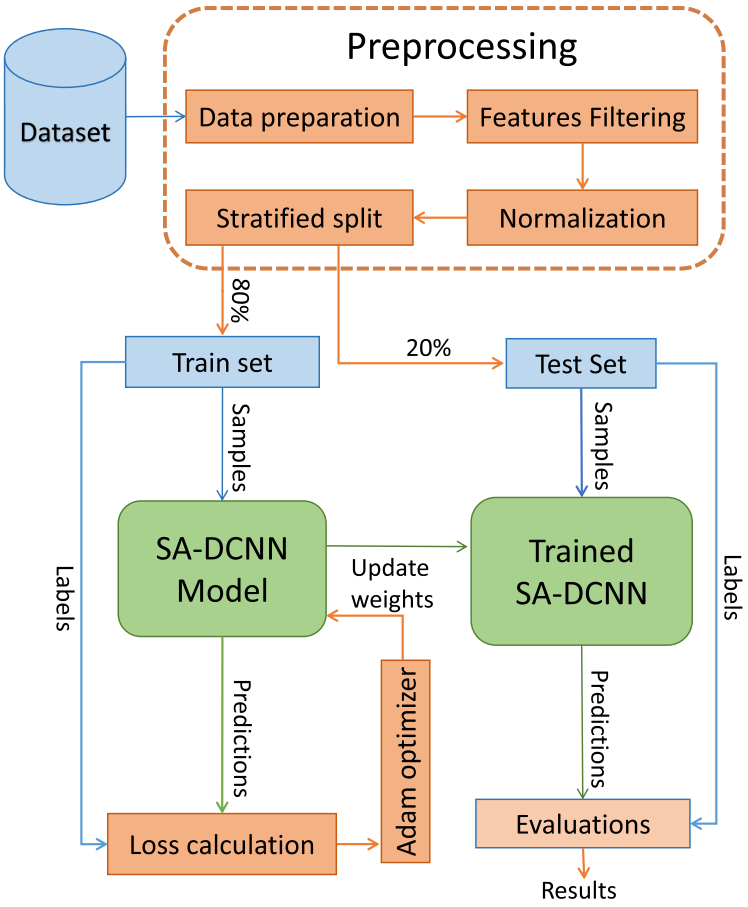


FIGURE 3. The proposed architecture block diagram.

b: FEATURES ENCODING

The datasets we employed contain numerous features in categorical form, necessitating conversion into numerical format for compatibility with DL models aimed at predicting network activity behaviors. To accomplish this, we opted for the label encoder method. This method assigns a unique numerical value to each category of values within an attribute, following an alphabetic order. We chose this approach due to its efficiency in terms of memory usage and processing power, as opposed to the one-hot encoder. The one-hot encoder, while effective, demands additional memory for the conversion of categorical features.

2) FEATURES FILTERING

In this experiment, we employed a feature filtering method to identify influential attributes within the dataset, while excluding features that have a negative impact on the classifier. The negative impact of certain attributes arises due to the amalgamation of data from different classes without providing discernible patterns. To filter the attributes in the utilized datasets, we employed the mutual information method, which demonstrates the impact of each feature and ranks them in descending order based on entropy. We selected all attributes with a value greater than 0.1, eliminating those with zero or near-zero impact values. Out of 83 attributes, 56 were chosen for the IoTID20 dataset, and for the Edge-IIoTset dataset, 29 out of 62 features were selected for the experiment.

3) NORMALIZATION

Normalization involves rescaling data to a standardized range. The performance of classifiers is impacted by features with diverse ranges. The utilized datasets encompass attributes with varying scales, necessitating normalization. In this experiment, we employ the min-max normalization technique to normalize features within the range of 0 to 1, as presented in Equation 10.

$$X_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{10}$$

4) STRATIFIED SPLIT

The stratified method is utilized to divide the data into training and testing sets, maintaining specified percentages to ensure a balanced representation of each class in the splits. In this instance, we applied the stratified approach to allocate 80% of the data to the training set and 20% to the test set.

TABLE 1. Performance assessment with various layers combination on IoTID20 category.

Layers			FFNN hidden units	Performance metrics					
Conv	Max Pool	FFNN		Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
1	1	5	128, 64, 32, 16, 5	0.9543	0.9522	0.9531	0.9794	41	5
1	1	4	64, 32, 16, 5	0.9434	0.9587	0.9505	0.9781	40	5
1	1	3	64, 32, 5	0.9448	0.9495	0.9471	0.9765	38	5
1	1	2	64, 5	0.9412	0.9646	0.9526	0.9788	52	8
2	1	5	128, 64, 32, 16, 5	0.9525	0.9414	0.9465	0.9767	55	7
2	1	4	64, 32, 16, 5	0.9163	0.8641	0.8754	0.9519	54	6
2	1	3	64, 32, 5	0.9464	0.9618	0.9538	0.9795	57	7
2	1	2	64, 5	0.9474	0.9586	0.9528	0.9789	54	6
4	2	5	128, 64, 32, 16, 5	0.9465	0.9449	0.9456	0.9764	90	9
4	2	4	64, 32, 16, 5	0.9537	0.9569	0.9547	0.9801	86	8
4	**2**	**3**	**64, 32, 5**	**0.9535**	**0.9596**	**0.9564**	**0.9805**	**85**	**8**

TABLE 2. Performance assessment with various layers combination on IoTID20 sub-category.

Layers			FFNN hidden units	Performance metrics					
Conv	Max Pool	FFNN		Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
1	1	5	128, 64, 32, 16, 9	0.9001	0.8839	0.8919	0.9645	43	5
1	1	4	64, 32, 16, 9	0.8784	0.8759	0.8771	0.9591	38	5
1	1	3	64, 32, 9	0.9012	0.8445	0.8719	0.9562	41	5
1	1	2	64, 9	0.8871	0.8761	0.8816	0.9655	52	6
2	1	5	128, 64, 32, 16, 9	0.8851	0.8635	0.8741	0.9623	54	6
2	1	4	64, 32, 16, 9	0.8889	0.8774	0.8831	0.9662	56	5
2	1	3	64, 32, 9	0.8889	0.8737	0.8812	0.9645	54	6
2	1	2	64, 9	0.8986	0.8684	0.8832	0.9651	53	6
4	2	5	128, 64, 32, 16, 9	0.9149	0.8595	0.8863	0.9648	69	7
4	2	4	64, 32, 16, 9	0.9011	0.8748	0.8878	0.9674	68	7
4	**2**	**3**	**64, 32, 9**	**0.9239**	**0.8783**	**0.9005**	**0.9689**	**67**	**7**

C. THE PROPOSED SA-DCNN HYPERPARAMETERS

In this experiment, we utilize various hyperparameters to achieve optimal performance. In all convolutional layers, 64 filters, a kernel size of 3, the same padding, and the ReLU activation function are utilized. The max pooling layer employs a pool size of 2. In the feedforward neural network layer, three layers are used. The first two layers are hidden layers with 64 and 32 hidden units, respectively, employing the ReLU activation function. The final layer of the feedforward neural network is the output layer of the model, where the softmax function is employed to produce probabilities for multi-classification.

The sparse categorical cross-entropy function is employed for loss calculation, and the Adam optimizer is utilized to optimize weights during training. A batch size of 32, along with a configured number of 100 epochs, has been selected for the IoTID20 dataset, aiming to facilitate an efficient and effective training process. For the EdgeIIoTset dataset, a batch size of 32 and a configured number of 20 epochs have been chosen to achieve optimal training performance.

V. EXPERIMENTATION AND FINDINGS

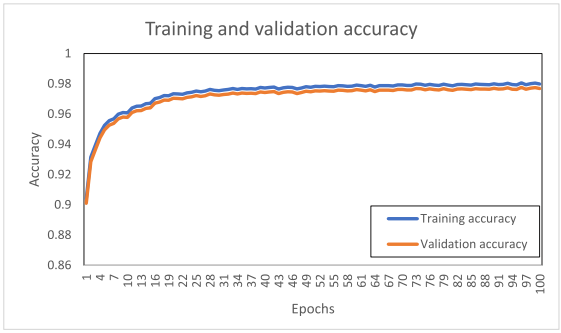
This section primarily focuses on the experimental findings. Initially, it presents the evaluation metrics used in the experiments. Following that, a brief overview of the experimental system environment where all the experiments were conducted is provided. Subsequently, a detailed presentation of the outcomes of the proposed model is given, along with a comparison with other models and state-of-the-art articles. To assess the effectiveness of the proposed SA-DCNN model, we employed four evaluation metrics, namely accuracy, precision, recall, and F1-score.

A. IMPLEMENTATION ENVIRONMENT

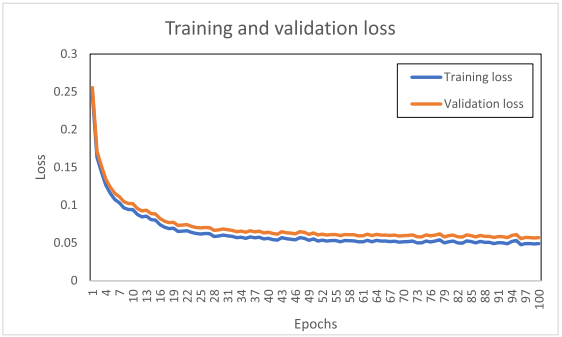
Experiments were conducted on an HP desktop system equipped with a core-i9 nine-generation CPU, a GEFORCE RTX 2080 GPU, and 32 GB of RAM. The Python 3.11 programming language, along with Jupyter Notebook, was employed for the implementation of classifiers. Various libraries, such as Tensorflow, Pandas, sci-kit-learn, and Numpy, were leveraged to support the implementation. It is noteworthy that all these tools were run on a Windows 11 Pro 64-bit operating system to ensure consistency and compatibility.

B. THE PROPOSED SA-DCNN OUTCOMES

In this section, we present the experimental outcomes of the proposed SA-DCNN with various hyperparameter variations for two scenarios: multi-class category and multi-class sub-category classifications, utilizing both datasets. Furthermore, to assess the efficacy of the SA-DCNN model, we conducted experiments with several other traditional ML and DL models in the same environment and compared the results with those of the proposed model. Additionally,

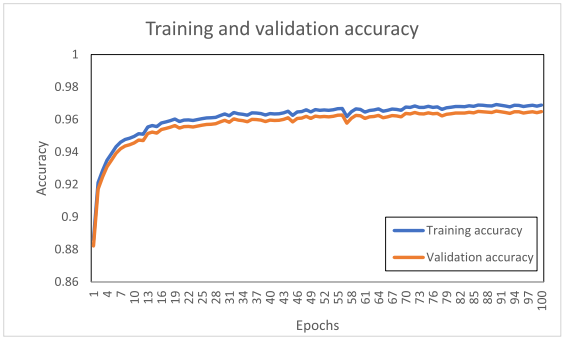


(a) Accuracy

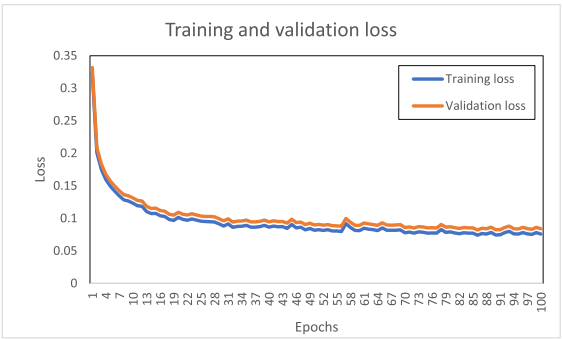


(b) Loss

FIGURE 4. Performance of the proposed SA-DCNN on IoTID20 category.



(a) Accuracy



(b) Loss

FIGURE 5. Performance of the proposed SA-DCNN on IoTID20 sub-category.

TABLE 3. Performance assessment with various layers combination on Edge-IIoTset category.

Layers			FFNN hidden units	Performance metrics					
Conv	Max Pool	FFNN		Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
1	1	5	128, 64, 32, 16, 6	0.9975	0.9916	0.9945	0.9994	157	18
1	1	4	64, 32, 16, 6	0.9981	0.9961	0.9971	0.9996	147	16
1	1	3	64, 32, 6	0.9974	0.9955	0.9964	0.9994	140	15
1	1	2	64, 6	0.9981	0.9962	0.9971	0.9996	155	16
2	1	5	128, 64, 32, 16, 6	0.9979	0.992	0.9949	0.9995	177	19
2	1	4	64, 32, 16, 6	0.9978	0.9936	0.9957	0.9995	168	18
2	1	3	64, 32, 6	0.9919	0.9981	0.9951	0.9995	163	17
2	1	2	64, 6	0.9982	0.9963	0.9972	0.9996	157	16
4	2	5	128, 64, 32, 16, 6	0.8265	0.8273	0.8269	0.9983	239	23
4	2	4	64, 32, 16, 6	0.9924	0.9131	0.9395	0.9983	231	22
4	**2**	**3**	**64, 32, 6**	**0.9983**	**0.9979**	**0.9981**	**0.9996**	**213**	**22**

we compare the performance achieved by the proposed model with state-of-the-art articles on the same datasets to demonstrate the efficacy of the proposed model. The outcomes are validated through a fivefold cross-validation process.

1) OUTCOMES WITH VARIOUS HIDDEN LAYERS ON IoTID20

As mentioned earlier, the datasets were divided into training and testing sets, with proportions of 80% and 20%, respectively. Following that, the model underwent training on the training set using various configurations of hidden layers. The analysis covered two scenarios: multi-class category and sub-category classification. Tables 1 and 2 provide a comprehensive analysis of the testing results for the proposed SA-DCNN model across various layer combinations. Upon evaluating the results, it becomes evident that the proposed SA-DCNN demonstrated optimal performance with four convolutional, two max-pooling, and three fully connected FFNN layers.

Additionally, Figures 4 and 5 depict the training and validation performance, serving as an evaluation of the proposed SA-DCNN model for potential overfitting issues.

TABLE 4. Performance assessment with various layers combination on Edge-IIoTset sub-category.

Layers			FFNN hidden units	Performance metrics					
Conv	Max Pool	FFNN		Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
1	1	5	128, 64, 32, 16, 15	0.9956	0.9893	0.9923	0.9994	154	17
1	1	4	64, 32, 16, 15	0.9952	0.9917	0.9934	0.9995	154	18
1	1	3	64, 32, 15	0.9951	0.9936	0.9943	0.9995	145	16
1	1	2	64, 15	0.997	0.9893	0.9929	0.9995	156	17
2	1	5	128, 64, 32, 16, 15	0.9959	0.9835	0.9892	0.9994	181	20
2	1	4	64, 32, 16, 15	0.9784	0.9919	0.9841	0.9992	173	19
2	1	3	64, 32, 15	0.9967	0.9868	0.9914	0.9995	167	18
2	1	2	64, 15	0.9964	0.9938	0.9951	0.9995	158	17
4	2	5	128, 64, 32, 16, 15	0.9011	0.854	0.8581	0.9975	244	24
4	2	4	64, 32, 16, 15	0.9871	0.9281	0.9365	0.9992	231	23
4	**2**	**3**	**64, 32, 15**	**0.9946**	**0.9961**	**0.9953**	**0.9995**	**223**	**23**

TABLE 5. Results comparison with other models on IoTID20 category.

Models	Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
GNB	0.6737	0.7708	0.6786	0.7741	1	1
LR	0.6349	0.6114	0.6123	0.8682	8	0.1
DNN	0.9169	0.9245	0.9176	0.9643	9	2
DAE	0.9127	0.9147	0.9126	0.9609	9	2
CNN	0.9393	0.9642	0.9512	0.9781	36	4
GRU	0.9122	0.9071	0.9094	0.9569	211	13
LSTM	0.9089	0.8981	0.9033	0.9566	191	16
SA-DCNN	**0.9535**	**0.9596**	**0.9564**	**0.9805**	**85**	**8**

TABLE 6. Results comparison with other models on IoTID20 sub-category.

Models	Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
GNB	0.6948	0.6952	0.6949	0.7875	0.2	0.3
LR	0.5826	0.6234	0.5931	0.8607	11	0.2
DNN	0.9082	0.8556	0.8557	0.9609	9	2
DAE	0.9009	0.8574	0.8508	0.9587	10	2
CNN	0.8945	0.8831	0.8808	0.9659	34	4
GRU	0.786	0.7778	0.7781	0.9293	220	14
LSTM	0.8855	0.8518	0.8466	0.9545	204	18
SA-DCNN	**0.9239**	**0.8783**	**0.9005**	**0.9689**	**67**	**7**

Accuracy and loss during each epoch were scrutinized for both the training and validation results. The visual analysis of the training and validation results reveals closely aligned performance, indicating that the proposed model did not demonstrate signs of overfitting.

2) OUTCOMES WITH VARIOUS HIDDEN LAYERS ON EDGE-IIoTSET

For the Edge-IIoTset dataset, a similar experimental setup was employed as the IoTID20 dataset, with the data partitioned into 80% for training and 20% for testing. The SA-DCNN model underwent training on the training set with varying hidden layer configurations to explore its performance. The evaluation focused on multi-class category and sub-category classification scenarios. Tables 3 and 4 present a detailed examination of the testing outcomes

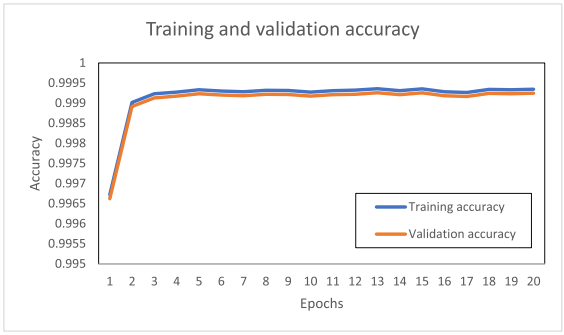
TABLE 7. Results comparison with other models on Edge-IIoTset category.

Models	Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
GNB	0.8193	0.8132	0.7944	0.9331	1	0.5
LR	0.9431	0.9171	0.9291	0.9808	55	0.1
DNN	0.9972	0.9974	0.9973	0.9993	99	11
DAE	0.9981	0.9982	0.9981	0.9995	113	12
CNN	0.9971	0.9962	0.9966	0.9991	142	16
GRU	0.9941	0.9959	0.9951	0.9991	741	82
LSTM	0.9951	0.9972	0.9961	0.9989	619	63
SA-DCNN	**0.9983**	**0.9979**	**0.9981**	**0.9996**	**213**	**22**

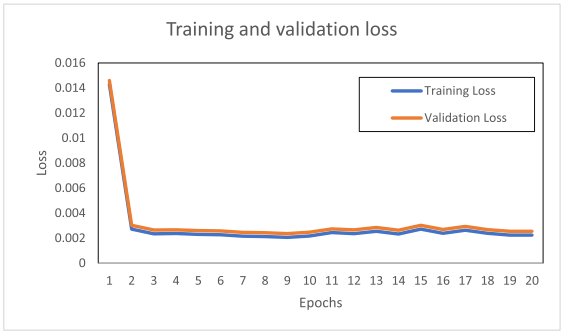
across different layer combinations for the Edge-IIoTset dataset. Notably, the SA-DCNN model exhibited optimal performance when configured with four convolutional layers, two max-pooling layers, and three fully connected FFNN layers. To further assess the model's generalization capability, Figures 6 and 7 illustrate the training and validation performance, ensuring the absence of overfitting concerns. The alignment of accuracy and loss trends across epochs for both training and validation sets indicates the robustness of the proposed SA-DCNN architecture in handling the Edge-IIoTset dataset.

3) PERFORMANCE COMPARISON WITH OTHER ML AND DL MODELS

The effectiveness of the SA-DCNN model was affirmed through a comprehensive validation process, which involved comparing its outcomes with those of various cutting-edge methods. For comparison, traditional ML and sophisticated DL models were employed, encompassing the multi-layer perceptron (MLP), gaussian naive Bayes (GNB), linear regression (LR), deep-autoencoder (DAE), LSTM, GRU, and CNN. It's noteworthy that these models were executed within the same environment, incorporating identical preprocessing steps as the proposed model. This approach ensured an equitable and meaningful assessment of their respective performances. All the implemented DL models utilized the sparse categorical cross-entropy loss function, employed the Adam optimizer, and were trained with a batch size of 32. The training phase of each model was iterated for 100 epochs

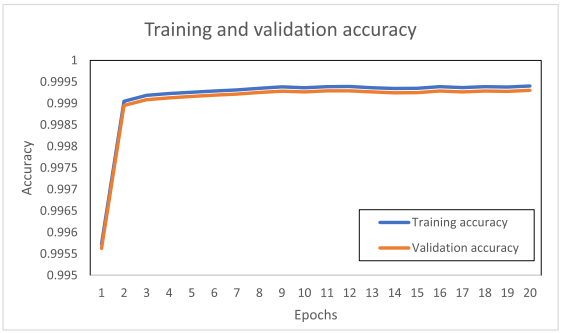


(a) Accuracy

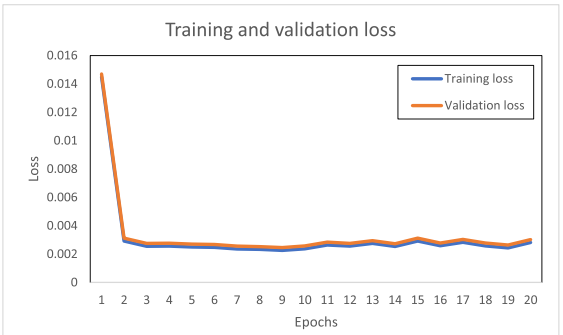


(b) Loss

FIGURE 6. Performance of the proposed SA-DCNN on Edge-IIoTset category.



(a) Accuracy



(b) Loss

FIGURE 7. Performance of the proposed SA-DCNN on Edge-IIoTset sub-category.

TABLE 8. Results comparison with other models on Edge-IIoTset sub-category.

Models	Precision	Recall	F1-score	Accuracy	Training Time (in sec)	Test Time (in sec)
GNB	0.9597	0.9318	0.9398	0.9962	1	2
LR	0.9598	0.8841	0.8947	0.9932	107	0.2
DNN	0.9971	0.9893	0.9931	0.9995	104	11
DAE	0.9959	0.9948	0.9954	0.9995	121	14
CNN	0.9977	0.9951	0.9963	0.9996	131	15
GRU	0.9941	0.9959	0.9951	0.9991	622	66
LSTM	0.9951	0.9972	0.9961	0.9989	733	82
SA-DCNN	**0.9946**	**0.9961**	**0.9953**	**0.9995**	**223**	**23**

TABLE 9. Performance comparison with related articles.

Dataset	Articles	Precision	Recall	F1-score	Accuracy
IoTID20	[25]	0.7867	0.7343	0.7600	0.7755
	This study	0.9239	0.8783	0.9005	0.9689
Edge-IIoTset	[26]	-	-	-	0.9869
	[27]	0.9603	-	-	0.9727
	[28]	0.9746	0.9657	0.9691	0.9888
	[29]	0.9878	0.9722	-	0.9832
	[35]	0.885	0.613	0.724	-
	This study	0.9946	0.9961	0.9953	0.9995

on the IoTID20 dataset and 20 epochs on the Edge-IIoTset dataset.

a: PERFORMANCE COMPARISON ON IoTID20 DATASET

The comparative analysis of testing performance between the proposed SA-DCNN and alternative models is outlined in Table 5 for category classification and Table 6 for sub-category classification on the IoTID20 dataset, respectively. The examination of test results highlights the superior performance of the proposed model over other models.

b: PERFORMANCE COMPARISON ON EDGE-IIoTset DATASET

The comparison on the Edge-IIoTset dataset is presented in Table 7 and Table 8 for category and sub-category classification, respectively. Evaluation of the test results shows that the proposed model gives optimal performances compared to other algorithms, with superior performance in detecting malicious activities within IIoT networks.

c: PERFORMANCE COMPARISON WITH RELATED ARTICLES

To evaluate the enhancement in the detection performance of the proposed study, encompassing both preprocessing and model performance, we conducted a comparative analysis with state-of-the-art articles related to the same dataset. Detailed results comparisons are presented in Table 9 showcasing an in-depth examination of the outcomes from other related articles and our study. The analysis of results demonstrates an improvement compared to existing studies,

highlighting its excellent capabilities in efficiently detecting malicious activities within IIoT networks.

VI. CONCLUSION

This paper introduces a self-attention-based deep convolutional neural network (SA-DCNN) model designed for monitoring IIoT networks and detecting malicious activities. Additionally, a two-step cleaning method has been implemented to eliminate redundancy within the training data, considering both intra-class and cross-class samples. The proposed method overcomes the existing DL-based model's challenges and improves the detection performance of cyberattacks in the IIoT network. The performance of the SA-DCNN model is assessed using IoTID20 and Edge-IIoTset datasets. The proposed model demonstrates 96.89% accuracy, 92.39% precision, 87.83% recall, and 90.05% F1 score on the IoTID20 dataset. Additionally, it achieves 99.95% accuracy, 99.46% precision, 99.61% recall, and a 99.53% F1 score on the Edge-IIoTset dataset. These results represent optimal performance when compared to other traditional ML and DL paradigms. The other models were implemented under the same experimental environment, and the preprocessing steps were consistent for all models, including the proposed SA-DCNN. Furthermore, the outcomes of this study were compared with the results of other related articles, indicating the improved performance of the proposed study. In the future, the number of attack classes is expected to increase further, considering additional subcategories of attacks.

ACKNOWLEDGMENT

The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the research groups funding program grant code (NU/RG/SERC/12/3). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R319), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial Internet of Things traffic in fog environment," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7704–7715, Nov. 2021.

[2] M. M. Alani, "An explainable efficient flow-based industrial IoT intrusion detection system," *Comput. Electr. Eng.*, vol. 108, May 2023, Art. no. 108732.

[3] O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K.-K.-R. Choo, "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT," *Comput. Secur.*, vol. 127, Apr. 2023, Art. no. 103097.

[4] P. Ruzafa-Alcázar, P. Fernández-Saura, E. Mármol-Campos, A. González-Vidal, J. L. Hernández-Ramos, J. Bernal-Bernabe, and A. F. Skarmeta, "Intrusion detection based on privacy-preserving federated learning for the industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1145–1154, Feb. 2023.

[5] M. H. Jamal, M. A. Khan, S. Ullah, M. S. Alshehri, S. Almakdi, U. Rashid, A. Alazeb, and J. Ahmad, "Multi-step attack detection in industrial networks using a hybrid deep learning architecture," *Math. Biosci. Eng.*, vol. 20, no. 8, pp. 13824–13848, 2023.

[6] S. Li, G. Chai, Y. Wang, G. Zhou, Z. Li, D. Yu, and R. Gao, "CRSF: An intrusion detection framework for industrial Internet of Things based on pretrained CNN2D-RNN and SVM," *IEEE Access*, vol. 11, pp. 92041–92054, 2023.

[7] Y. Zhang, C. Yang, K. Huang, and Y. Li, "Intrusion detection of industrial Internet-of-Things based on reconstructed graph neural networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2894–2905, Sep./Oct. 2023.

[8] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An effective intrusion detection approach based on ensemble learning for IIoT edge computing," *J. Comput. Virol. Hacking Techn.*, vol. 19, no. 4, pp. 469–481, Dec. 2022.

[9] M. Nuaimi, L. C. Fourati, and B. B. Hamed, "Intelligent approaches toward intrusion detection systems for industrial Internet of Things: A systematic comprehensive review," *J. Netw. Comput. Appl.*, vol. 215, Jun. 2023, Art. no. 103637.

[10] M. Tanveer and S. Shabala, "Entangling the interaction between essential and nonessential nutrients: Implications for global food security," in *Plant Nutrition and Food Security in the Era of Climate Change*. Amsterdam, The Netherlands: Elsevier, 2022, pp. 1–25.

[11] M. Tanveer, A. Badshah, A. U. Khan, H. Alasmary, and S. A. Chaudhry, "CMAF-IIoT: Chaotic map-based authentication framework for industrial Internet of Things," *Internet Things*, vol. 23, Oct. 2023, Art. no. 100902.

[12] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the IoT and industrial IoT: A review," *Internet Things*, vol. 10, Jun. 2020, Art. no. 100081.

[13] E. Gyamfi and A. D. Jurcut, "Novel online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3827–3839, Mar. 2023.

[14] S. Ullah, W. Boulila, A. Koubaa, Z. Khan, and J. Ahmad, "ABDNN-IDS: Attention-based deep neural networks for intrusion detection in industrial IoT," in *Proc. IEEE 98th Veh. Technol. Conf. (VTC-Fall)*, Oct. 2023, pp. 1–5.

[15] N. W. Khan, M. S. Alshehri, M. A. Khan, S. Almakdi, N. Moradpoor, A. Alazeb, S. Ullah, N. Naz, and J. Ahmad, "A hybrid deep learning-based intrusion detection system for IoT networks," *Math. Biosciences Eng.*, vol. 20, no. 8, pp. 13491–13520, 2023.

[16] S. Aldhaheri and A. Alhuzali, "SGAN-IDS: Self-Attention-Based generative adversarial network against intrusion detection systems," *Sensors*, vol. 23, no. 18, p. 7796, Sep. 2023.

[17] B. B. Zarpelão, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[18] S. Ullah, M. A. Khan, J. Ahmad, S. S. Jamal, Z. E. Huma, M. T. Hassan, N. Pitropakis, and W. J. Buchanan, "HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles," *Sensors*, vol. 22, no. 4, p. 1340, Feb. 2022.

[19] T. Wang, J. Li, W. Wei, W. Wang, and K. Fang, "Deep-Learning-Based weak electromagnetic intrusion detection method for zero touch networks on industrial IoT," *IEEE Netw.*, vol. 36, no. 6, pp. 236–242, Nov. 2022.

[20] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: A comprehensive review and future directions," *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, Dec. 2023.

[21] S. Ullah, W. Boulila, A. Koubâa, and J. Ahmad, "MAGRU-IDS: A multi-head attention-based gated recurrent unit for intrusion detection in IIoT networks," *IEEE Access*, vol. 11, pp. 114590–114601, 2023.

[22] S. Ullah, J. Ahmad, M. A. Khan, M. S. Alshehri, W. Boulila, A. Koubaa, S. U. Jan, and M. M. I. Ch, "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT networks," *Comput. Netw.*, vol. 237, Dec. 2023, Art. no. 110072.

[23] I. Al-Turaiki and N. Altwaijry, "A convolutional neural network for improved anomaly-based network intrusion detection," *Big Data*, vol. 9, no. 3, pp. 233–252, Jun. 2021.

[24] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105124.

[25] S. Ullah, J. Ahmad, M. A. Khan, E. H. Alkhammash, M. Hadjouni, Y. Y. Ghadi, F. Saeed, and N. Pitropakis, "A new intrusion detection system for the Internet of Things via deep convolutional neural network and feature engineering," *Sensors*, vol. 22, no. 10, p. 3607, May 2022.

[26] A. Khacha, R. Saadouni, Y. Harbi, and Z. Aliouat, "Hybrid deep learning-based intrusion detection system for industrial Internet of Things," in *Proc. 5th Int. Symp. Informat. its Appl. (ISIA)*, Nov. 2022, pp. 1–6.

[27] P. Dini, A. Begni, S. Ciavarella, E. De Paoli, G. Fiorelli, C. Silvestro, and S. Saponara, "Design and testing novel one-class classifier based on polynomial interpolation with application to networking security," *IEEE Access*, vol. 10, pp. 67910–67924, 2022.

[28] A. A. Alashhab, M. S. M. Zahid, A. Muneer, and M. Abdukkahi, "Low-rate DDoS attack detection using deep learning for SDN-enabled IoT networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, pp. 1–7, 2022.
[29] D. Javeed, T. Gao, M. S. Saeed, and P. Kumar, "An intrusion detection system for edge-envisioned smart agriculture in extreme environment," *IEEE Internet Things J.*, early access, 2023, doi: 10.1109/JIOT.2023.3288544.
[30] G. Scarpa, M. Gargiulo, A. Mazza, and R. Gaetano, "A CNN-based fusion method for feature extraction from sentinel data," *Remote Sens.*, vol. 10, no. 2, p. 236, Feb. 2018.
[31] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Comput.*, vol. 24, no. 22, pp. 17265–17278, Nov. 2020.
[32] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "IoT network intrusion dataset," IEEE Dataport, Sep. 2019, doi: 10.21227/q70p-q449.
[33] I. Ullah and Q. H. Mahmoud, "A technique for generating a botnet dataset for anomalous activity detection in IoT networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Ottawa, ON, Canada. Cham, Switzerland: Springer, Oct. 2020, pp. 508–520.
[34] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.
[35] T. Shen, L. Ding, J. Sun, C. Jing, F. Guo, and C. Wu, "Edge computing for IoT security: Integrating machine learning with key agreement," in *Proc. 3rd Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2023, pp. 474–483.

MOHAMMED S. ALSHEHRI received the B.S. degree in computer science from King Khalid University, Abha, Saudi Arabia, in 2010, the M.S. degree in computer science from the University of Colorado Denver, Denver, CO, USA, in 2014, and the Ph.D. degree in computer science with concentration on information security from the University of Arkansas, Fayetteville, AR, USA, in 2021. He received a Graduate Certificate in cybersecurity from the University of Arkansas, in 2020. His research interests include cybersecurity, computer networks, blockchain, machine learning, and deep learning.

OUMAIMA SAIDANI received the M.Sc. degree in computer sciences from Paris Dauphine University, France, and the Ph.D. degree in computer sciences from Paris 1-Panthéon Sorbonne University, France. She is currently an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences (CCIS-IS), Princess Nourah bint Abdulrahman University (PNU), Saudi Arabia. Her research interests include information systems engineering, business process engineering, the IoT, context-aware computing, deep learning, and artificial intelligence.

FATMA S. ALRAYES received the M.Sc. degree in e-business and information systems from Newcastle University and the Ph.D. degree in computer science and informatics from Cardiff University. She is currently an Associate Professor with the Information Systems Department, College of Computer and Information Sciences (CCIS-IS), Princess Nourah bint Abdulrahman University (PNU), Saudi Arabia. Her research interests include privacy protection, usable privacy and security, privacy awareness, data science and analytics, and social web.

SAADULLAH FAROOQ ABBASI is currently an experienced Researcher with more than ten years of cutting-edge research and teaching experience in prestigious institutes, including the University of Birmingham, U.K.; the University of Glasgow, U.K.; Fudan University, Shanghai, China; and the National University of Science and Technology Islamabad, Pakistan. He has taught various courses both at undergraduate (UG) and postgraduate (PG) levels during his career. He has coauthored more than 20 research papers in international journals and peer-reviewed international conference proceedings. His research interests include machine learning, artificial intelligence, security, and biomedical engineering. He is an invited reviewer for numerous world-leading high-impact journals (reviewed more than 100 journal articles to date).

JAWAD AHMAD (Senior Member, IEEE) is a highly experienced Teacher with more than 13 years of teaching and research experience in prestigious institutes. He has taught at renowned institutions such as Edinburgh Napier University (U.K.), Glasgow Caledonian University (U.K.) and HITEC University Taxila (Pakistan) etc. He has also served as a supervisor for several Ph.D., M.Sc., and bachelor's students, providing guidance and support for their dissertations. He has published in renowned journals including IEEE Transactions, ACM *Transactions*, Elsevier, and Springer with over 180 research papers and 5000 citations (H-Index 40). For the past three years, his name has appeared on the list of the world's top 2% scientists in Cybersecurity, as published by Clarivate (a list endorsed by Stanford University, USA). Furthermore, in 2020, he received the endorsement of U.K., exceptional talent candidate ('Emerging Leader') for pioneering work in the field of Cybersecurity and AI. To date, he has secured research and funding grants of more than £200K as a Principal Investigator (PI) and a Co-Investigator (Co-I). In terms of academic achievements, he has earned a Gold medal for his outstanding performance in M.S. and a Bronze medal for his achievements in B.S.