


SACNN-IDS: A self-attention convolutional neural network for intrusion detection in industrial internet of things

Mimonah Al Qathrady¹ | Safi Ullah² | Mohammed S. Alshehri³ | Jawad Ahmad⁴  | Sultan Almakdi³ | Samar M. Alqhtani¹ | Muazzam A. Khan^{2,5} | Baraq Ghaleb⁴

¹Department of Information Systems, College of Computer Science and Information Systems, Najran University, Najran, Saudi Arabia

²Department of Computer Science, Quaid-i-Azam University, Islamabad, Pakistan

³Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran, Saudi Arabia

⁴School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, UK

⁵ICESCO Chair Big Data Analytics and Edge Computing, Quaid-i-Azam University, Islamabad, Pakistan

Correspondence

Jawad Ahmad.
Email: J.Ahmad@napier.ac.uk

Funding information

Deputy for Research and Innovation - Ministry of Education, Kingdom of Saudi Arabia, Grant/Award Number: NU/IFC/02/SERC/-/31; Institutional Funding Committee at Najran University, Kingdom of Saudi Arabia

Abstract

Industrial Internet of Things (IIoT) is a pervasive network of interlinked smart devices that provide a variety of intelligent computing services in industrial environments. Several IIoT nodes operate confidential data (such as medical, transportation, military, etc.) which are reachable targets for hostile intruders due to their openness and varied structure. Intrusion Detection Systems (IDS) based on Machine Learning (ML) and Deep Learning (DL) techniques have got significant attention. However, existing ML and DL-based IDS still face a number of obstacles that must be overcome. For instance, the existing DL approaches necessitate a substantial quantity of data for effective performance, which is not feasible to run on low-power and low-memory devices. Imbalanced and fewer data potentially lead to low performance on existing IDS. This paper proposes a self-attention convolutional neural network (SACNN) architecture for the detection of malicious activity in IIoT networks and an appropriate feature extraction method to extract the most significant features. The proposed architecture has a self-attention layer to calculate the input attention and convolutional neural network (CNN) layers to process the assigned attention features for prediction. The performance evaluation of the proposed SACNN architecture has been done with the Edge-IIoTset and X-IIoTID datasets. These datasets encompassed the behaviours of contemporary IIoT communication protocols, the operations of state-of-the-art devices, various attack types, and diverse attack scenarios.

KEYWORDS

convolutional neural network, deep learning, industrial internet of things, intrusion detection, self-attention

1 | INTRODUCTION

Researchers have shown a great deal of interest in the Internet of Things (IoT), which is considered to be one of the most advanced technologies. The proliferation trend of IoT applications has grabbed several industries including agriculture, health, smart security, air and water pollution, transport, smart cities, and smart homes to implement IoT features [1, 2]. Industrial IoT is a subcategory of IoT that refers to the

expansion of IoT into industrial sectors such as factory floors and warehouses. Industrial Internet of Things (IIoT) consists of interlinked smart machinery and real-time analytics systems and intelligent services that process the data produced by those machines [3–6]. For instance, an IIoT-managed inventory system can handle ordering supplies just before they run out of stock, significantly simplifying the task of maintaining inventory and freeing up the employee to do other duties [7].

Abbreviations: ACC, accuracy; AE, autoencoder; AUC, area under the curve; CNN, convolutional neural network; DL, deep learning; DNN, deep neural network; GRU, gated recurrent units; IDS, intrusion detection system; IIoT, industrial internet of things; IoT, internet of things; LR, linear regression; LSTM, long short-term memory; ML, machine learning; MLP, multi-layer perceptron; NB, naive bayes; P, precision; R, recall; SACNN, self-attention convolutional neural network; SVM, support vector machine.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

Nevertheless, despite the IIoT paradigm's unquestionable benefits, such benefits are accompanied by critical security flaws [8]. For example, numerous IIoT applications deal with sensitive and confidential data, such as medical, transportation or military systems making them attractive targets for adversarial intruders [9–11]. Additionally, IIoT devices are usually poorly designed with many overlooked security aspects rendering them vulnerable to reprogramming and manipulating. This could have a profound impact on the IIoT system, resulting in significant economic and reputational losses. To overcome such security concerns, numerous Intrusion Detection Systems (IDS) in which Machine Learning (ML) and Deep Learning (DL) play vital roles have been developed [12–15]. However, existing ML and DL-based IDS still face a number of obstacles that must be overcome. First, the state-of-the-art IDS systems have difficulty detecting multi-class category and multi-class sub-category attacks efficiently. Second, existing DL approaches require an extremely large volume of data for efficiently training the models placing a high burden on the limited storage and computing resources of IIoT devices. Third, existing IDS techniques address primarily binary classification with balanced datasets rendering them inefficient for multi-class learning with imbalanced datasets.

This paper proposes a self-attention convolutional neural network (SACNN) architecture and a suitable feature extraction technique to identify malicious activity in IIoT networks, addressing the limitations of current IDS approaches. The proposed system handles the imbalance issue of input data by dividing input features into equal sizes of vectors and processing them in parallel. Parallel processing of these vectors increases the learning rate and improves the detection performance of malicious activities in IIoT networks. The proposed architecture has a self-attention layer to calculate the input attention and CNN layers to process the assigned attention features for prediction. The fundamental benefit of CNN over other DL algorithms is its ability to capture the importance of features [16]. Moreover, CNN operates with fewer parameters, resulting in faster performance [17]. The performance assessment of the proposed SACNN architecture has been done with the Edge-IIoTset and X-IIoTTID datasets. Edge-IIoTset contains 14 attacks associated with IoT and IIoT communication protocols that are classified into five classes: DoS/DDoS, Information gathering, Man in the middle, Injection, and Malware attacks [18]. On the other hand, the X-IIoTTID dataset comprises real-time IIoT network traffic data, encompassing contemporary IIoT communication protocol behaviours, state-of-the-art devices operations, a wide range of attack types, diverse attack scenarios, and various attack protocols. Moreover, several ML and DL algorithms were tested in the same environment and compared with the SACNN architecture. In summary, this article introduces the following contributions:

- A novel approach self-attention convolutional neural network (SACNN) is proposed for the detection of malicious activities in IIoT networks. It focuses on multi-class categories and multi-class sub-categories of attacks. A self-attention layer is used to divide input features into equal

sizes of heads and process them in parallel to calculate the attention value for each head, and CNN layers are used to process the calculated attention and predict the communication activities in the network.

- A new feature extraction approach based on the extra tree classifier (ETC) is adopted to enable more efficient extraction of the most significant features.
- An extensive empirical evaluation has been conducted using both heavy and light versions of the dataset to showcase the effectiveness of the SACNN model in comparison to state-of-the-art methods.

This paper is organised as follows: Section II overviews the latest related works pertaining to IDS in IIoT-based networks. Section III describes the pre-processing steps and our proposed SACNN model highlighting its main features. Section IV thoroughly discusses experiments and performance evaluation results. Finally, Section V contains the conclusion of the paper.

2 | RELATED WORK

Several experts have been actively dedicated to enhancing the security of IIoT networks. Significant research efforts have recently been directed towards developing more efficient DL-based models for intrusion detection. Li et al. [19] designed a multi-CNN fusion paradigm for the identification of cyberattacks in IIoT network communication. They evaluate the designed paradigm with the NSL-KDD dataset. Bovenzi et al. [20] introduced a multimodal deep autoEncoder (M2-DAE) method for identifying cyberattacks in IoT network communications. The model's performance was evaluated using the Bot-IoT dataset, and it achieved an F1-score of 99% on the utilised dataset.

Abdel-Basset et al. [21] presented a forensics-based DL framework for identifying malicious attacks in IIoT traffic. The presented framework was tested in a fog computing environment and the Bot-IIoT dataset was used to prove the efficiency of the presented framework. Kasongo [22] proposed a genetic algorithm (GA) for attributes extraction and random forest method to detect intrusions in IIoT network communication. They utilised UNSW-NB15 dataset to assess the effectiveness of the model.

Liu et al. [23] proposed a variational autoencoder (VAE) paradigm for intrusion detection utilising a conditional balancing strategy. The VAE paradigm was evaluated using the CSE-CIC-IDS2018 dataset. Telikani et al. [24] designed a combined architecture of stacked autoencoders and CNNs for malicious activity detection. They utilised ToN-IoT and UNSW-NB15 datasets to assess the efficacy of the model. Zhang et al. [25] adopted IDS based on the graph neural network (GNN) to detect cyberattacks in IIoT networks communications. They utilised Mississippi cyberattack datasets to validate the proposed method. Khan et al. [26] designed a DAE IDS based on LSTM networks to distinguish between normal and malicious traffic in the IIoT networks. To analyse the effectiveness of the proposed system, the UNSW-NB15 dataset was utilised. Le et al. [27]

presented an extreme gradient boosting (XGboost) paradigm for detecting malicious activities in IIoT networks focusing on imbalanced datasets. X-IIoTDS and TON_IoT datasets were utilised to assess the efficacy of the presented algorithm. Li et al. [28] proposed a hybrid architecture of CNN and Bi-directional long short-term memory (BiLSTM) for the classification of cyberattacks in IIoT networks. The proposed architecture was analysed with the NSL-KDD dataset.

Altunay et al. [29] introduced a hybrid DL model known as CNN-LSTM for detecting cyberattacks in industrial IoT. They assessed the CNN-LSTM model using the UNSW-NB15 and X-IIoTID datasets, achieving detection accuracies of 92.9% and 99.8%, respectively. Lilhore et al. [30] developed an Optimised CNN-LSTM architecture for detecting suspicious network flow in industrial IoT. They evaluated the proposed system using the ToN-IoT and UNSW-NB15 datasets. The designed model achieved precision rates of 92.7% and 94.25% for the utilised datasets, respectively. Wang et al. [31] presented a combined ResNet, Transformer, and BiLSTM (Res-Tran-BiLSTM) algorithm for the detection of malicious activities in IIoT. The model was evaluated using the NSL-KDD and CIC-

IDS2017 datasets, and they addressed the data imbalance issue by applying the SMOTE method. The results of the presented model demonstrated detection accuracies of 90.99%, 99.15%, and 99.56% on the utilised datasets, respectively.

Table 1 provides an overview of the related work of cyber-attack prediction in IIoT. The analysis of the relevant literature reveals that most studies have focused on a limited number of attacks due to data imbalance issues in the datasets. As a result, when these systems are confronted with a diverse range of attack classes, these systems face challenges in achieving precise detection outcomes. Furthermore, from the literature analysis, we have observed that most papers concentrate on multi-class classification within major categories and do not explore the subcategories of attacks. Moreover, we noted that all the related papers worked with extensive datasets, without considering lightweight data suitable for low-memory devices. This paper addresses these issues by considering a higher number of attack classes using both extensive and lightweight data. Additionally, for performance improvement, this paper introduces a novel DL model called SACNN, designed to address the dataset imbalance issue for a limited and diverse number of attack classes.

TABLE 1 Related work overview of cyberattacks detection in IIoT.

Papers	Years	Method	Dataset	Evaluation metrics	Average finding score (in %)	No. of attacks	Multi-class category	Multi-class sub-category	Light data
[19]	2020	Multi-CNN	NSL-KDD	Accuracy, precision, recall, F1-Score	86.95, 89.56, 87.25, 88.41	4	✓	×	×
[20]	2020	M2-DAE	Bot-IoT	F1-score	99.7	3	✓	×	×
[21]	2021	Forensics-DL	Bot-IIoT, UNSW-NB15	Accuracy, precision, recall, F1-Score	98.93, 97.52, 98.1, 97.82	5, 9	✓	×	×
[22]	2021	GA	UNSW-NB15	Accuracy, precision, recall, F1-Score	77.64, 83.09, 77.64, 80.27	9	✓	×	×
[23]	2022	VAE	CSE-CIC-IDS2018	Accuracy, precision, recall, F1-Score	98.57, 91.33, 82.18, 84.03	6	✓	×	×
[24]	2022	SAE-CNN	ToN-IoT, UNSW-NB15	Precision, recall, F1-Score	93.35, 97.6, 95.2	9, 9	✓	×	×
[25]	2022	GNN	Mississippi	Accuracy, precision, recall, F1-Score	97.2, 98, 90, 93	7	✓	×	×
[26]	2022	DAE	UNSW-NB15	Accuracy, precision, recall, F1-Score	97.95, 98, 96.63, 97.89	9	×	×	×
[27]	2022	XGboost	X-IIoTDS, TON_IoT	Precision, recall, F1-Score	9.91, 99.84, 99.88	9, 7	✓	×	×
[28]	2022	CNN-BiLSTM	NSL-KDD	Accuracy, Detection rate, Precision	96.3, 97.1, 98.9	4	✓	×	×
[29]	2023	CNN-LSTM	UNSW-NB15, X-IIoTID	Accuracy, precision, recall, F1-Score	96, 96.06, 96.09, 96.07	9, 9	✓	×	×
[30]	2023	OCNN-LSTM	ToN_IoT UNSW-NB15	Accuracy, precision, recall, F1-Score	93.56, 93.48, 53.7, 50.86	7, 9	✓	×	×
[31]	2023	Res-TranBiLSTM	NSL-KDD, CIC-IDS2017	Accuracy, precision, recall, F1-Score	95.07, 95.27, 95.04, 94.02	4, 6	✓	×	×
This study	2023	SACNN	Edge-IIoTset, X-IIoTID	Accuracy, precision, recall, F1-Score	99.62, 99.44, 99.11, 99.27	14, 9	✓	✓	✓

3 | THE PROPOSED INTRUSION DETECTION SYSTEM

This section overviews our proposed approach for detecting cyberattacks within IIoT networks. It also describes the preliminary steps required including data preparation, features extraction, normalisation, and data splitting.

3.1 | Datasets description

Edge-IIoTset and X-IIoTID are renowned datasets that are used by several researchers in the field of ML and DL-based IDS. These datasets contain IoT and IIoT traffic samples generated by a real-world testbed deployment consisting of seven interconnected layers, including cloud computing, network functions virtualisation, blockchain network, fog computing, software-defined networking, and edge computing, in addition to IoT and IIoT perception layers. More than ten types of devices were used to generate the data including soil and water monitoring, temperature, and humidity among other IoT devices. Edge-IIoTset contains 14 attacks associated with IoT and IIoT communication protocols that are classified into five classes: DoS/DDoS, Information gathering, Man in the middle, Injection, and Malware attacks. Data were collected from network packets in the form of pcap files which were converted to CSV using Zeek and TShark tools [18]. This dataset is available in two versions: heavy and light. The heavy version contains 2,219,201 instances, while the light version contains 157,800 instances. The X-IIoTID dataset was created by monitoring a real-time IIoT network, which encompassed

the behaviours of contemporary IIoT communication protocols, the operations of state-of-the-art devices, various attack types, and diverse attack scenarios, as well as several attack protocols [32]. The dataset consists of 65 input features and a total of 820,834 instances. Among these instances, 421,417 are categorised as normal, while the remaining 399,417 instances correspond to different attack types. A full breakdown of these datasets is given in Table 2.

3.2 | Preprocessing techniques

This section comprises preprocessing procedures. In this experiment, three preprocessing procedures were used: data preparation, feature extraction, and normalisation.

3.2.1 | Data preparation

This is the first step in the preprocessing stage to address the problem of missing values and convert categorical features into numeric features. There are no null values in the Edge-IIoTset dataset. The utilised dataset contains categorical attributes that include numerous data categories. We considered using a one-hot encoder to map categorical attributes to numeric values; however, this mechanism requires a large amount of memory and introduces significant latency [33]. As a result, we instead used the label encoder mechanism for the conversion task. In this method, each label is assigned a distinct numeric value based on alphabetical order that does not require additional memory.

TABLE 2 A detailed presentation of datasets.

Heavy version of Edge-IIoTset				Light version of Edge-IIoTset				X-IIoTID	
Category	Instances	Sub category	Instances	Category	Instances	Sub category	Instances	Class	Instances
Normal	1,615,643	Normal	1,615,643	Normal	24,301	Normal	24,301	Normal	421,417
DDoS	337,977	UDP	121,568	DDoS	49,396	DDoS_UDP	14,498	RDOS	141,261
		ICMP	116,436			DDoS_ICMP	14,090	Reconnaissance	127,590
		TCP	50,062			DDoS_TCP	10,247		
		HTTP	49,911			DDoS_HTTP	10,561		
Injection	104,752	XSS	15,915	Injection	30,632	XSS	10,052	Weaponization	67,260
		SQL	51,203			SQL	10,311	Exfiltration	22,134
		Uploading	37,634			Uploading	10,269		
Malware	85,940	Password	50,153	Malware	31,109	Password	9989	Lateral	31,596
		Backdoor	24,862			Backdoor	10,195	Movement	5122
		Ransomware	10,925			Ransomware	10,925	Tampering	
Scanning	73,675	Port	22,564	Scanning	21,148	Port	10,071	C&C	2863
		Fingerprinting	1001			Fingerprinting	1001	Cryoti	458
		Vulnerability	50,110			Vulnerability	10,076	Ransomware	
MITM	1214	MITM	1214	MITM	1214	MITM	1214	Exploitation	1133

3.2.2 | Feature extraction

Feature extraction is the process of reducing a high-dimensional dataset into a low-dimensional dataset with the goal of selecting the most significant features. The key benefit of feature extraction is to improve the effectiveness of the classification model by avoiding overfitting and introducing less processing power, and better memory utilisation [34–37]. In our proposed technique, we opted to use the extra-tree classifier (ETC) method to extract the most significant features due to its many advantages over the other extraction methods [38]. The ETC operates on the gain value that represents the impact of an attribute on the output category.

The ETC is an ensemble learning technique that combines the outcomes of numerous decision trees, which are uncorrelated, to form a ‘forest’ to produce its classification results. In this method, each decision tree is built from the original training sample. During testing at each node, a random subset of k features is provided to each tree from the feature set. Consequently, each decision tree independently selects the optimal feature for data splitting using Equation (1).

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \quad (1)$$

Where $\text{Gain}(S, A)$ is the information gain which splits the dataset S based on attribute A . $\text{Values}(A)$ are distinct values that the attribute A can take. S_v is the Subset of examples in S for which attribute A has value v . $|S_v|$ represents the number of examples in subset S_v . And $|S|$ denotes the total number of examples in the dataset. The final information gain for each feature is computed as an average of overall decision trees in the ETC. The goal is to identify features that consistently provide high information gain across multiple trees. Attributes with a total information gain value greater than 0 were extracted, while 0 information gain value attributes had no impact on the output. After the filtering process, 54 attributes were extracted with greater than 0 gain value. The remaining eliminated 7 attributes have 0 gain value.

3.2.3 | Normalisation and splittings

Normalisation is a method of rescaling data into a common range. For ML and DL classifiers, there is no need to rescale the dataset if the scales of the attributes are not much variant. Different range-scaled attributes of the dataset affect the effectiveness of the classifiers [39, 40]. The edge-IIoT set dataset has a varied range of attributes that need to be normalised. In our proposed approach, the min–max normalisation technique is used to scale the attributes between 0 and 1, as represented in Equation (2). To verify the efficacy of the SACNN, we divided the normalised form of the dataset into two portions, one containing 80% of the data for training and

the other 20% for testing purposes. The stratified technique is applied to divide the data into identical sets for each class.

$$X_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (2)$$

3.3 | The proposed SACNN architecture

The proposed SACNN architecture consists of a self-attention layer and convolutional neural networks (CNN) layers, as shown in Figure 1. The self-attention layer divides the input features into vectors of equal size called heads. It processes all heads in parallel and computes the attention value for each head. CNN layers are used to process the computed attention and predict the attack class, as Algorithm 1 outlines the details using a pseudo-formatted flow. The proposed SACNN approach operates input shape (instances_set, attributes, 1), where ‘instances_set’ represents the batch size, ‘attributes’ represents the number of input features, and ‘1’ represents the individual input instance. In the proposed architecture, self-attention splits the input features into eight equal vectors (heads) [41]. The head size is calculated in Equation (3), where H_s is the size of the head, I_s is the total number of input attributes, and N_b is the number of splitting heads. Self-attention layer computes the attention based on queries (Q), keys (k), and values (V). Q, K, and V are demonstrated in Equation (4), Equation (5), and Equation (6), respectively, where X represents the input vector and W represents the weight. The attention of each head is computed using Equation (7), where d_q is the length of Q. All the computed attentions of the heads are combined to generate the output of the self-attention layer, as expressed in Equation (8). Add and norm layer was used to handle the vanishing gradient issue [42].

$$H_s = \lceil \frac{I_s}{N_b} \rceil \quad (3)$$

$$Q = X \times W_Q \quad (4)$$

$$K = X \times W_K \quad (5)$$

$$V = X \times W_V \quad (6)$$

$$Z_i = \text{softmax} \left(\frac{Q \times K^T}{\sqrt{d_q}} \right) \times V \quad (7)$$

$$Z = Z_i(1, \dots, n) \quad (8)$$

Algorithm 1 Proposed SACNN algorithm

Require: Input data vector X

Ensure: Output predicted probabilities Y

```

1: function SELFATTENTION( $X$ )
2:    $Q, K, V \leftarrow \text{LinearTransform}(X)$ 
    $\triangleright$  Linear transformations
3:    $H \leftarrow \text{Attention\_score}(Q, K, V)$ 
    $\triangleright$  Self attention
4:   return  $H$ 
5: end function
6: function CNNLAYERS( $H_{\text{norm}}$ )

```

```

7:    $H_{Convo} \leftarrow \text{Convo}(H_{Norm})$ 
   ▷ Convolutional layers
8:    $H_{CNN} \leftarrow \text{MaxPool}(H_{Convo})$            ▷ Max
   Pooling layer
9:   return  $H_{CNN}$ 
10: end function
11: function DENSELAYERS( $H_{CNN}$ )
12:    $D_{dense} \leftarrow \text{ReLU}(H_{CNN})$            ▷ Dense
   ReLU layer
13:    $Y \leftarrow \text{Softmax}(D_{dense})$            ▷ Softmax
   layer for predictions
14:   return  $Y$ 
15: end function
16:  $H \leftarrow \text{SELFATTENTION}(X)$ 
17:  $H_{Norm} \leftarrow \text{Add\_}\&\_ \text{Norm}(H + X)$            ▷ Add
   and layers normalisation
18:  $H_{CNN} \leftarrow \text{CNNLAYERS}(H_{Norm})$ 
19:  $Y \leftarrow \text{DENSELAYERS}(H_{CNN})$ 
20: return  $Y$ 

```

The self-attention layer computed the significance of heads, where each head consists of multiple features. In the next stage, we pass the output of the self-attention layer to the CNN layers for network intrusion detection. The fundamental benefit of a CNN is its ability to capture the importance of features. Moreover, a CNN operates with fewer parameters than other DL algorithms, resulting in faster performance [43]. A CNN is typically made up of convolutional layers, pooling layers, and fully connected layers [44]. In the proposed architecture, two 1D convolutional layers with a kernel size of three and 26 filters, a max-pooling layer with a pool size of four, flatten, and two fully connected layers were utilised. The convolutional layer highlights the significance of the features as well as diminishes the noise [45]. The utilised convolutional layers are expressed in Equation (9) and Equation (10), where the input to the CNN is represented by x_k , s_i signifies the neurons of the previous layer, w_{ik} signifies the kernel size, and b_k depicts the bias. The output of the convolution operation is denoted by y_k where the ReLU activation method is employed. The yield of the convolution operations passes into a

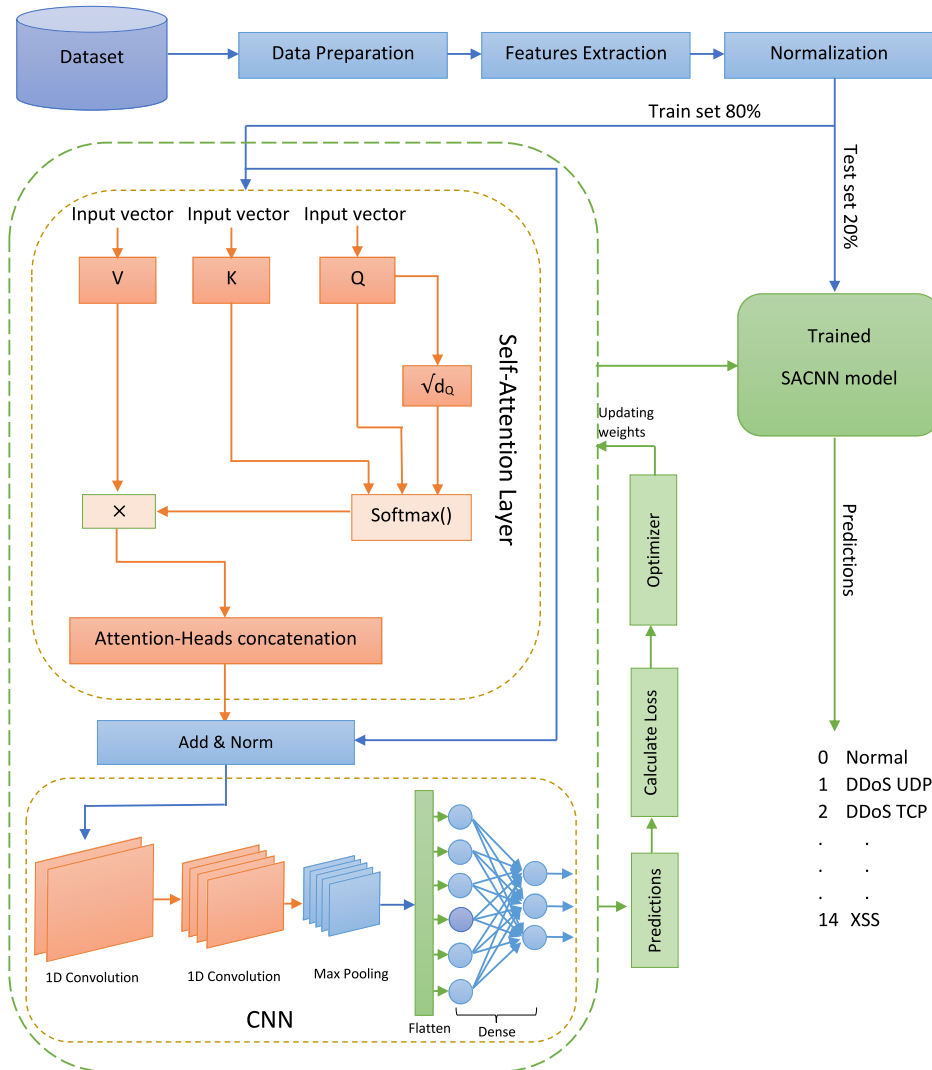


FIGURE 1 Flow diagram of the proposed architecture.

max-pooling operation that converges to the most prominent features as expressed in Equation (11). The flatten mechanism is utilised to transform the yield of max-pooling operations into a 1D vector that passes into ReLU dense layer. The final layer of the approach is the softmax dense layer that produces the output, as demonstrated in Equation (12).

$$x_k = b_k + \sum_{i=1}^N (s_i, w_{ik}) \quad (9)$$

$$y_k = \max(0, x_k) \quad (10)$$

$$s_k = \max_{i \in \mathcal{R}} y_k \quad (11)$$

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (12)$$

3.3.1 | Hyperparameters

Hyperparameters are established prior to the training of neural networks, enabling the DL models to learn from the training data. These hyperparameters play a crucial role in the training process and can significantly impact the performance of the model. Table 3 presents the most important hyperparameters utilised in the proposed SACNN model.

3.4 | Experimental setup

The experiments of our proposed approach were carried out on a Jupyter notebook with an Intel Core i5 8th generation processor and 24-GB RAM running Windows 11 Pro 64-bit operating system and Python 3.8. Several Python libraries were utilised including Keras, TensorFlow, Pandas, Scikit Learn, and NumPy.

4 | PERFORMANCE ASSESSMENT

This section covers a comprehensive assessment of the proposed approach. Edge-IIoTset datasets were used in this experiment to assess the effectiveness of the proposed approach. Experiments were conducted for multi-class categories and multi-class sub-categories, and outcomes of the SACNN approach are assessed in comparison to other ML and DL methods.

TABLE 3 Utilised hyperparameters in the proposed SACNN.

Optimiser	Learning		Batch size	Epochs
	rate	Loss function		
Adam	0.001	Sparse categorical cross-entropy	32	6

4.1 | Evaluation metrics

Several metrics are used in this study to assess the efficacy of our proposed classification approach including accuracy, macro-precision, macro-recall, and macro F1-score.

- Accuracy (ACC) refers to the percentage of correctly predicted instances made by the classification model out of all the predictions made and it is calculated as in Equation (13). Where α , β , γ , and δ represent true positive, true negative, false positive, and false negative, respectively.

$$\text{ACC} = \frac{\alpha + \beta}{\alpha + \beta + \gamma + \delta} \quad (13)$$

- Precision (P) refers to the ratio between the True Positives and all the instances classified as positive. Our model refers to the number of true instances classified as abnormal out of all instances classified as abnormal by the model as given in Equation (14).

$$P = \frac{\alpha}{\alpha + \gamma} \quad (14)$$

- Recall (R) refers to the ratio between the True Positives and all the positive instances. In our model, it refers to the number of true instances classified as abnormal out of all abnormal instances as given in Equation (15).

$$R = \frac{\alpha}{\alpha + \delta} \quad (15)$$

It is worth indicating that normal samples are designated as Negative, whereas abnormal samples are designated as Positive in this study. There are situations when the precision and recall measures conflict and thus, they should be carefully investigated. Several researchers used the F1 score which is the harmonic mean of precision and recall, as shown in Equation (16).

$$\text{F1 Score} = \frac{2 \times (P \times R)}{P + R} \quad (16)$$

The area under the curve (AUC) is a metric that quantifies the area beneath the receiver operating characteristic (ROC) curve. The ROC curve is generated by graphing the true positive rate (TPR) against the false positive rate (FPR) across various classification thresholds. TPR and FPR are determined using Equation (17) and Equation (18), respectively.

$$\text{TPR} = \frac{\alpha}{\alpha + \delta} \quad (17)$$

$$\text{FPR} = \frac{\gamma}{\gamma + \beta} \quad (18)$$

4.2 | Results and discussion

The Adam optimiser, loss function sparse categorical cross-entropy, and batch size of 32 were used in this study. The model was trained over a period of six epochs. For training and testing the proposed architecture, a fivefold cross-validation approach is also used.

4.2.1 | Results on multi-class categories of Edge-IIoTset

Table 4 shows the performance evaluation results of the proposed SACNN approach on the Edge-IIoTset dataset multi-class categories for both the heavy and light versions with varying numbers of CNN layers. It is evident from the table that the optimum intrusion detection performance of the SACNN approach was achieved with two convolutional and one max pooling layer on both heavy and light versions of the dataset. All of the evaluation results were optimal when these layers were used.

4.2.2 | Results on multi-class sub-categories of Edge-IIoTset

Table 5 shows the performance evaluation results of the proposed SACNN approach on the Edge-IIoTset dataset multi-class sub-categories for both the heavy and light versions and again with varying numbers of CNN layers. Similar to the results on the multi-class categories, it is also clear that the optimal performance of the proposed approach was achieved under two convolutional layers and one max pooling layer. All the evaluation results were optimal for these CNN layers.

TABLE 4 Results of the proposed SACNN on Edge-IIoTset multi-class categories.

Layers		Heavy version of Edge-IIoTset					Light version of Edge-IIoTset				
Convolutional	Max pooling	P	R	F1-score	ACC	AUC	P	R	F1-score	ACC	AUC
1	1	0.9974	0.9976	0.9975	0.9994	0.9998	0.991	0.994	0.9923	0.9927	0.9998
2	1	0.9979	0.998	0.9979	0.9995	0.9999	0.9945	0.9955	0.995	0.9949	0.9999
2	2	0.9948	0.9947	0.9947	0.9987	0.9991	0.9891	0.9895	0.9893	0.9892	0.9987
4	2	0.9961	0.9965	0.9963	0.9991	0.9995	0.9919	0.9904	0.9911	0.9914	0.9989

TABLE 5 Results of the proposed SACNN on Edge-IIoTset multi-class sub-categories.

Layers		Heavy version of Edge-IIoTset					Light version of Edge-IIoTset				
Convolutional	Max pooling	P	R	F1-score	ACC	AUC	P	R	F1-score	ACC	AUC
1	1	0.9948	0.9889	0.9916	0.9994	0.9995	0.9906	0.9818	0.9856	0.9897	0.9993
2	1	0.9966	0.9911	0.9937	0.9995	0.9999	0.9921	0.9876	0.9897	0.9927	0.9998
2	2	0.9934	0.9867	0.9898	0.9993	0.9991	0.9802	0.9733	0.976	0.982	0.9995
4	2	0.9936	0.9846	0.9888	0.9992	0.9987	0.9792	0.9896	0.9837	0.9914	0.9997

4.2.3 | Results on X-IIoTID

Table 6 presents the performance evaluation results of the proposed SACNN approach on the X-IIoTID dataset for multi-class classification, with different numbers of CNN layers. Similar to the findings in the Edge-IIoTset results, it is evident that the optimal performance of the proposed approach was achieved with two convolutional layers and one max pooling layer. All the evaluation results were optimal for these CNN layers.

4.2.4 | Discussion

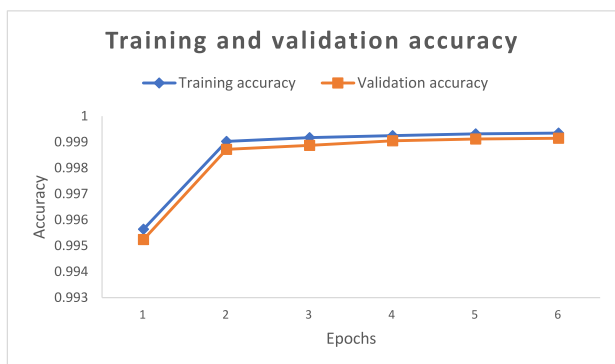
The optimal results for the utilised datasets on multi-class categories and multi-class sub-categories were achieved with two 1D convolutional layers and a max pooling layer. It is important to note that these datasets had imbalanced class distributions. The average accuracy of the proposed DL approach reached 99.66% for multi-class categories and multi-class sub-categories on both the heavy and light versions of the Edge-IIoTset dataset. Moreover, it achieved an accuracy of 99.72% on the X-IIoTID dataset. To ensure that the proposed

TABLE 6 Results of the proposed SACNN on X-IIoTID multi-class.

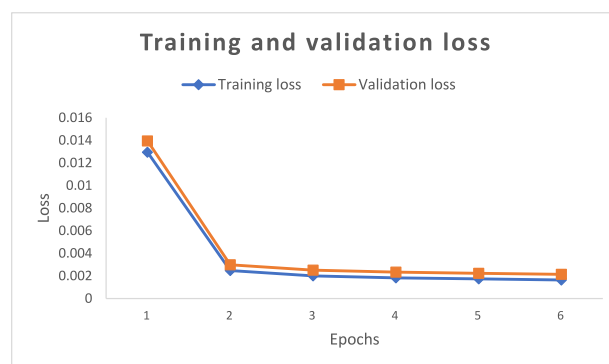
Layers		P	R	F1-score	ACC	AUC
Convolutional	Max pooling					
1	1	0.9788	0.9697	0.9728	0.9917	0.9978
2	1	0.9911	0.9835	0.9871	0.9981	0.9995
2	2	0.9425	0.9534	0.9457	0.9899	0.9982
4	2	0.9922	0.9378	0.9601	0.9972	0.9987

model did not suffer from overfitting or underfitting issues, we analysed its training and validation performance. Figures 2–5 shows the training and validation performance for the selected optimal layers of the SACNN model when applied to both the heavy and light versions of the Edge-IIoTset dataset.

Meanwhile, Figure 6 illustrates the training and validation performance on the X-IIoTID dataset. Notably, the training and validation performance demonstrate consistent results, providing evidence that the proposed SACNN model does not suffer from overfitting or underfitting issues.

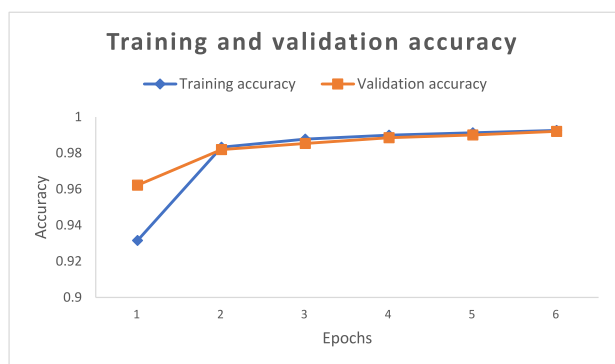


(a) Accuracy

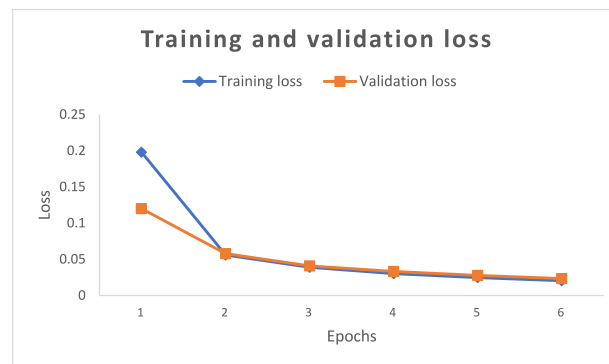


(b) Loss

FIGURE 2 Training and validation performance of the proposed self-attention convolutional neural network on the heavy version of Edge-IIoTset multi-class categories.

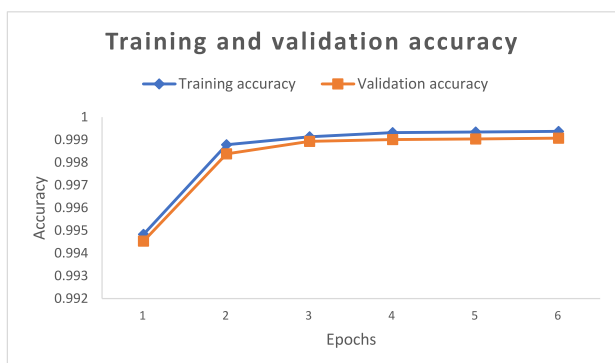


(a) Accuracy

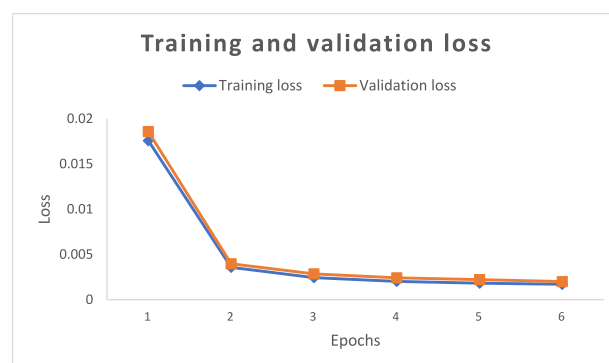


(b) Loss

FIGURE 3 Training and validation performance of the proposed self-attention convolutional neural network on the light version of Edge-IIoTset multi-class categories.



(a) Accuracy



(b) Loss

FIGURE 4 Training and validation performance of the proposed self-attention convolutional neural network on the heavy version of Edge-IIoTset multi-class sub-categories.

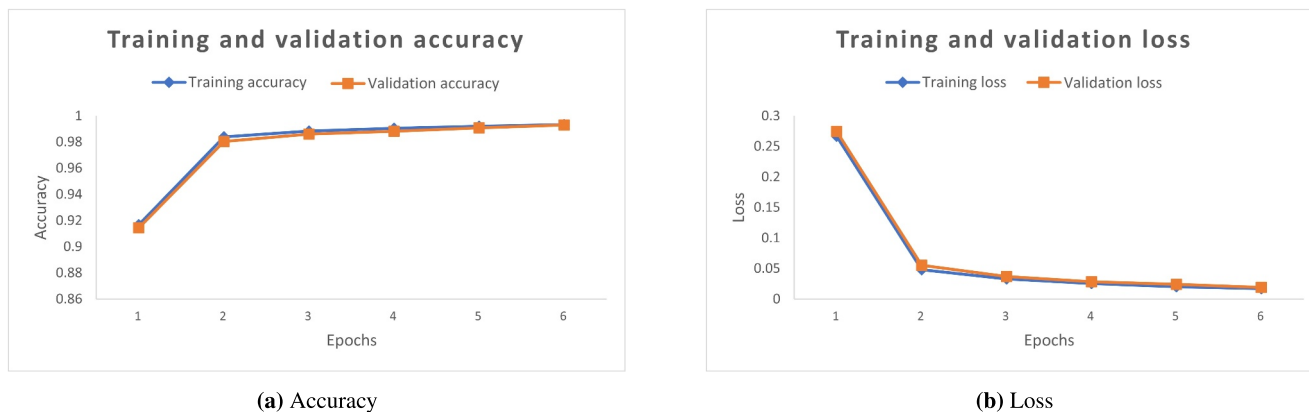


FIGURE 5 Training and validation performance of the proposed self-attention convolutional neural network on the light version of Edge-IIoTset multi-class sub-categories.



FIGURE 6 Training and validation performance of the proposed self-attention convolutional neural network on X-IIoTID dataset.

4.3 | Comparison with existing Machine Learning and DL methods

We also compared the performance of our proposed approach to that of several other ML and DL classifiers including CNN, gated recurrent units (GRU), LSTM, autoencoder (AE), deep neural network (DNN) multi-layer perceptron (MLP), linear regression (LR), Naive Bayes (NB), and support vector machine (SVM) on the same dataset and using the same experimental settings. In this research, we applied the same preprocessing steps to prepare data for all DL and ML-based IDS as the proposed SACNN. This consistent approach is crucial for making a fair comparison with other ML and DL techniques. If different data preparation methods were used for each model, it would be challenging to determine whether any differences in performance were due to the model's design or the data preprocessing. For all DL-based models, we employed the Adam optimisation function with sparse categorical cross-entropy loss and trained them for six epochs with a batch size of 32. The implementation of DL-based models, such as GRU, LSTM, CNN, AE, and DNN, includes 2, 2, 4, 6, and 4 hidden layers, respectively. In the CNN, the convolutional layers consist of 32 filters with a kernel size of 3 and use the same padding, while the max pooling has a pool size of 4.

Tables 7 and 8 provide a comprehensive summary of the evaluation results for the respective models on the multi-class categories of the Edge-IIoTID dataset, both for the heavy and light versions. These results clearly demonstrate that the proposed SACNN model consistently outperforms all other models in terms of accuracy, precision, recall, and F1-score across both versions of the Edge-IIoTset dataset in the multi-class category. Similarly, Tables 9 and 10 present the evaluation results for the respective models on the multi-class sub-categories of the Edge-IIoTset dataset, considering both the heavy and light versions. Once again, the results clearly indicate that the proposed model surpasses its counterparts in terms of accuracy, precision, recall, and F1-score for both versions of the Edge-IIoTset dataset in the multi-class sub-category. In addition, Table 11 displays the evaluation results for the same models applied to the X-IIoTID dataset. As expected, the results in this table also highlight the superiority of the proposed model, showcasing higher accuracy, precision, recall, and F1-score when compared to other models for the X-IIoTID dataset.

It is worth noting that the superiority of our proposed model over other models was obtained using an imbalanced dataset with a smaller number of instances. It is also noticeable that while other models did not perform well using the light

TABLE 7 Results comparison of the proposed SACNN with other models on the heavy version of Edge-IIoTset multi-class categories.

Algorithm	P	R	F1-score	ACC	AUC	Training time (in sec)	Test time (in sec)
LSTM [29]	0.9921	0.9957	0.9938	0.9983	0.9991	2689	171
GRU [46]	0.987	0.9887	0.9878	0.9983	0.9993	2514	202
CNN [29]	0.995	0.9938	0.9944	0.9986	0.9994	83	12
AE [47]	0.9952	0.9981	0.9966	0.9995	0.9998	98	12
DNN [48]	0.9944	0.9838	0.9889	0.9988	0.9996	81	10
MLP [49]	0.9937	0.9952	0.9944	0.9989	0.9997	104	1
LR [48]	0.9161	0.9111	0.9127	0.9859	0.9958	68	1
NB [48]	0.8767	0.8371	0.831	0.9422	0.9775	2	1
SVM [48]	0.9817	0.9855	0.9841	0.9954	0.9976	19,975	677
Proposed SACNN	0.9979	0.998	0.9979	0.9995	0.9999	181	18

TABLE 8 Results comparison of the proposed SACNN with other models on the light version of Edge-IIoTset multi-class categories.

Algorithm	P	R	F1-score	ACC	AUC	Training time (in sec)	Test time (in sec)
LSTM [29]	0.9675	0.9711	0.9692	0.9722	0.9949	162	12
GRU [46]	0.88	0.8584	0.8683	0.8952	0.9684	178	15
CNN [29]	0.9826	0.9827	0.9826	0.9816	0.9986	8	1
AE [47]	0.9926	0.9937	0.9931	0.9933	0.9993	7	1
DNN [48]	0.9915	0.9922	0.9918	0.9923	0.9989	6	1
MLP [49]	0.7932	0.8004	0.7968	0.9576	0.9642	46	0.5
LR [48]	0.9442	0.9468	0.9451	0.9403	0.9952	5	0.2
NB [48]	0.8198	0.7557	0.7454	0.7292	0.9683	1	0.4
SVM [48]	0.9727	0.9724	0.9725	0.9695	0.9962	110	15
Proposed SACNN	0.9945	0.9955	0.995	0.9949	0.9999	13	1

TABLE 9 Results comparison of the proposed SACNN with other models on the heavy version of Edge-IIoTset multi-class sub-categories.

Algorithm	P	R	F1-score	ACC	AUC	Training time (in sec)	Test time (in sec)
LSTM [29]	0.9883	0.9911	0.9896	0.9991	0.9997	3096	211
GRU [46]	0.9414	0.9774	0.9547	0.9977	0.9994	3219	239
CNN [29]	0.9887	0.9836	0.9859	0.9988	0.9996	86	12
AE [47]	0.9829	0.9942	0.9879	0.9984	0.9991	103	12
DNN [48]	0.9944	0.9838	0.9889	0.9988	0.9995	81	10
MLP [49]	0.995	0.9845	0.9893	0.9992	0.9989	137	1
LR [48]	0.9297	0.9023	0.91	0.9962	0.9972	114	1
NB [48]	0.9582	0.9337	0.9404	0.9962	0.996	2	3
SVM [48]	0.9836	0.98	0.9817	0.9985	0.9992	432	241
Proposed SACNN	0.9966	0.9911	0.9937	0.9995	0.9999	187	19

TABLE 10 Results comparison of the proposed SACNN with other models on the light version of Edge-IIoTset multi-class sub-categories.

Algorithm	P	R	F1-score	ACC	AUC	Training time (in sec)	Test time (in sec)
LSTM [29]	0.9473	0.9502	0.9485	0.9676	0.9841	156	11
GRU [46]	0.8424	0.8477	0.8439	0.9337	0.9493	167	13
CNN [29]	0.9824	0.9647	0.9718	0.9835	0.9978	6	1
AE [47]	0.9781	0.9914	0.9837	0.9918	0.9986	7	1
DNN [48]	0.9805	0.9895	0.9844	0.99	0.9989	6	1
MLP [49]	0.9254	0.9103	0.9153	0.9474	0.9924	52	0.1
LR [48]	0.9759	0.9529	0.9614	0.9757	0.9987	9	0.2
NB [48]	0.9071	0.8776	0.8801	0.9184	0.9951	0.5	1
SVM [48]	0.9882	0.9772	0.982	0.9879	0.9983	21	11
Proposed SACNN	0.9921	0.9876	0.9897	0.9927	0.9998	13	2

TABLE 11 Results comparison of the proposed SACNN with other models on X-IIoTID multi-class.

Algorithms	P	R	F1-score	ACC	AUC	Training time (in sec)	Test time (in sec)
LSTM [29]	0.9821	0.9022	0.9344	0.9915	0.9951	710	61
GRU [46]	0.9652	0.945	0.9523	0.9948	0.9984	780	47
CNN [29]	0.9818	0.9624	0.9706	0.9903	0.9992	42	7
AE [47]	0.9887	0.9796	0.9836	0.9976	0.9998	33	6
DNN [48]	0.9804	0.9761	0.9781	0.988	0.9978	28	6
MLP [49]	0.9779	0.9692	0.9723	0.9926	0.9959	235	0.2
LR [48]	0.9526	0.9164	0.9301	0.9764	0.9921	39	0.1
NB [48]	0.5068	0.8145	0.4832	0.4858	0.7861	2	1
SVM [48]	0.9632	0.9671	0.9647	0.9818	0.9937	42,956	374
Proposed SACNN	0.9911	0.9835	0.9871	0.9981	0.9995	259	53

version compared to the heavy version, the proposed model achieved comparable results under both dataset variants. This demonstrates that our proposed model can detect attacks with high accuracy even when trained on a small subset of the dataset.

5 | CONCLUSION

This paper proposes a SACNN architecture for the detection of malicious activity in IIoT networks and an appropriate feature extraction method to extract the most significant feature. The proposed architecture has a self-attention layer to calculate the input attention and CNN layers to process the assigned attention features for prediction. The performance evaluation of the proposed SACNN architecture has been done with the Edge-IIoTset and X-IIoTID datasets. The proposed approach achieved an average accuracy of 99.66% for multi-class categories and multi-class sub-categories on both versions of the Edge-IIoTID dataset. Moreover, it achieved an accuracy of 99.72% on the X-IIoTID dataset. The SACNN model successfully

addressed the issue of imbalance and fewer data and improved the intrusion detection performance in IIoT networks. Compare the performance of the proposed approach with other classifiers to validate its efficacy. The proposed approach has higher efficiency than other classifiers for multi-class categories and multi-class sub-categories on both datasets.

The proposed model improves the performance of cyberattack detection in IIoT networks. Additionally, the proposed SACNN demonstrates effective functionality on low-power and low-memory devices, with an average processing time compared to other models. While some other models may offer quicker training and testing times, they fail to match the detection performance of the proposed model. Furthermore, the proposed model has the potential for additional compression and optimisation, leading to reduced detection time and enhanced performance.

ACKNOWLEDGEMENTS

Authors would like to acknowledge the support of the Deputy for Research and Innovation - Ministry of Education, Kingdom of Saudi Arabia for this research through grant (NU/IFC/02/

SERC/-/31) under the Institutional Funding Committee at Najran University, Kingdom of Saudi Arabia.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The dataset used in this research was obtained from publicly available sources: [Edge-IIoTset: <https://iee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iiot-and-iiot-applications> and <https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iiot-iiot>, X-IIoTID: <https://iee-dataport.org/documents/x-iiotid-connectivity-and-device-agnostic-intrusion-dataset-industrial-internet-things>].

ORCID

Jawad Ahmad  <https://orcid.org/0000-0001-7495-2248>

REFERENCES

- Kaur, B., et al.: Internet of things (iot) security dataset evolution: challenges and future directions. *IoT* 22, 100780 (2023). <https://doi.org/10.1016/j.iot.2023.100780>
- Anand, A., Singh, A.: A hybrid optimization-based medical data hiding scheme for industrial internet of things security. *IEEE Trans. Ind. Inf.* 19(1), 1051–1058 (2022). <https://doi.org/10.1109/tii.2022.3164732>
- Haq, M.I.U., et al.: Robust graph-based localization for industrial internet of things in the presence of flipping ambiguities. *CAAI Transactions on Intelligence Technology* 8(4), 1140–1149 (2023). <https://doi.org/10.1049/cit2.12203>
- Tang, S., et al.: Computational intelligence and deep learning for next-generation edge-enabled industrial iot. *IEEE Transactions on Network Science and Engineering* 10(5), 2881–2893 (2022). <https://doi.org/10.1109/tNSE.2022.3180632>
- Abdel-Basset, M., et al.: Deep-ifs: intrusion detection approach for industrial internet of things traffic in fog environment. *IEEE Trans. Ind. Inf.* 17(11), 7704–7715 (2020). <https://doi.org/10.1109/tii.2020.3025755>
- Shi, Y., et al.: Joint online optimization of data sampling rate and pre-processing mode for edge-cloud collaboration enabled industrial iot. *IEEE Internet Things J.* 9(17), 16402–16417 (2022). <https://doi.org/10.1109/jiot.2022.3150386>
- Hassan, M.M., et al.: Increasing the trustworthiness in the industrial iot networks through a reliable cyberattack detection model. *IEEE Trans. Ind. Inf.* 16(9), 6154–6162 (2020). <https://doi.org/10.1109/tii.2020.2970074>
- Bovenzi, G., et al.: Network anomaly detection methods in iot environments via deep learning: a fair comparison of performance and robustness. *Comput. Secur.* 128, 103167 (2023). <https://doi.org/10.1016/j.cose.2023.103167>
- Rejeb, A., et al.: The internet of things (iot) in healthcare: taking stock and moving forward. *IoT* 22, 100721 (2023). <https://doi.org/10.1016/j.iot.2023.100721>
- Wani, A., Khaliq, R.: Sdn-based intrusion detection system for iot using deep learning classifier (idsiot-sdl). *CAAI Transactions on Intelligence Technology* 6(3), 281–290 (2021). <https://doi.org/10.1049/cit2.12003>
- Liang, W., et al.: Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial iot. *IEEE Trans. Ind. Inf.* 18(8), 5087–5095 (2021). <https://doi.org/10.1109/tii.2021.3116085>
- Samanta, R.K., et al.: Scope of machine learning applications for addressing the challenges in next-generation wireless networks. *CAAI Transactions on Intelligence Technology* 7(3), 395–418 (2022). <https://doi.org/10.1049/cit2.12114>
- Xingxin, C., Xin, Z., Gangming, W.: Research on online fault detection tool of substation equipment based on artificial intelligence. *J. King Saud Univ. Sci.* 34(6), 102149 (2022). <https://doi.org/10.1016/j.jksus.2022.102149>
- Mirsky, Y., et al.: Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection (2018). arXiv preprint arXiv:1802.09089
- Gao, Z.J., Pansare, N., Jermaine, C.: Declarative parameterizations of user-defined functions for large-scale machine learning and optimization. *IEEE Trans. Knowl. Data Eng.* 31(11), 2079–2092 (2018). <https://doi.org/10.1109/tkde.2018.2873325>
- Al-Turaiki, I., Altwaijry, N.: A convolutional neural network for improved anomaly-based network intrusion detection. *Big Data* 9(3), 233–252 (2021). <https://doi.org/10.1089/big.2020.0263>
- Aldweesh, A., Derhab, A., Emam, A.Z.: Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. *Knowl. Base Syst.* 189, 105124 (2020). <https://doi.org/10.1016/j.knsys.2019.105124>
- Ferrag, M.A., et al.: Edge-iiotset: a new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access* 10, 40281–40306 (2022). <https://doi.org/10.1109/access.2022.3165809>
- Li, Y., et al.: Robust detection for network intrusion of industrial iot based on multi-cnn fusion. *Measurement* 154, 107450 (2020). <https://doi.org/10.1016/j.measurement.2019.107450>
- Bovenzi, G., et al.: A hierarchical hybrid intrusion detection approach in iot scenarios. In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–7. IEEE (2020)
- Abdel-Basset, M., et al.: Deep-ifs: intrusion detection approach for industrial internet of things traffic in fog environment. *IEEE Trans. Ind. Inf.* 17(11), 7704–7715 (2021). <https://doi.org/10.1109/tii.2020.3025755>
- Kasongo, S.M.: An advanced intrusion detection system for iiot based on ga and tree based algorithms. *IEEE Access* 9, 113199–113212 (2021). <https://doi.org/10.1109/access.2021.3104113>
- Liu, C., et al.: Intrusion detection system after data augmentation schemes based on the vae and cvae. *IEEE Trans. Reliab.* 71(2), 1000–1010 (2022). <https://doi.org/10.1109/tr.2022.3164877>
- Telikani, A., et al.: Industrial iot intrusion detection via evolutionary cost-sensitive learning and fog computing. *IEEE Internet Things J.* 9(22), 23260–23271 (2022). <https://doi.org/10.1109/jiot.2022.3188224>
- Zhang, Y., et al.: Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks. *IEEE Transactions on Network Science and Engineering* 10(5), 2894–2905 (2022). <https://doi.org/10.1109/tNSE.2022.3184975>
- Khan, I.A., et al.: Enhancing iiot networks protection: a robust security model for attack detection in internet industrial control systems. *Ad Hoc Netw.* 134, 102930 (2022). <https://doi.org/10.1016/j.adhoc.2022.102930>
- Le, T.-T.-H., Oktian, Y.E., Kim, H.: Xgboost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability* 14(14), 8707 (2022). <https://doi.org/10.3390/su14148707>
- Li, A., Yi, S.: Intelligent intrusion detection method of industrial internet of things based on cnn-bilstm. *Secur. Commun. Network.* 2022, 1–8 (2022). <https://doi.org/10.1155/2022/5448647>
- Altunay, H.C., Albayrak, Z.: A hybrid cnn+ lstm-based intrusion detection system for industrial iot networks. *Engineering Science and Technology, an International Journal* 38, 101322 (2023). <https://doi.org/10.1016/j.jestech.2022.101322>
- Lilhore, U.K., et al.: Hidm: hybrid intrusion detection model for industry 4.0 networks using an optimized cnn-lstm with transfer learning. *Sensors* 23(18), 7856 (2023). <https://doi.org/10.3390/s23187856>
- Wang, S., Xu, W., Liu, Y.: Res-tranbilstm: an intelligent approach for intrusion detection in the internet of things. *Comput. Network.* 235, 109982 (2023). <https://doi.org/10.1016/j.comnet.2023.109982>
- Al-Hawawreh, M., Sitnikova, E., Aboutorab, N.: X-iiotid: a connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. *IEEE Internet Things J.* 9(5), 3962–3977 (2021). <https://doi.org/10.1109/jiot.2021.3102056>

33. Dahouda, M.K., Joe, I.: A deep-learned embedding technique for categorical features encoding. *IEEE Access* 9, 114381–114391 (2021). <https://doi.org/10.1109/access.2021.3104357>
34. Sarhan, M., et al.: Feature extraction for machine learning-based intrusion detection in iot networks. *Digital Communications and Networks* 10(1), 205–216 (2022). <https://doi.org/10.1016/j.dcan.2022.08.012>
35. Uddin, M.F., et al.: Proposing enhanced feature engineering and a selection model for machine learning processes. *Appl. Sci.* 8(4), 646 (2018). <https://doi.org/10.3390/app8040646>
36. Globerson, A., Tishby, N.: Sufficient dimensionality reduction. *J. Mach. Learn. Res.* 3(Mar), 1307–1331 (2003)
37. Adeel, A., et al.: Entropy-controlled deep features selection framework for grape leaf diseases recognition. *Expet Syst.* 39(7), e12569 (2022). <https://doi.org/10.1111/exsy.12569>
38. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Mach. Learn.* 63(1), 3–42 (2006). <https://doi.org/10.1007/s10994-006-6226-1>
39. Chiba, Z., et al.: A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection. *Comput. Secur.* 75, 36–58 (2018). <https://doi.org/10.1016/j.cose.2018.01.023>
40. Larriva-Novo, X., et al.: An iot-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. *Sensors* 21(2), 656 (2021). <https://doi.org/10.3390/s21020656>
41. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30 (2017)
42. He, K., et al.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
43. Scarpa, G., et al.: A cnn-based fusion method for feature extraction from sentinel data. *Rem. Sens.* 10(2), 236 (2018). <https://doi.org/10.3390/rs10020236>
44. Riyaz, B., Ganapathy, S.: A deep learning approach for effective intrusion detection in wireless networks using cnn. *Soft Comput.* 24(22), 17265–17278 (2020). <https://doi.org/10.1007/s00500-020-05017-0>
45. Zhang, H., et al.: An effective convolutional neural network based on smote and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Network.* 177, 107315 (2020). <https://doi.org/10.1016/j.comnet.2020.107315>
46. Ansari, M.S., Bartoš, V., Lee, B.: Gru-based deep learning approach for network intrusion alert prediction. *Future Generat. Comput. Syst.* 128, 235–247 (2022). <https://doi.org/10.1016/j.future.2021.09.040>
47. Aygun, R.C., Yavuz, A.G.: Network anomaly detection with stochastically improved autoencoder based models. In: *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 193–198. IEEE (2017)
48. Vinayakumar, R., et al.: Deep learning approach for intelligent intrusion detection system. *IEEE Access* 7, 41525–41550 (2019). <https://doi.org/10.1109/access.2019.2895334>
49. Rosay, A., Carlier, F., Leroux, P.: Mlp4nids: an efficient mlp-based network intrusion detection for cids2017 dataset. In: *Machine Learning for Networking: Second IFIP TC 6 International Conference, MLN 2019, Paris, France, December 3–5, 2019, Revised Selected Papers 2*, pp. 240–254. Springer (2020)

How to cite this article: Qathraday, M. A., et al.: SACNN-IDS: a self-attention convolutional neural network for intrusion detection in industrial internet of things. *CAAI Trans. Intell. Technol.* 9(6), 1398–1411 (2024). <https://doi.org/10.1049/cit2.12352>