Neural Computing and Applications (2024) 36:14643–14659
https://doi.org/10.1007/s00521-024-09857-x

ORIGINAL ARTICLE

Enhancing intrusion detection in IIoT: optimized CNN model with multi-class SMOTE balancing

Abdulrahman Mahmoud Eid[1] · Bassel Soudan[1] · Ali Bou Nassif[1] · MohammadNoor Injadat[2]

Received: 12 December 2023 / Accepted: 12 April 2024 / Published online: 9 May 2024

Abstract
This work introduces an intrusion detection system (IDS) tailored for industrial internet of things (IIoT) environments based on an optimized convolutional neural network (CNN) model. The model is trained on a dataset that was balanced using a novel multi-class implementation of synthetic minority over-sampling technique (SMOTE) that ensures equal representation of all classes. Additionally, systematic optimization will be used to fine tune the hyperparameters of the CNN model and mitigate the effects of the increased size of the training dataset. Evaluation results will demonstrate substantial improvement in performance when the optimized CNN model is trained on the balanced dataset. The proposed IDS will be evaluated using the IIoT-specific WUSTL-IIOT-2021 dataset, and then its generalization capability will be verified using the non-domain specific UNSW_NB15 dataset. The model's performance will be evaluated using accuracy, precision, recall, and $F1$-score metrics. The results will demonstrate that the proposed IDS is highly effective with performance exceeding 99.9% on all performance metrics. The IDS is also highly effective in detecting intrusion for generic IT networks achieving improvements in excess of 30% compared to the default baseline model. The results emphasize the versatility and effectiveness of the proposed IDS model, making it a reliable and adaptable solution for enhancing network security across diverse network environments.

Keywords Intrusion detection system · IIoT · CNN · SMOTE · Hyperparameter tuning · WUSTL-IIOT-2021 · UNSW_NB15

1 Introduction

The industrial internet of things (IIoT) refers to the integration of industrial physical assets with sensors and actuators that can be monitored and controlled over the internet. This integration allows the collection, processing, and analysis of extensive real-time data about the operation of the industrial process. The data collected through this process can lead to the development of actionable insight as well as predictions about the possible future behavior of the process and its underlying machinery. Today, IIoT has become significantly important for the automation, productivity, and enhanced efficiency of industrial processes [1]. IIoT enables improved monitoring of processes, planning predictive maintenance, and data-driven decision-making through the analysis of the real-time status of the industrial process, leading to cost savings and operational improvements.

IIoT requires the collection of significant amounts of detailed real-time data that is then exchanged with industrial control systems (ICS) as well as computing systems for analysis and development of the expected insight. In addition, IIoT enables the feedback process where commands are sent directly to the industrial apparatus as a result of the analysis and predictions. This exchange of

✉ Bassel Soudan
bsoudan@sharjah.ac.ae

Abdulrahman Mahmoud Eid
U20105546@sharjah.ac.ae

Ali Bou Nassif
anassif@sharjah.ac.ae

MohammadNoor Injadat
minjadat@zu.edu.jo

[1] Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, Sharjah, UAE

[2] Department of Data Science and AI, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

data and commands is carried over communication networks controlled by either standard internet protocols, or specialized protocols designed specifically for the IIoT system [2]. Regardless, significant amounts of information that affects the industrial process will be carried over the communication network. This has attracted significant attention from malicious actors aiming to exploit the possible vulnerabilities in these networks through cyberattacks.

Cyberattacks on IIoT networks can lead to severe consequences, such as production downtime, safety hazards, and financial losses, making it essential to develop effective security mechanisms for securing these networks [3, 4]. Traditional security tools like firewalls and encryption cannot be directly applied to IIoT due to the distinct nature and constraints of IoT/IIoT devices. Intrusion detection systems (IDSs) have emerged as a critical defense mechanism, constantly monitoring traffic on the network of the IIoT system with the goal of identifying abnormal patterns or potential security breaches.

1.1 The need for an IIoT-specific intrusion detection system

The IIoT architecture presents unique security challenges that distinguish it significantly from conventional networks [5]. IIoT environments are characterized by the sheer volume of data generated by the enormous number of sensors [6]. These sensors are interfaced through a vast array of less-capable IoT devices that are riddled with vulnerabilities [7]. This combination expands the attack surface considerably and exposes the system to novel threats and intrusions, especially taking into consideration the heterogeneous nature of the devices within the IIoT system [8]. Moreover, the use of multiple connectivity protocols, such as TCP/IP, MQTT, Modbus TCP, Cellular, and LoRaWAN, adds layers of complexity to securing these networks [9]. In addition to the privacy and security threats, IIoT systems are also susceptible to zero-day vulnerabilities, a risk less prevalent in traditional networks [10]. These differences between the characteristics of conventional IT and IIoT systems are summarized in Table 1.

In general, IIoT systems consist of resource-constrained devices that have limited computation power and storage capacity, connected through lightweight communication protocols. Conventional cybersecurity methods, including standard IDSs, are insufficient to handle the heterogeneity, unique vulnerabilities, and the complexity of devices and protocols inherent in IIoT networks [11]. This necessitates a separate and specialized approach to security and intrusion detection directed specifically at IIoT systems.

1.2 Gaps and challenges in current IIoT intrusion detection systems

There has been some work in the literature related to the development of IDS systems dedicated specifically for detecting network-based attacks in IIoT systems [12, 13]. A detailed analysis of this literature is presented in another work, with the results presented in Tables 2, 3, and 4 [14]. The tables provide a comprehensive assessment of the literature, considering various aspects such as the specific security concerns addressed by the presented IDS systems, the machine learning (ML) models employed, and the datasets utilized for evaluation purposes. In summary, the literature has shown that there is significant interest in utilizing ML algorithms for classifying the network traffic and identifying normal patterns versus malicious behaviors [15–22]. The literature analysis has also shown that most of these works have unfortunately utilized datasets of little relevance to the behavior of IIoT systems and the type of traffic carried over IIoT networks for training and evaluating the ML models. A number of the works have used the non-domain-specific NSL-KDD and UNSW-NB15 datasets [17, 18], while only a few used the more domain-specific WUSTLIIOT-2018 [15], and WUSTL-IIOT-2021 [20–22]. Two of the works depend on private datasets that are not publicly accessible for reproducibility of the results [16, 19].

It can be seen that collectively, the solutions presented in the literature have the following gaps:

1. Lack of comprehensive security models: Table 2 reveals a notable gap in IDS models that address all five major security concerns in IIoT systems. Existing models predominantly focus on addressing specific security threats individually, rendering the implementation of multiple IDS necessary to ensure comprehensive security. This approach proves impractical for real-life scenarios and necessitates the development of integrated solutions that encompass all security concerns.
2. Limited dataset relevance: Table 3 shows that many of the works in the literature rely on outdated, non-domain-specific, private, or simulated datasets for training and evaluation purposes. While these models often demonstrate high accuracy within the limitations of their datasets, their effectiveness in detecting novel cyberattack scenarios remains questionable.
3. Underutilization of CNNs: Table 4 shows that CNN models are underexplored and underutilized for IIoT IDS. CNN, renowned for its capability to extract intricate features from input data, hold significant potential for enhancing the detection and classification performance of IDS models.

Table 1 Characteristics of conventional IT systems as compared to IIoT [2]

Category	Conventional IT system	IIoT system
Performance requirements	Non-real time, consistent response, high throughput	Real-time, time-critical response, modest throughput
	High delay and jitter may be acceptable	High delay and/or jitter not acceptable
	Tightly restricted access control for security	Strict control with minimal interference in human–machine interaction
Availability requirements	Responses like rebooting acceptable	Rebooting may not be acceptable, necessitating redundant systems
	Availability deficiencies tolerated based on operational needs	High availability requires exhaustive pre-deployment testing
Risk management requirements	Managing data, data confidentiality, and integrity paramount	Control physical world, prioritize human safety, and process protection
	Fault tolerance less important, momentary downtime acceptable	Fault tolerance essential, momentary downtime may be unacceptable
System operation	Designed for typical operating systems, straightforward upgrades	Software changes require careful consideration due to specialized control algorithms
Resource constraints	Systems specified with enough resources for third-party applications	Systems designed for industrial processes may lack resources for additional security capabilities
Communications	Standard communications protocols, primarily wired networks	Proprietary and standard communication protocols, complex networks, primarily wireless

Table 2 Categorization of IDS systems in the literature based on security concerns

References	Addressed security concern(s)				
	Confidentiality	Integrity	Availability	Authorization	Authentication
[15]	X				
[16]		X			X
[17]			X		
[18]			X		
[19]	X	X	X		
[20]	X	X	X	X	
[21]	X	X	X	X	
[22]	X	X	X	X	

Table 3 Categorization of the IDS systems in the literature based on dataset

References	Dataset used	Dataset access status	Comments
[15]	WUSTLIIOT-2018 dataset	Public	IIoT-specific
[16]	IEEE 118 bus network simulations	Private	Not IIoT-specific
[17]	NSL-KDD & UNSW-NB15 datasets	Public	Not IIoT-specific
[18]	NSL-KDD & UNSW-NB15 datasets	Public	Not IIoT-specific
[19]	Flow-based dataset	Private	IIoT-specific
[20]	WUSTL-IIOT-2021 dataset	Public	IIoT-specific
[21]	WUSTL-IIOT-2021 dataset	Public	IIoT-specific
[22]	WUSTL-IIOT-2021 dataset	Public	IIoT-specific

4. Absence of mitigation for dataset imbalance: The majority of the reviewed works did not attempt to address class imbalance issues in the datasets. By failing to consider the effect of the dataset imbalance, the performance of the developed IDS models may be negatively affected, leading to biased results and reduced effectiveness in real-world scenarios.

Table 4 Categorization of the IDS systems in the literature based on the utilized algorithm

References	ML model considered				
	DBN	ANN	MLP	CNN	AE
[15]			X		
[16]	X	X			
[17]					X
[18]				X	
[19]		X			
[20]	X	X			
[21]					X
[22]				X	

1.3 Objectives and contributions

This work aims to innovatively address the complex and evolving challenge of detecting cybersecurity intrusions in ICS and IIoT environments. Specifically, the objective is to develop and optimize a CNN-based multi-class attack classifier that not only accurately identifies a broad spectrum of security attacks but also adapts to the unique network traffic patterns characteristic of IIoT systems. This involves focusing on real-time intrusion detection capabilities, minimizing false positives, and enhancing detection accuracy for both common and emerging attack vectors. The research seeks to bridge the gap in current intrusion detection strategies, which often struggle to effectively manage the distinct and diverse data flows in IIoT networks.

The contributions of this work can be summarized in the following points:

1. Development of a novel CNN-based IDS model specifically tailored for multi-class intrusion detection in IIoT networks.
2. Optimization of the CNN model's hyperparameters through grid search to develop a highly effective IDS tailored to the unique demands of intrusion detection in IIoT networks.
3. Implementation of a robust comprehensive evaluation methodology that encompasses both binary and multi-class attack classification.
4. Demonstration of the model's generalizability and effectiveness across diverse IIoT-specific and non-IIoT-specific environments.

2 Technical background

This section explores key elements underlying the research, focusing on the application of CNNs for IDS in the domain of IIoT. The section discusses the selection of CNN for the implementation of the IDS for IIoT networks. It also discusses the two datasets that will be used for training and evaluating the different aspects of the IDS. Furthermore, this section discusses the evaluation metrics that will be used to evaluate the performance of the model and the IDS.

2.1 Convolutional neural networks for IIoT IDS

It can be observed from the survey summary presented in Table 4 that the majority of IDS models proposed in the literature for IIoT systems have been developed using traditional ML algorithms, with only a limited number utilizing DL algorithms. However, studies have shown that DL models, particularly CNN, offer superior predictive capabilities compared to ML models [23, 24]. The inherent feature extraction capability of CNN eliminates the need for manual feature engineering, resulting in improved model efficiency and performance. This makes CNN a highly suitable algorithm for IDS applications, as it can effectively analyze network traffic attributes [25, 26].

Typically, a CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. To create a deep model, these modules are stacked together, and in some cases, combined with additional deep neural network layers, as illustrated in Fig. 1 [27]. The deep hierarchical structure of CNN enables it to learn complex patterns and representations from the input data, making it well-suited for capturing intricate characteristics of network traffic and identifying potential threats.

2.2 The domain-specific WUSTL-IIOT-2021 dataset

This research leverages the WUSTL-IIOT-2021 dataset for training and optimizing the CNN model that will be used to construct the IDS system [28]. This is the most up-to-date industry-specific dataset tailored for intrusion detection in IIoT environments. It was developed in 2021 at the Washington University in St. Louis, and originates from a realistic IIoT testbed that captures a diverse range of network activities encountered in industrial environments [20]. It is comprised of close to 1.2 million samples that document network traffic under normal behavior as well as four authentic cyberattacks that were deliberately launched at the IIoT system: Denial of Service (DoS), Reconnaissance, Command Injection, and Backdoor. The diagram in in Fig. 2 depicts the distribution of the dataset and

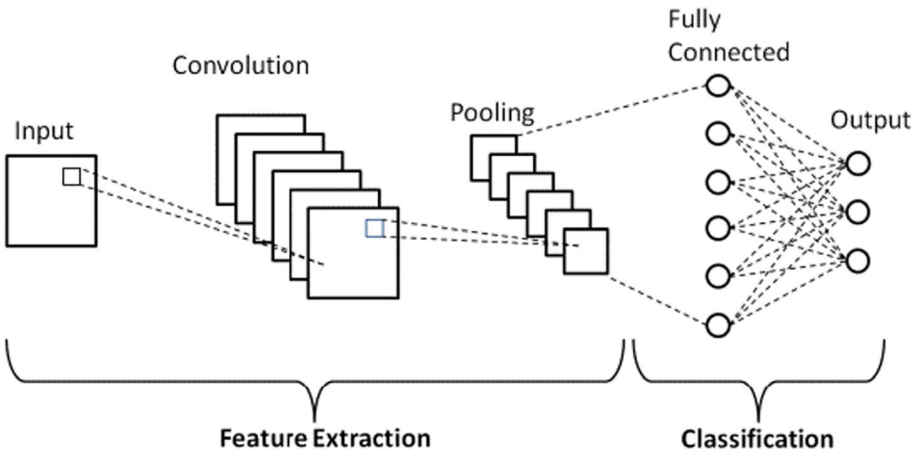

Fig. 1 Illustration of a deep CNN architecture [27]

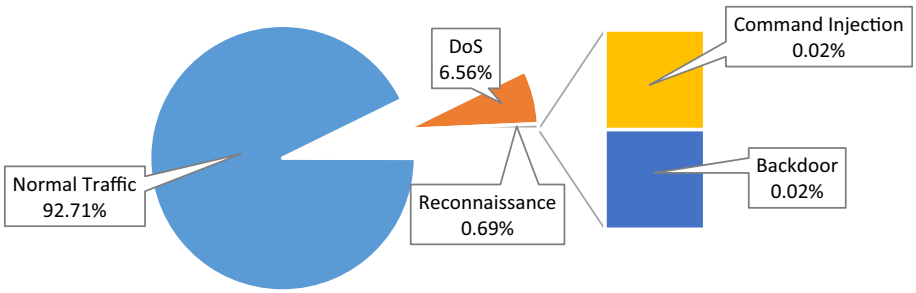

Fig. 2 Distribution of records in the WUSTL-IIOT-2021 dataset

highlights a significant imbalance across the different scenarios, which is a critical factor to address for the effective performance of the CNN model.

2.3 The non-domain-specific UNSW_NB15 dataset

This research will also utilize the UNSW_NB15 dataset to assess the generalizability of the developed IDS model beyond the specific context of IIoT. The goal of using this non-IIoT-specific dataset is to demonstrate the robustness and adaptability of the model in the wider scope of conventional IT environments. The UNSW_NB15 dataset was created in 2015 by researchers at the Australian Centre for Cyber Security (ACCS) and is widely used for research on intrusion detection in conventional IT networks [29]. This dataset was created using a simulated environment that reflects the nature of generic IT networks. It combines real-world normal behavior with synthetic attack activities representing nine attack types. The distribution of the records is presented in the diagram of Fig. 3 and clearly highlights a very significant class imbalance, similar to the WUSTL-IIOT-2021 Dataset.

2.4 Evaluation metrics

Typically, the evaluation of DL models relies on the standard effectiveness metrics Accuracy, Recall, Precision, and *F*1-Score, which are derived from the elements of the confusion matrix depicted in Table 5 [30, 31]. For this application, the confusion matrix can be defined as follows [32]:

- True negatives (TN): The number of legitimate packets that have been correctly classified as normal, indicating the model's ability to correctly identify normal instances.
- True positives (TP): The number of malicious packets that have been correctly classified as an attack, demonstrating the model's capacity to accurately detect attacks.
- False positives (FP): The number of legitimate packets that have been incorrectly classified as an attack, representing instances where the model misclassifies normal packets as malicious.
- False negatives (FN): The number of malicious packets that have been incorrectly classified as legitimate, indicating instances where the model fails to detect an attack.

Considering that the primary goal of an IDS is to reduce false negatives (FNs) while maximizing true negatives

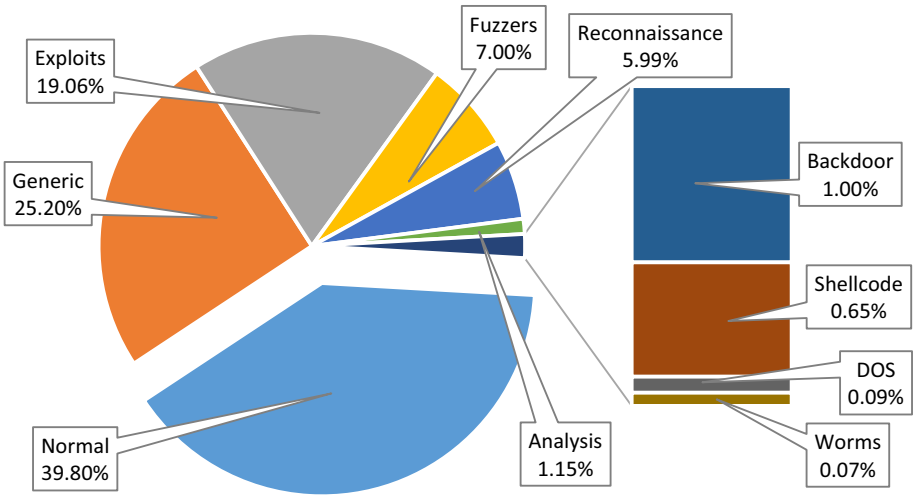

Fig. 3 Distribution of records in the UNSW_NB15 dataset

Table 5 Confusion matrix parameters for evaluating the IIoT IDS

	Actual legitimate traffic (0)	Actual malicious traffic (1)
Predicted legitimate traffic (0)	True negative (TN)	False negative (FN)
Predicted malicious traffic (1)	False positive (FP)	True positive (TP)

(TNs), the evaluation of the proposed model will primarily emphasize the *recall* metric. This metric assesses the model's capability to effectively detect and classify malicious network traffic.

Two additional critical figures of merit for the performance of DL models are training time and testing time. Training time refers to the duration required to train the model on the dataset. It reflects the computational efficiency and resource consumption of the model during the learning phase. A shorter training time is desirable in industrial contexts where rapid deployment and updates of the IDS are necessary. Testing time, on the other hand, indicates the time the model takes to evaluate and classify new data after training. This metric is essential for assessing the model's practicality in real-time detection scenarios. In IIoT environments, where timely response to potential threats is critical to prevent disruptions or damage to industrial processes, a model with a shorter testing time is highly advantageous.

3 Methodological approach

This section details the procedural framework that has been adopted in this research. It will discuss the process used for creating the optimized CNN model using the IIoT-specific WUSTL-IIOT-2021 Dataset, then evaluating the generalizability of the developed model to the non-IIoT-specific conventional IT environment using the UNSW_NB15 Dataset. These processes are summarized in the procedure shown in Fig. 4, and will be discussed in detail in the following sub-sections.

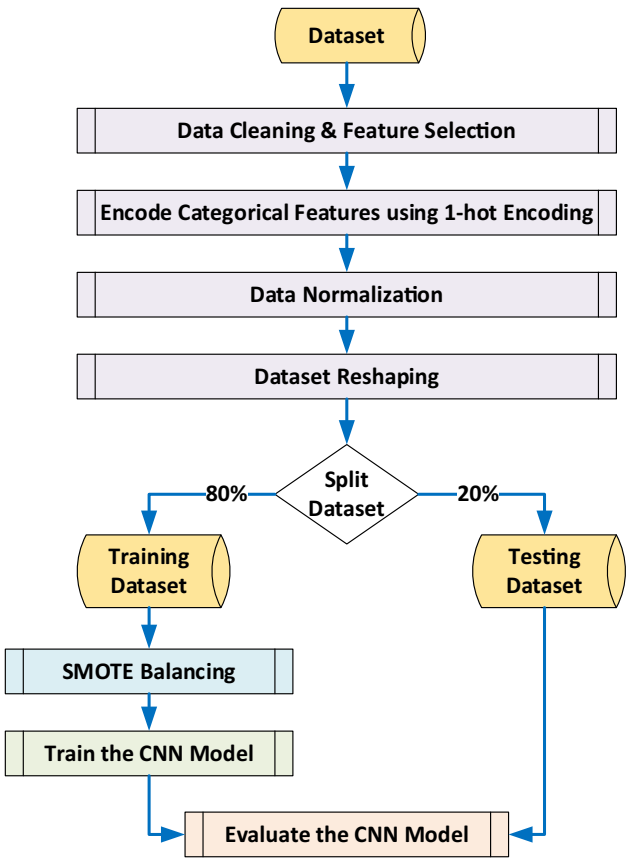

Fig. 4 Procedure for constructing and the IDS

3.1 Dataset cleaning

A number of data preprocessing steps were implemented to enhance the quality and reliability of the data in the datasets [33–35]. First, all duplicate records and duplicate features were removed to avoid biasing the results of the ML model. Additionally, all records with missing data were dropped from consideration. This has negligible impact on the model's performance given the abundance of records in the datasets and the small number of duplicated records. Also, non-essential elements such as hashtags, URLs, emoticons, HTML tags, and dates were also removed.

3.2 Feature selection

A thorough feature selection was applied to each of the datasets to remove non-essential features and potential attack identifiers. The goal was ensuring the model's focus on relevant data, fostering improved generalization, and achieving robust performance in practical deployment scenarios.

The WUSTL-IIOT-2021 dataset documented 41 features [28]. Following the recommendation by the publishers of the dataset, four features were excluded because they are "unique to the attacks and would expose the type of the attack to the model; therefore, the model would not be generalized for unseen data." The recommended features for exclusion were 'StartTime', 'LastTime', 'SrcAddr' and 'DstAddr', 'sIpId', and 'dIpId' [20, 28]. After excluding these specific features, all remaining features in the dataset were retained. On the other hand, the UNSW-NB15 dataset documented 49 features, none of which was excluded [29].

3.3 Data type conversion and normalization

The datasets are comprised of categorical and numerical features. Therefore, in order to unify the handling, categorical features were encoded using one-hot encoding. Then, all feature values were unified under the "Int64" data type to ensure data representation uniformity and consistency. After that, all attributes were rescaled using Min–Max normalization to achieve proportionally and preserve the relationships between the data points, contributing to improved ML model accuracy.

3.4 Dataset reshaping

CNN models are designed to detect spatial patterns within the input data. These models typically include multiple convolutional layers that apply a sliding kernel over the input data to detect features [36]. This requires the input data to be multidimensional (images or matrices) in order to represent spatial features and allow the detection of complex patterns. Therefore, the textual network traffic datasets were reshaped into multi-dimensional arrays (known as tensors). The normalized dataset was first converted into a matrix by mapping the dataset entries to integer values ranging from 0 to 255. Subsequently, padding was applied to ensure that each row of data consisted of 64 values. The images in Fig. 5 demonstrate the tensors created by transforming each of the five classes of the WUSTL-IIOT-2021 dataset into a corresponding 2-dimensional matrix [37].

3.5 Dataset partitioning

Upon completion of the above-mentioned data preparation steps, the datasets were split into a training dataset (80% of the samples) and a testing dataset (20% of the samples). To ensure an unbiased and robust evaluation of the model, the datasets were randomly shuffled first, then samples were selected through a random process to create the testing subsets. Afterward, the remaining samples were used to create the training sub-sets. This random sampling approach enhances the comprehensive assessment of the model's robustness and minimizes potential biases in the evaluation process. The partitioning of the two datasets is reflected in Table 6 for the WUSTL-IIOT-2021 dataset and Table 7 for the UNSW_NB15 dataset.

3.6 Balancing the training datasets

It was noted in Sects. 2.2 and 2.3 that both datasets were severely imbalanced. It was decided to deliberately maintain the testing sub-sets inherent imbalance to mirror the real-world conditions of an IIoT network under attack, providing a realistic scenario for evaluating the model's performance. On the other hand, it was decided to evaluate the effect of artificially balancing the training datasets on the performance of the CNN model. To support this evaluation, two versions of the training sub-sets were created: an unbalanced version reflecting the nature of the datasets, and a forcefully balanced version.

A previous study explored a range of balancing techniques for this particular application [38]. The study determined that Synthetic Minority Over-Sampling Technique (SMOTE) produced the best performance in mitigating the class imbalance issues [39]. Accordingly, a novel multi-class implementation of the SMOTE balancing technique was used to build multiclass balanced training datasets. The target of this process was to force a complete balance of the records representing the different attack types in the training datasets.

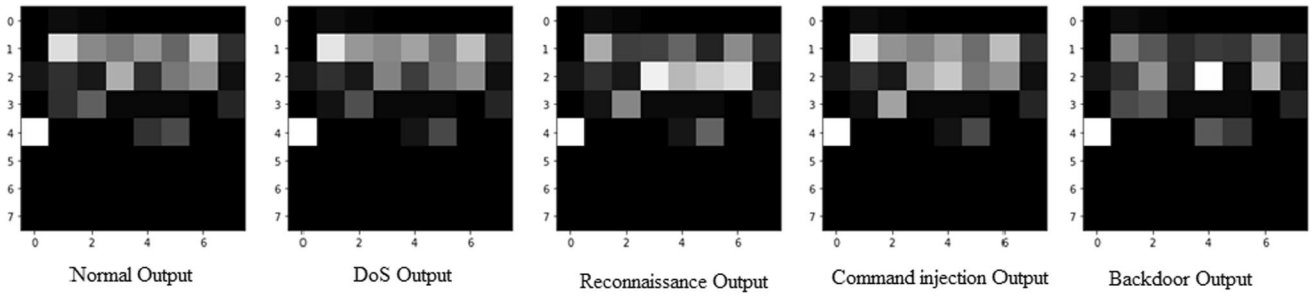

Fig. 5 Tensor representation for the features in each of the WUSTL-IIOT-2021 classes

Table 6 Applying multi-class SMOTE balancing to the WUSTL-IIOT-2021 dataset

Category	Records after data cleaning	Testing dataset (20%)	Unbalanced training dataset (80%)	SMOTE-balanced training dataset
Normal traffic	972,190	194,438	777,752	777,752
DoS	68,750	13,750	55,000	777,752
Reconnaissance	7216	1443	5773	777,752
command injection	230	46	184	777,752
Backdoor	243	49	194	777,752
Total samples	1,048,629	209,726	838,903	3,888,760

Table 7 Applying multi-class SMOTE balancing to the UNSW_NB15 dataset

Category	Records after data cleaning	Testing dataset (20%)	Unbalanced training dataset (80%)	SMOTE-balanced training dataset
Normal	93,000	18,600	74,444	74,444
Generic	58,871	11,774	47,056	74,444
Exploits	44,525	8905	35,613	74,444
Fuzzers	24,246	4849	19,391	74,444
DOS	16,353	3271	13,093	74,444
Reconnaissance	13,987	2797	11,184	74,444
Analysis	2677	535	2155	74,444
Backdoor	2329	466	1879	74,444
Shellcode	1511	302	1185	74,444
Worms	174	35	138	74,444
Total samples	257,673	51,534	206,138	744,440

The training sub-set extracted from the WUSTL-IIOT-2021 dataset contained 777,752 normal traffic samples, while the different attacks were represented by much lower sample counts (as shown in the third column of Table 6). To achieve a balanced representation, SMOTE was used to increase the number of samples for each of the attacks to match the number of normal traffic samples (as shown in the last column of Table 7). It is to be noted that the size of the training dataset grew by about 460% as a result of the balancing operation compared to the unbalanced dataset.

On the other hand, the training sub-set extracted from the UNSW_NB15 dataset was initially comprised of 206,138 samples, with normal traffic far exceeding the representations of the different malicious attacks. A similar process was applied to create the multi-class SMOTE-balanced training dataset shown in the last column of Table 7.

3.7 Development of the baseline CNN model

A baseline CNN model was constructed to systematically extract and learn the pertinent features from the input images. The layers of this baseline model were organized into 4 blocks as shown in Fig. 6, and the base configuration for the different layers is documented in Table 8.

3.8 Optimizing the CNN model hyperparameters

The data in Tables 6 and 7 show that the size of the training datasets increased significantly when the datasets were balanced. This manifested in similarly significant increases in the training and testing times of early experiments. To address this challenge, a comprehensive optimization process was carried out for the most important hyperparameters that affect the model's speed and performance. The objective was to produce a streamlined and computationally efficient model while preserving its predictive performance.

Ten of the most critical hyperparameters affecting the performance of the CNN model were chosen for optimization as shown in Table 9. The grid search approach was used to explore the various hyperparameter combinations and identify the optimal configuration that maximized accuracy and minimized loss [40]. In cases where multiple hyperparameter settings yielded the same accuracy and loss values, preference was given to the configuration that minimized computational time. The range of values considered for the different parameters is tabulated in the second column of Table 10.

After an extensive process, the optimal value for each of the parameters was determined as shown in the last column of Table 11. In summary, the baseline model described in Fig. 6 and Table 8 was reduced to the optimized model shown in Fig. 7 with the layer configuration shown in Table 11.

4 Results and discussion

This section presents a detailed analysis of the experiments that have been conducted to fulfill the objectives outlined in Sect. 1.3. These experiments were designed to test the effectiveness and efficiency of the proposed IDS under various scenarios. As illustrated in Fig. 8, a number of stages were implemented to culminate with the development of the optimized model. The development of the optimized model utilized the WUSTL-IIOT-2021 IIoT-specific dataset exclusively to ensure capturing the specific characteristics of the IIoT environment. Then, an experiment was conducted using the UNSW_NB15 Non-IIoT-Specific dataset to demonstrate the generalizability and effectiveness of the resulting IDS to the wider scope of conventional IT networks. The following sub-sections will present the experiments and discuss their results.

4.1 Performance of the baseline CNN model on the unbalanced dataset

The first experiment evaluated the performance of the baseline CNN model on the default WUSTL-IIOT-2021 dataset for classification between the different attack scenarios present in the dataset. The baseline CNN model (described in Table 8) was trained using the *unbalanced* training dataset and then evaluated on the testing dataset. The results in Table 12 show the performance metrics obtained for this experiment and will be used as the baseline for the other experiments.

Additionally, Fig. 9 shows the confusion matrix of the baseline model's predictive accuracy for the different types of attacks. The matrix shows that the baseline model was able to classify the records correctly for the classes were there are abundant records for the model to train on (normal traffic, and the Denial of Service and Reconnaissance attacks). It did not perform as well for the severely under-represented classes (Command Injection and Backdoor attacks). In fact, the model exhibited concerning misclassifications for these under-represented classes.

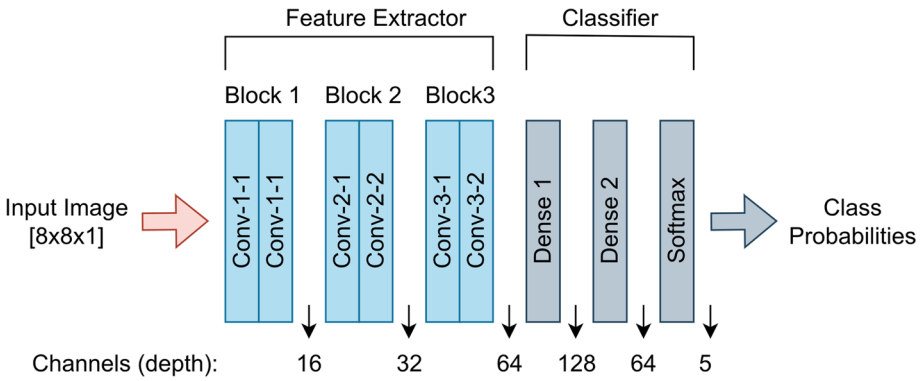

Fig. 6 Block diagram of the baseline CNN model

Table 8 Structure of the baseline CNN model

Type	Block	Layer	Details
Feature extraction	First block	Conv-1-1	Batch normalization
			Activation = swish
			Conv2D, filter size = 16, dropout rate = 0.0
		Conv-1-2	Batch normalization
			Activation = swish
			Conv2D, filter size = 16, dropout rate = 0.0
	Second block	Conv-2-1	Batch normalization
			Activation = swish
			Conv2D, filter size = 32
		Conv-2-2	Batch normalization
			Activation = swish
			Conv2D, filter size = 32, dropout rate = 0.0
	Third block	Conv-3-1	Batch normalization
			Activation = swish
			Conv2D, filter size = 64, dropout rate = 0.0
		Conv-3-2	Batch normalization
			Activation = swish
			Conv2D, filter size = 64, dropout rate = 0.0
Classification	Output	Flatten	
		Dense 1	Dense units = 128, activation = swish
			Dropout rate = 0.3
		Dense 2	Dense units = 64, activation = swish
			Dropout rate = 0.3
		SoftMax	Dense = 5, activation = SoftMax

Table 9 The CNN model hyperparameters selected for optimization

Hyperparameter	Justification
Number of convolution layers Number of dense layers	Optimizing the number of convolutional and dense layers is critical in achieving both effective representation and computational efficiency. Using more layers allows learning more complex features and patterns. However, an excessive number of layers can lead to overfitting and computational complexity
Filter size	Choosing an optimal filter size improves the model's adaptability to various feature scales. Larger filters capture global patterns, while smaller filters focus on local details
Number of neurons in each dense layer	The number of neurons in the dense layers determines the model's capacity to learn and generalize from the data
Activation functions	Selecting the appropriate activation function impacts the effectiveness of the model's ability to learn complex patterns and make predictions
Batch sizes	The batch size affects the model's convergence, computational efficiency, and generalization ability
Convolutional layer dropout rate Dense layer dropout rate	Dropout is a regularization technique that helps prevent overfitting. It is important to find a balance between regularizing the model to prevent overfitting and allowing it to learn complex representations
Optimizer	Optimizers manage the model's learning process by adjusting weights to minimize the loss function. Choosing the right optimizer influences convergence speed, and ensures effective learning
Learning rates	Learning rate influences the speed, stability, and quality of model convergence during training, thereby affecting the model's performance and accuracy

Table 10 Optimization values for the CNN model hyperparameters

Hyperparameter	Values considered	Optimal value
Number of convolution layers	2, 4, 6	2
Number of dense layers	0, 1, 2, 3	2
Filter size	8, 16, 24, 32, 48	16
Number of neurons in each dense layer	32, 64, 128, 256, 512, 1024, 2048	128/64
Activation functions	ReLu, Leaky_ReLu, Swish	Leaky ReLU
Batch sizes	128, 256, 512, 1024	512
Dropout rate in the convolutional layer	0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%	0%
Dropout rate in the dense layer	0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%	10%
Optimizer	Sgd, rmsprop, adagrad, adadelta, adam, adamax, nadam	Adam
Learning rates	1E-2, 1E-3, 1E-4, 1E-5, 1E-6, 1E-7	1E-4

Table 11 Structure of the optimized CNN model

Type	Block	Layer	Details
Feature Extractor	First block	Conv-1-1	Batch normalization = 512
			Activation = Leaky ReLU
			Conv2D, filter size = 16
		Conv-1-2	Batch normalization = 512
			Activation = Leaky ReLU
			Conv2D, filter size = 16
Classifier	Output	Flatten	
		Dense 1	Dense units = 128, activation = LeakyReLU
			Dropout rate = 0.1
		Dense 2	Dense units = 64, activation = LeakyReLU
			Dropout rate = 0.1
		SoftMax	Dense = 5, activation = SoftMax

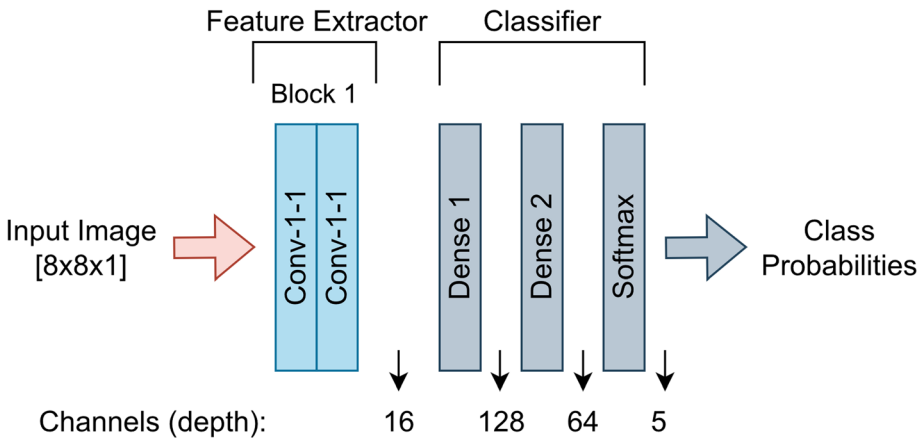

Fig. 7 Block diagram of the optimized CNN model

4.2 Effect of dataset balancing on the performance of the baseline CNN model

To determine the effect of balancing the training dataset on detection performance, the baseline model was trained on the multi-class SMOTE-balanced training dataset and evaluated using the unbalanced testing dataset. The results presented in Table 13 demonstrate that balancing the training dataset with the novel multi-class SMOTE implementation had a significantly positive effect on the performance of the model, in general. The performance

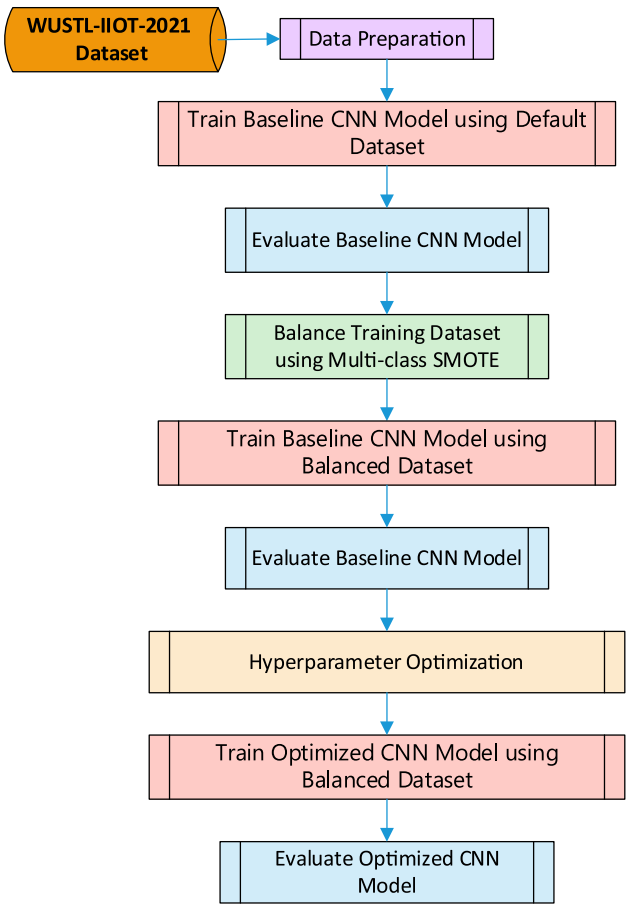

Fig. 8 Development process for the optimized CNN model

Table 12 Performance of the baseline model using the default WUSTL-IIOT-2021 dataset

Metric	Performance
Accuracy	99.94%
Precision	94.74%
Recall	92.29%
*F*1-score	93.41%
Training time (s)	264
Testing time (s)	89

improved beyond 99.9% by all metrics, with a maximum improvement of 8.27% compared to the unbalanced case.

In contrast, Table 13 shows that the training time increased in excess of 300%, and the testing time increased by about 230% when the model was trained on the balanced dataset. The increase in the training time was expected given the very significant expansion of the training dataset as a result of the balancing process. It can be argued that this increase in the training time will have little effect on the overall responsiveness of the IDS since the model will be trained offline before it is deployed. However, the substantial rise in testing time raises concerns, especially considering that the testing dataset was identical. This clearly points to a highly adverse impact on the responsiveness of the IDS when it will be used to classify real-time data.

The confusion matrix for the baseline CNN model trained using the multi-class SMOTE-balanced dataset is shown in Fig. 10. The matrix essentially shows that the model was able to classify each of the attacks at 100% accuracy. It also shows 0% misclassification between the different attacks. It is clear that balancing the training dataset had a very positive effect on the ability of the CNN model to classify the different attacks in the dataset.

The results in Table 14 highlight the effect of balancing the training dataset on the performance of the baseline model in detecting each of the different attacks in the dataset. The results reveal a notable improvement in precision, recall, and *F*1-score for detecting all of the attack scenarios. However, it is important to point out the slight decrease in classification performance for the normal samples. This could be attributed to the introduction of synthetic data points during the SMOTE balancing process, which might slightly affect the classification accuracy for the majority class. Nonetheless, the results demonstrate the effectiveness of SMOTE balancing on enhancing the model's ability to classify and detect various types of attacks with higher precision and recall.

4.3 Performance of the optimized model in an IIoT-specific setting

The optimized model (as described in Fig. 7 and Table 11) was trained using the multi-class SMOTE-balanced training dataset, and then its performance was evaluated using the unbalanced testing dataset. The results in Table 15 show that the optimized model maintained the excellent

Fig. 9 Confusion matrix for the baseline CNN model trained using the default dataset

		Predicted				
		Backdoor	Command Injection	DoS	Recon.	Normal
Target	Backdoor	0.78	0.03	0.00	0.00	0.19
	Command Injection	0.12	0.84	0.00	0.00	0.04
	DoS	0.00	0.00	0.99	0.00	0.01
	Recon.	0.00	0.00	0.00	1.00	0.00
	Normal	0.00	0.00	0.00	0.00	1.00

Table 13 Effect of SMOTE balancing on the general performance of the baseline CNN model

Metric	Performance		Effect
	Unbalanced dataset	Balanced dataset	
Accuracy	99.94%	99.99%	0.05% Improvement
Precision	94.74%	99.92%	5.47% Improvement
Recall	92.29%	99.92%	8.27% Improvement
*F*1	93.41%	99.92%	6.97% Improvement
Training time (s)	264	1134	330% Worse
Testing time (s)	89	283	229% Worse

Fig. 10 Confusion matrix for the baseline CNN model trained using the balanced dataset

Target \ Predicted	Backdoor	Command Injection	DoS	Recon.	Normal
Backdoor	1.00	0.00	0.00	0.00	0.00
Command Injection	0.00	1.00	0.00	0.00	0.00
DoS	0.00	0.00	1.00	0.00	0.00
Recon.	0.00	0.00	0.00	1.00	0.00
Normal	0.00	0.00	0.00	0.00	1.00

Table 14 Effect of dataset balancing on the classification of the individual attack scenarios

Attack scenario	Precision (%)		Recall (%)		*F*1-score (%)	
	Unbalanced dataset	Balanced dataset	Unbalanced dataset	Balanced dataset	Unbalanced dataset	Balanced dataset
Backdoor	76.85	99.94	78.30	99.65	77.57	99.8
Command injection	97.31	99.62	83.78	100	90.04	99.81
Denial of service	99.9	99.96	99.39	99.96	99.65	99.95
Reconnaissance	99.71	100	99.99	99.90	99.85	99.85
Normal	99.95	99.89	99.99	99.89	99.97	99.89

Table 15 Comparative performance of the optimized CNN model

Metric	Performance of different models			Improvement of optimized model	
	Baseline model unbalanced dataset	Baseline model balanced dataset	Optimized model balanced dataset	Compared to baseline model unbalanced dataset (%)	Compared to baseline model balanced dataset (%)
Accuracy	99.94%	99.99%	99.90%	− 0.04	− 0.09
Precision	94.74%	99.92%	99.90%	5.45	− 0.02
Recall	92.29%	99.92%	99.90%	8.25	− 0.02
*F*1-score	93.41%	99.92%	99.90%	6.95	− 0.02
Training time (s)	264	1134	317	20.08	− 72.05
Testing time (s)	89	283	231	159.55	− 21.16

performance in classifying the different attack scenarios, while significantly reducing the negative effect of the dataset balancing on the training and testing times of the model.

Overall, these results show that the combination of a balanced dataset and an optimized model significantly improved the performance of the CNN model in classifying the different attack scenarios. Additionally, the model maintained a reasonable level of responsiveness, rendering it well-suited for application in a real-time IDS system.

4.4 Performance of the optimized model in binary (normal vs. attack) classification

The results presented in the preceding section demonstrated the effectiveness of the optimized CNN model in detecting the five different behaviors documented in the WUSTL-IIOT-2021 dataset in the network traffic of the IIoT system. However, standard practice for network operators is to initially determine whether the network is functioning normally or exhibiting symptoms of a cybersecurity attack, before trying to determine the specific attack type. Accordingly, an experiment was conducted to assess the performances of the developed model (optimized for multi-classification between the attack types) in binary classification between normal and potentially malicious traffic.

A customized binary version of the WUSTL-IIOT-2021 dataset was created using the following procedure:

1. 20% of samples from the cleaned WUSTL-IIOT-2021 dataset were selected randomly to create the testing dataset. No balancing was applied to this sub-set to ensure accurate representation of typical network traffic.
2. The remaining 80% of samples were used to form the training sub-set. This sub-set contained a total of 777,752 samples of normal traffic. In order to create a binary balanced training dataset, the combined number of samples for the attacks needed to be raised to match this count. However, it was decided to balance the samples from the different attacks to ensure equal representation of the different attack types. Accordingly, the SMOTE technique was used to raise the number of samples for each attack type to 194,440. This achieved a balance between the different attack types, and a balance between the combined number of attack samples and normal traffic samples, as illustrated in the fourth column of Table 16.
3. Finally, all malicious samples in both the training and testing datasets were relabeled with a generic "attack" label to create the binary (normal vs. attack) classification.

The optimized model was trained using the *binary-labeled balanced* training dataset and evaluated using the *binary-labeled unbalanced* testing dataset. The results in Table 17 show that the model was very effective in the binary classification between normal and attack traffic. This demonstrates that the developed IDS model can be effectively used initially to detect that the system is under attack, and subsequently determine the specific type of attack.

4.5 Generalizability of the proposed IDS beyond IIoT

In order to demonstrate the versatility of the resulting IDS model, its generalizability was finally evaluated using the non-domain-specific UNSW_NB15 dataset, which is a widely recognized benchmark for intrusion detection in conventional IT networks. This choice was deliberate, aiming to evaluate the versatility of the developed IDS model beyond the specific demands of IIoT.

Two experiments were conducted to evaluate the performance of the model based on the UNSW_NB15 dataset. In the first experiment, the baseline CNN model was trained on the default unbalanced dataset. On the other hand, the second experiment reflected the methodology of the proposed IDS, where the optimized model was trained using the SMOTE-balanced training dataset. Both experiments utilized the same unbalanced testing dataset to ensure a fair and consistent comparison.

The results in Table 18 demonstrate that the developed IDS model is *generally* generalizable and is able to detect intrusions in a generic IT network, even though it was designed and optimized using IIoT-specific data. The third column of the table also shows that the model's intrusion

Table 16 Applying binary SMOTE balancing to the training dataset

Category	Records after data cleaning	Unbalanced training dataset (80%)	SMOTE-balanced training dataset	Binary-labeled training dataset
Normal traffic	972,190	777,752	777,752	Normal
DoS	68,750	55,000	194,440	Attack
Reconnaissance	7216	5773	194,440	
Command Injection	230	184	194,440	
Backdoor	243	194	194,440	
Total samples	1,048,629	838,903	1,555,512	

Table 17 Performance of the optimized model using the binary-labeled balanced dataset

Metric	Performance (%)
Accuracy	99.82
Precision	94.69
Recall	92.84
*F*1-score	93.77

detection performance improved when the full IDS methodology was implemented. Interestingly, despite increasing the training dataset's size by almost 4 times, the optimized model exhibited a slightly shorter training time, suggesting significant efficiency improvements. On the other hand, testing time for the optimized model was significantly higher even though the size of the testing dataset was the same.

Table 19 presents a comparison between the performance of the developed model using the IIoT-specific dataset (WUSTL-IIOT-2021) versus the non-IIoT-specific dataset (UNSW_NB15). The highly superior performance of the developed model on the IIoT-specific dataset emphasizes that the methods employed in this work have succeeded in developing a model that is highly fine-tuned to the distinct nature of intrusions in IIoT networks. On the other hand, this analysis demonstrated that while the proposed model is generalizable to the broader context of conventional IT networks, its effectiveness reduces significantly when used in a conventional IT setting.

5 Limitations and future work

While the current study has successfully developed an effective IDS for IIoT environments, several limitations are recognized which compel further exploration. These limitations encompass the exploration of diverse datasets other than the ones used in this work, the evaluation of alternative DL algorithms, the refinement of feature engineering techniques, the investigation of real-time implementation, and addressing challenges related to interpretability and explainability. By addressing these limitations, the performance and applicability of IDSs in IIoT environments can be significantly enhanced.

6 Conclusion

This work presents the development of an Intrusion Detection System (IDS) for Industrial Internet of Things (IIoT) environments. The proposed IDS uses an optimized convolutional neural network (CNN) model trained on a multi-class-balanced dataset, allowing the CNN model to learn and generalize more effectively across all classes. Extensive evaluation showed that Synthetic Minority Over-Sampling Technique (SMOTE) was particularly effective in improving the performance of CNN-based IDS systems. Therefore, a novel multi-class implementation of SMOTE was developed to create a training dataset where all classes were equally represented. Results presented in this work

Table 18 Performance of the Proposed IDS on the UNSW_NB15 Dataset

Metric	Performance		Effect on performance (%)
	Baseline model unbalanced dataset	Optimized model balanced dataset	
Accuracy	86.11%	82.78%	− 4
Precision	83.61%	84.44%	1
Recall	61.92%	82.55%	33
*F*1-score	64.8%	82.8%	28
Training time (s)	197	178	− 10
Testing time (s)	8	25	213

Table 19 Comparison of the performance of the proposed IDS on the IIoT-specific versus non-IIoT-specific dataset

Metric	Performance	
	WUSTL-IIOT-2021 dataset (%)	UNSW_NB15 dataset (%)
Accuracy	99.90	82.78
Precision	99.90	84.44
Recall	99.90	82.55
*F*1-score	99.90	82.8

have shown a very significant improvement in the performance of the CNN model when this balanced training dataset was used.

However, balancing the training dataset resulted in a significant increase in the number of records which increased the training and testing times of the CNN model. Therefore, a systematic hyperparameter optimization was carried out to fine tune the model parameters. This optimization resulted in a highly efficient and effective IDS model that could classify network intrusions with performance reaching a level of 99.9% on all metrics. Additionally, the training and testing times for the IDS reduced by about 70% and returned to the same general level compared to their levels before the dataset balancing.

The proposed system was evaluated using the IIoT-specific WUSTL-IIOT-2021 dataset and consistently demonstrated the superiority of the proposed IDS model. Significant improvements in accuracy, precision, recall, and $F1$-score affirm the model's proficiency in identifying and classifying diverse network intrusions within IIoT contexts. The optimized model was also evaluated using the non-domain-specific UNSW_NB15 dataset to showcase its generalization capability across diverse scenarios and unseen data. The study's findings highlight the potential of the proposed IDS to detect intrusions beyond specific IIoT environments, positioning it as a reliable solution for generic network behavior in intrusion detection.

In summary, the developed IDS, with its optimized model and balanced dataset, presents a promising solution to bolstering the security of IIoT ecosystems. As technology advances and threats evolve, ongoing research efforts will play a pivotal role in ensuring the continuous enhancement and adaptability of IDSs to safeguard critical industrial systems.

Acknowledgements The authors would like to convey their thanks and appreciation to the "University of Sharjah" for supporting this work.

Data availability This study relies on publicly available datasets (WUSTL-IIOT-2021 and UNSW-NB15). These datasets are available from the following references.Zolanvari, M., Gupta, L., Khan, K. M., & Jain, R. (2021). WUSTL-IIOT-2O2l Dataset for IIoT Cybersecurity Research. Washington University in St. Louis, USA. Accessible from-https://www.cse.wustl.edu/∼jain/iiot2/index.html.N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015—Proc. (2015). https://doi.org/10.1109/MilCIS.2015.7348942. Accessible from-https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Informed consent This study does not involve any experiments on animals.

References

1. Attar H (2023) Joint IoT/ML platforms for smart societies and environments: a review on multimodal information-based learning for safety and security. J Data Inf Qual. https://doi.org/10.1145/3603713
2. Abdul Rahman Al-chikh Omar A, Soudan B, Altaweel A (2023) A comprehensive survey on detection of sinkhole attack in routing over low power and Lossy network for internet of things. Internet of Things (Netherlands). https://doi.org/10.1016/j.iot.2023.100750
3. Alamleh A, Albahri OS, Zaidan AA, Albahri AS, Alamoodi AH, Zaidan BB, Qahtan S, Alsatar HA, Al-Samarraay MS, Jasim AN (2023) Federated learning for IoMT applications: a standardization and benchmarking framework of intrusion detection systems. IEEE J Biomed Heal Inform 27:878–887. https://doi.org/10.1109/JBHI.2022.3167256
4. Samara G, Aljaidi M, Alazaidah R, Qasem MH, Hassan M, Al-Milli N, Al-Batah MS, Kanan M (2023) A comprehensive review of machine learning-based intrusion detection techniques for IoT networks. In: Artificial intelligence, internet of things, and society 5.0, pp 465–473. https://doi.org/10.1007/978-3-031-43300-9_38
5. Khan IA, Keshk M, Pi D, Khan N, Hussain Y, Soliman H (2022) Enhancing IIoT networks protection: a robust security model for attack detection in internet industrial control systems. Ad Hoc Netw. https://doi.org/10.1016/j.adhoc.2022.102930
6. Vaiyapuri T, Sbai Z, Alaskar H, Alaseem NA (2021) deep learning approaches for intrusion detection in IIoT networks—opportunities and future directions. Int J Adv Comput Sci Appl 12:86–92. https://doi.org/10.14569/IJACSA.2021.0120411
7. Zhang L, Jiang S, Shen X, Gupta BB, Tian Z (2021) PWG-IDS: an intrusion detection model for solving class imbalance in IIoT networks using generative adversarial networks. http://arxiv.org/abs/2110.03445
8. Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A (2020) TON-IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access 8:165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862
9. Kasongo SM (2021) An advanced intrusion detection system for IIoT based on GA and tree based algorithms. IEEE Access 9:113199–113212. https://doi.org/10.1109/ACCESS.2021.3104113
10. Awotunde JB, Chakraborty C, Adeniyi AE (2021) Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wirel Commun Mob Comput. https://doi.org/10.1155/2021/7154587
11. Yao H, Gao P, Zhang P, Wang J, Jiang C, Lu L (2019) Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection. IEEE Netw 33:75–81. https://doi.org/10.1109/MNET.001.1800479
12. Nti IK, Adekoya AF, Narko-Boateng O, Somanathan AR (2022) Stacknet based decision fusion classifier for network intrusion detection. Int Arab J Inf Technol 19:478–490. https://doi.org/10.34028/iajit/19/3A/8
13. Surakhi O, García A, Jamoos M, Alkhanafseh M (2022) The intrusion detection system by deep learning methods: issues and challenges. Int Arab J Inf Technol 19:501–513. https://doi.org/10.34028/iajit/19/3A/10
14. Eid AM, Soudan B, Nassif AB, Injadat MN (2024) Comparative study of ML models for IIoT intrusion detection: impact of data

preprocessing and balancing. Neural Comput Appl 36:6955–6972. https://doi.org/10.1007/s00521-024-09439-x
15. Vulfin AM, Vasilyev VI, Kuharev SN, Homutov EV, Kirillova AD (2021) Algorithms for detecting network attacks in an enterprise industrial network based on data mining algorithms. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/2001/1/012004
16. He Y, Mendis GJ, Wei J (2017) Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans Smart Grid 8:2505–2516. https://doi.org/10.1109/TSG.2017.2703842
17. AL-Hawawreh M, Moustafa N, Sitnikova E (2018) Identification of malicious activities in industrial internet of things based on deep learning models. J Inf Secur Appl 41:1–11. https://doi.org/10.1016/j.jisa.2018.05.002
18. Li Y, Xu Y, Liu Z, Hou H, Zheng Y, Xin Y, Zhao Y, Cui L (2020) Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Meas J Int Meas Confed 154:107450. https://doi.org/10.1016/j.measurement.2019.107450
19. Teixeira MA, Zolanvari M, Khan KM, Jain R, Meskin N (2021) Flow-based intrusion detection algorithm for supervisory control and data acquisition systems: a real-time approach. IET Cyber Phys Syst Theory Appl 6:178–191. https://doi.org/10.1049/cps2.12016
20. Zolanvari M, Teixeira MA, Gupta L, Khan KM, Jain R (2019) Machine learning-based network vulnerability analysis of industrial internet of things. IEEE Internet Things J 6:6822–6834. https://doi.org/10.1109/JIOT.2019.2912022
21. Zolanvari M, Ghubaish A, Jain R (2021) ADDAI: anomaly detection using distributed AI. In: ICNSC 2021—18th IEEE international conference on networking, sensing and control, Industry 4.0 AI. https://doi.org/10.1109/ICNSC52481.2021.9702157
22. Alani MM, Damiani E, Ghosh U (2022) DeepIIoT: an explainable deep learning based intrusion detection system for industrial IOT. In: Proceedings—2022 IEEE 42nd international conference on distributed computing systems workshops, ICDCSW 2022, pp 169–174. https://doi.org/10.1109/ICDCSW56584.2022.00040
23. Zhang YP, Zhang LN, Wang YC (2010) Cluster-based majority under-sampling approaches for class imbalance learning. In: Proceedings—2010 2nd IEEE international conference on information and financial engineering, ICIFE 2010, pp 400–404. https://doi.org/10.1109/ICIFE.2010.5609385
24. Xiao Y, Xiao X (2019) An intrusion detection system based on a simplified residual network. Information. https://doi.org/10.3390/info10110356
25. Hussain F, Abbas SG, Husnain M, Fayyaz UU, Shahzad F, Shah GA (2020) IoT DoS and DDoS attack detection using RresNet. In: Proceedings—2020 IEEE 23rd international multitopic conference, INMIC 2020. https://doi.org/10.1109/INMIC50486.2020.9318216
26. Li Z, Qin Z, Huang K, Yang X, Ye S (2017) Intrusion detection using convolutional neural networks for representation learning. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 10638 LNCS, pp 858–866. https://doi.org/10.1007/978-3-319-70139-4_87
27. Nassif AB, Elnagar A, Shahin I, Henno S (2021) Deep learning for Arabic subjective sentiment analysis: challenges and research opportunities. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106836
28. Teixeira MA, Gupta L, Khan KM, Machine RJ (2021) WUSTL-IIOT-2021 dataset for IIoT cybersecurity research, vol 6, pp 11–12. https://www.cse.wustl.edu/∼jain/iiot2/index.html
29. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military communications and information systems conference, MilCIS 2015—proceedings. https://doi.org/10.1109/MilCIS.2015.7348942
30. Soudan B, Dandachi FF, Nassif AB (2022) Attempting cardiac arrest prediction using artificial intelligence on vital signs from electronic health records. Smart Health. https://doi.org/10.1016/j.smhl.2022.100294
31. Manderna A, Kumar S, Dohare U, Aljaidi M, Kaiwartya O, Lloret J (2023) Vehicular network intrusion detection using a cascaded deep learning approach with multi-variant metaheuristic. Sensors 23:8772. https://doi.org/10.3390/s23218772
32. Nassif AB, Soudan B, Azzeh M, Attilli I, Almulla O (2021) Artificial intelligence and statistical techniques in short-term load forecasting: a review. Int Rev Model Simul 14:408–430. https://doi.org/10.15866/iremos.v14i6.21328
33. Mesevage TG (2021) Data cleaning steps & process to prep your data for success. MonkeyLearn
34. Tableau (2022) Data cleaning: definition, benefits, and how-to. Tableau. https://www.tableau.com/learn/articles/what-is-data-cleaning
35. Al-Mimi H, Hamad NA, Abualhaj MM, Sh. Daoud M, Al-Dahoud A, Rasmi M (2023) An enhanced intrusion detection system for protecting HTTP services from attacks. Int J Adv Soft Comput Its Appl 15:67–84. https://doi.org/10.15849/IJASCA.230720.05
36. Elizar E, Zulkifley MA, Muharar R, Zaman MHM, Mustaza SM (2022) A review on multiscale-deep-learning applications. Sensors. https://doi.org/10.3390/s22197384
37. Liu X, Tang Z, Yang B (2019) Predicting network attacks with CNN by constructing images from NetFlow data. In: Proceedings—5th IEEE international conference on big data security on cloud, BigDataSecurity 2019,.5th IEEE international conference on high performance and smart computing, HPSC 2019, 4th IEEE international conference on intelligent data and security, pp 61–66. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00022
38. Eid AM, Nassif AB, Soudan B, Injadat MN (2023) IIoT network intrusion detection using machine learning. In: 2023 6th international conference on intelligent robotics and control engineering. IEEE, pp 196–201.https://doi.org/10.1109/IRCE59430.2023.10255088
39. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357. https://doi.org/10.1613/jair.953
40. Pontes FJ, Amorim GF, Balestrassi PP, Paiva AP, Ferreira JR (2016) Design of experiments and focused grid search for neural network parameter optimization. Neurocomputing 186:22–34. https://doi.org/10.1016/j.neucom.2015.12.061