Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2024.051779

ARTICLE

Intrusion Detection System for Smart Industrial Environments with Ensemble Feature Selection and Deep Convolutional Neural Networks

Asad Raza[1,*], Shahzad Memon[1], Muhammad Ali Nizamani[1] and Mahmood Hussain Shah[2]

[1]Dr. AHS Bukhari Postgraduate Centre of Information and Communication Technology, Faculty of Engineering and Technology, University of Sindh, Jamshoro, 76060, Pakistan

[2]Newcastle Business School, Northumbria University, Newcastle, NE1, UK

*Corresponding Author: Asad Raza. Email: asadraza825@gmail.com

Received: 15 March 2024 Accepted: 10 May 2024 Published: 11 July 2024

ABSTRACT

Smart Industrial environments use the Industrial Internet of Things (IIoT) for their routine operations and transform their industrial operations with intelligent and driven approaches. However, IIoT devices are vulnerable to cyber threats and exploits due to their connectivity with the internet. Traditional signature-based IDS are effective in detecting known attacks, but they are unable to detect unknown emerging attacks. Therefore, there is the need for an IDS which can learn from data and detect new threats. Ensemble Machine Learning (ML) and individual Deep Learning (DL) based IDS have been developed, and these individual models achieved low accuracy; however, their performance can be improved with the ensemble stacking technique. In this paper, we have proposed a Deep Stacked Neural Network (DSNN) based IDS, which consists of two stacked Convolutional Neural Network (CNN) models as base learners and Extreme Gradient Boosting (XGB) as the meta learner. The proposed DSNN model was trained and evaluated with the next-generation dataset, TON_IoT. Several pre-processing techniques were applied to prepare a dataset for the model, including ensemble feature selection and the SMOTE technique. Accuracy, precision, recall, F1-score, and false positive rates were used to evaluate the performance of the proposed ensemble model. Our experimental results showed that the accuracy for binary classification is 99.61%, which is better than in the baseline individual DL and ML models. In addition, the model proposed for IDS has been compared with similar models. The proposed DSNN achieved better performance metrics than the other models. The proposed DSNN model will be used to develop enhanced IDS for threat mitigation in smart industrial environments.

KEYWORDS

Industrial internet of things; smart industrial environment; cyber-attacks; convolutional neural network; ensemble learning

1 Introduction

Smart industrial environments use industrial revolution (IR) 4.0 technologies to foster their operations. These technologies have transformed industries by fostering a shift towards a more technology-driven approach for the enhancement of industrial production and overall efficiency. In today's fiercely competitive global markets, where businesses strive for success and recognition,

implementing innovative devices and systems is no longer optional–it is necessary to achieve increased production efficiency, productivity, and quality [1].

IR 4.0 focuses on individual activity tracing and management and is supported by Industrial Internet of Things (IIoT) technology for information management. IR 5.0 is also focused on smart manufacturing and urges the co-working of machines and humans with sustainability [2]. Industrial control systems, smart manufacturing, and transportation systems are some examples of smart industrial environments. The transition of industries towards smart manufacturing in developed countries is faster than in developing countries due to its relationship with the economy, because the slow adaption of emerging technologies may reduce economic activity, which consequently will stress industry and the economy. It requires a qualified workforce to manage critical smart industrial infrastructures [3].

In contrast to traditional industries that rely on human resources to increase production, smart industrial environments consist of a network of interconnected Industrial Internet of Things (IIoT) devices, routers, and machines that work together to collect, transfer, store, process, and control data, enabling remote monitoring of overall industrial operations. However, this continuous internet connectivity exposes these smart objects to various risks and threats. Smart environments often rely on third-party applications and services, which can introduce vulnerabilities. Threat actors could exploit those weaknesses to disrupt critical industrial processes, leading to significant financial losses. The cyber assets within these environments are susceptible to a range of emerging cyber-attacks, such as botnets, denial of service (DoS), Distributed Denial of Service (DDoS), Man in The Middle (MiTM), ransomware, and spyware [4]. Fig. 1 illustrates the threat model of an industrial smart environment.

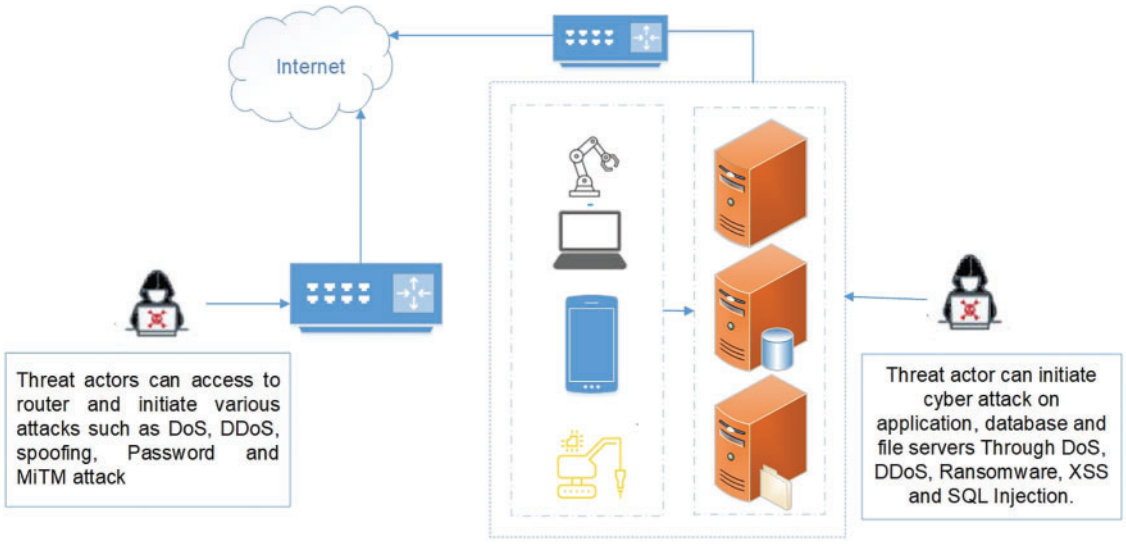

Figure 1: Smart industrial environment threat model

The intrusion detection system (IDS) is a security software that is used to detect intrusion. An IDS can be used to detect unauthorized access to the system or malicious traffic in the network. An IDS can be deployed at nodes, network hubs, switches, or routers. Traditional IDS struggle against emerging threats because their reliance on a known signature leaves them blind to novel attacks. This necessitates the development of advanced IDS, utilizing ML, particularly DL, which can learn from a vast amount of diverse security data generated from smart industrial environments which provide

a rich training ground for such DL-based IDS. As data volume increases, the model accuracy also increases in predicting both normal and attack events. Therefore, harnessing the latest ML techniques, especially DL, is crucial for shaping an effective IDS capable of detecting emerging attacks [5].

Ensemble learning is an ML technique to boost model performance by combining multiple model predictions. The DL model can mitigate industrial smart environments by combining multiple DL models to achieve a better model. Different work has been undertaken to develop an IDS with individual and ensemble ML and DL models, e.g., the Convolutional Neural Network (CNN) and Few-shot learning (FSL) [6], Recurrent neural network (RNN) [7], ensemble CNN, RNN and multilayer perceptron (MLP) [8], Gated recurrent unit (GRU), and Long Short-Term Memory (LSTM) [9]. These researchers have used individual ML, DL, and an ensemble of ML, and DL models; however, they have low accuracy, so as a result, they have a low detection rate so there is space to further improve IDS performance. We have thus proposed a deep stacked neural network (DSNN) based IDS for smart industrial environments which consists of two CNNs as base learners, and Extreme Gradient Boosting (XGB) as the meta learner.

IIoT devices generate huge security logs that contain heterogeneous data and can be used to develop security systems [10]. Standard public IIoT datasets are also available, such as KDDCUP99 and DARPA which are used for IDS development [11]. These datasets are outdated and do not contain emerging network events, so such datasets are not viable in the era of IIoT. TON_IoT is a next-generation dataset for cyber-physical systems (CPS) and IIoT, which contain heterogeneous telemetry data. The dataset contains cyber-attack events that affect network and application layers such as MiTM, DoS, and DDoS from attacks at the network layer. Injection, password, and Cross-site scripting (XSS) attack the application layers. Ransomware, backdoor, injection, password, and scanning attack both the application and network layer [12].

This paper contributes to knowledge in the domain of DL-based IDS for threat mitigation, as summarized below:

- This research proposes a new ensemble feature selection method, consisting of two techniques: zero drop, and information gain. The zero drop selects the features considering zero values, and the information gain technique selects significant values from the dataset.
- A novel ensemble stacking-based DL model has been proposed for binary classification, called the Deep Stacked Neural Network (DSNN). The DSNN model consists of two CNN as base learners, and XGB as meta learners.
- The proposed model was evaluated with various performance metrics. such as accuracy, recall precision, F1-score, false positive rate (FPR), false negative rate (FNR), and Matthews Coefficient Correlation (MCC). The model was also compared with similar models proposed in the literature.

The paper is further organized as follows: Section 2 provides a literature review of existing related work. Section 3 describes the methodology used to develop the proposed IDS, including the TON_IoT dataset, preprocessing techniques, and the proposed ensemble model. Section 4 of this paper provides experimental results and discussions. Finally, Section 5 presents the conclusion and future work.

2 Related Work

In this section, we review various studies related to secure smart environments using ML and DL based IDS. Saheed et al. proposed LSTM and modified genetic algorithm (MGA) for IDS development to protect internet of things (IoT) networks, called IoT defender. The authors used genetic

algorithm (GA) for feature selection and MGA for fine tuning of the LSTM model. The proposed approach was evaluated with BoT-IoT dataset and experimental results showed that the proposed model achieved 99.41% accuracy [13]. LSTM based IDS [14] was proposed by Elsayed et al. to improve IDS performance. The proposed IDS was evaluated with TON_IOT and InSDN datasets. The proposed IDS achieved 96.35%, 96% and 98.4% accuracy, detection rate and precision for TON_IoT, and 99.73%, 98.6% and 98.9% for the InSDN dataset. IDS performance metrics on TON_IoT dataset need further improvement. An explainable DL based framework to improve intrusion detection in critical IoT networks was proposed in [15]. The TON_IoT dataset was used to assess the proposed framework and achieved 99.15% accuracy and a 98.83% F1-score in attack detection. The attack detection performance could be better. The authors, however, did not handle the class imbalance learning problem. The attack detection performance could be further bettered with ensemble learning.

In [16], the authors proposed the deep reinforcement learning (DRL) model, which consists of a feed forward neural network for DL model and Q-learning for reinforcement learning (RL). The experimental results showed that DRL model achieved 91.4% accuracy, 90.2% recall and 92.8% precision. An IDS with DL method for vehicular networks was proposed in [17]. The authors trained IDS with KDDCUP99 and CICIDS 2018 datasets and evaluated them with a testbed dataset created from network simulation. The experimental results showed that the proposed model achieved an accuracy of 99.57%; however, the proposed model can be further evaluated with TON_IOT, and UNSW-NB15 dataset, and their performance can be enhanced with the ensemble stacking of DL models. In [18], three DL models, CNN, LSTM and hybrid ensemble CNN+LSTM, were proposed for intrusion detection in the IIoT network. The proposed model was evaluated with UNSW-NB15 and X-IIoTID datasets. Experimental results showed that CNN+LSTM achieved a binary and multiclass classification accuracy of 93.21% and 92.9% on UNSW-NB15, and 99.84% and 99.80% on X-IIoTID. The proposed model accuracy on UNSW-NB15 can be enhanced with ensemble stacking of CNN and LSTM. In [19], the researchers proposed a LSTM and auto-encoder (LSTM-AE) for the development of the network IDS. The proposed model was evaluated with CICIDS-2017 and CSE-CICDIS-2018 datasets and compared with CNN and deep neural network (DNN). The experimental results showed that LSTM-AE achieved 99.99%, and 99.10% accuracy on CICIDS-2017 and CSE-CICIDS-2018.

An ensemble classifier with ML techniques was proposed by [20] to detect and prevent botnet attacks. The Cyber Clean Center (CCC) dataset was used for evaluation, and experimental results showed that the ensemble model achieved 94.08% accuracy, 86.5% sensitivity, 85.68% specificity and 78.24% F1-score. INformation FUsion and Stacking Ensemble (INFUSE) was proposed for the network IDS [21]. It consists of support vector machine (SVM), k-nearest neighbors (KNN), decision tree (DT), random forest (RF) and Adaboost as base classifiers, and a 6-layer fully connected artificial neural network (ANN) as meta classifier. The proposed model was evaluated with NSL-KDD and results showed that the ensemble model achieved 91.6%, 94% and 91% for accuracy, recall and F1-score. Network IDS was proposed with the improved binary manta ray foraging (BMRF) optimization algorithm for feature selection, and RF for classification. It was evaluated with NSL-KDD and CICIDS-2017. The proposed model achieved 99.3%, which can be further improved [22]. ML based IDS was developed and evaluated with TON_IoT by Gad et al., using the chi-square technique for feature selection and Synthetic Minority Oversampling Technique (SMOTE) for the class imbalance problem. XGB achieved 99.1%, 98.4% and 99.1% accuracy, precision, and recall; however, performance needs improvement [23]. The attack detection system for vehicular networks was proposed by Sharma et al. and was evaluated with the TON_IoT dataset and compared with the RF, Naïve Bayes and KNN models. The KNN model achieved a high accuracy of 98.2%, which can be improved [24].

Real time IDS was proposed with DNN and evaluated with NSL-KDD. The experimental results showed that proposed IDS achieved 81% accuracy, with 96%, 70%, and 81% for precision, recall, and F1-score. The proposed IDS performance can be enhanced with ensemble stacking [25]. In [26], an IDS was proposed with stacking of five ANN models for the Supervisory Control and Data Acquisition (SCADA) system security and was evaluated with a real time dataset. The experimental results showed that proposed stacked DL model performed better than baseline individual DL and ML models. This work can be extended to develop stacked CNN model. The ensemble model was proposed for botnet attack detection in IoT traffic [27]. The model was evaluated with three datasets: IoT network traffic, KDDCUP99, and TON_IoT, and achieved 97.9% accuracy for binary classification to detect normal and malicious network traffic. A lightweight dense random neural network (DaRaNN) for intrusion detection in IoT networks was discussed in [28]. The proposed DaRaNN was evaluated with TON_IoT and achieved 99.14% accuracy. The ensemble stacking of ML models was proposed in [29] to detect attacks in an IIoT environment. The proposed model was evaluated on NSL-KDD and UNSW-NB15 datasets. The accuracy, precision, sensitivity, specificity and F1-score were used to evaluate the model performance score, and experimental results showed that the ensemble stacking model achieved accuracy of 95.15%, precision 96.47%, sensitivity 95.93%, specificity 93.76%, and F1-score 96.20% on the UNSW-NB15 dataset. In [30], anomaly-based IDS was proposed with CNN for threat mitigation in IoT networks. The proposed IDS was evaluated with NID and BoT-IoT datasets, and experimental results showed that it accomplished 99.51% and 92.85% accuracy on NID and BoT-IoT.

CNN and DT classifiers were proposed for intrusion detection. CNN was used to select features and a DT classifier was used for cyber-attack classification. NSL-KDD was used for evaluation of the proposed model and achieved 99.49% accuracy [31]. Deep random neural network (DRaNN) was proposed in [32] for attack detection in IIoT environments. The authors used fusion particle swarm optimization (PSO) with the sequential quadratic programming (SQP) technique for optimal training of DRaNN. The DRaNN was evaluated with three datasets and achieved accuracy of 99.57%, 99.12% and 98.64% on TON_IoT, UNSWNB15 and DS2OS datasets. LSTM and Deep Feed Forward Neural Network (DFNN) were proposed for abnormal traffic detection in IoT networks. UNSW-NB15 was used for the evaluation of the proposed models and achieved accuracy of 90.66% and 64.12% for LSTM and DFNN [33]. The authors [34] proposed CNN based IDS and evaluated with CSE-CIC-IDS2018. The authors used SMOTE for dataset augmentation and achieved 98.8% accuracy, 98.1% recall and 99.5% precision. The proposed CNN model accuracy and recall can be further enhanced with the ensemble stacking of CNNs. The CNN based attack detection system was developed for the IoT environment and evaluated with NSL-KDD dataset. The experimental results showed that the proposed system achieved accuracy of 99.3% for binary and 98.2% for multiclass classifications. The proposed system accuracy can be further enhanced with ensemble learning and can be evaluated with the TON_IoT dataset to detect emerging attacks [35]. The authors proposed ML based IDS for attack detection in vehicular ad hoc networks and evaluated with TON_IoT. The experimental results showed that XGB achieved high performance results with 99.0% accuracy; however, that accuracy can be further improved with ensemble models [36]. DNN based IDS was proposed by Aleesa et al. The proposed model was evaluated with enhanced UNSW-NB15 dataset for binary and multiclass classification. The experimental results showed that the proposed IDS achieved 99.26% and 99.59% accuracy for binary and multiclass classification. However, with stand-alone CNN and ensemble stacking, an improved IDS can be developed [37]. Ensemble stacking of CNN, RNN as base classifier and DNN as meta classifier was proposed for malware detection in IoT environments. The proposed ensemble model called malware threat hunting model based on advanced ensemble learning (MTHAEL) was achieved 99.98% accuracy on the IoT malware dataset. The MTHAEL can

be evaluated on a large scale IoT dataset and can be further extended for the development of network IDS for IoT/IIoT environments [38]. The authors of [39] proposed BAT-MC model which consists of Bidirectional Long Short-Term Memory (BLSTM), multiple convolutional layers and attention mechanism. The proposed model was evaluated with NSL-KDD, and experimental results showed that the BAT-MC achieved accuracy 84.25%, which can be further improved with ensemble learning. NSL-KDD does not contain emerging attack events, so the BAT-MC model needs further evaluation with the latest network datasets, such as TON_IoT. The researchers [40] developed an IDS with a Random Neural Network (RaNN) for IIoT. The proposed RaNN was evaluated with UNSW-NB15 dataset and experimental results showed that RaNN achieved 99.54% accuracy. The proposed work can be extended in various directions, such as its performance can be further enhanced with ensemble stacking of RaNN and CNN. It also needs evaluation with another emerging IIoT dataset, such as the TON_IoT dataset. The authors [41] used various supervised ML algorithms such as DT, KNN, RF, SVM and logistic regression. RF achieved 99.99% accuracy, 96.81% MCC and 97.44% sensitivity; however, using only the accuracy metric for imbalanced datasets is insufficient to evaluate the model's performance; MCC and sensitivity scores can be improved with DL models and ensemble learning. The authors proposed an ensemble of SVM, instance-based learning algorithms (IBK) and MLP. It was evaluated with the NSL-KDD dataset and experimental results showed that the proposed ensemble model achieved 98.24% accuracy, which can be improved with ensemble learning of DL models, to enhance accuracy [42]. Table 1 summarizes the techniques and performances of various techniques discussed in this section.

Table 1: Summary of related work

Proposed techniques	Dataset	Model performance metrics (%)
LSTM, MGA [13]	BoT-IoT	• Accuracy: 99.41
LSTM [14]	TON_IoT	• Accuracy: 96.35 • Detection rate: 98.6 • Precision: 98.9
DL [15]	TON_IoT	• Accuracy: 99.15 • F1-score: 9.83
RL [16]	NSL-KDD	• Accuracy: 91.4 • Recall: 90.2 • Precision: 92.8
DL [17]	KDDCUP99, CICIDS 2018	• Accuracy: 99.57
CNN-LSTM [18]	UNSW-NB15	• Accuracy: 93.21
LSTM-AE [19]	CSE-CICDIS-2018	• Accuracy: 99.10%
Ensemble classifier [20]	Cyber Clean Center	• Accuracy: 94.08 • Sensitivity: 86.5 • Specificity: 85.68 • F-score: 78.24
SVM, KNN, DT, RF, Adaboost, FNN [21]	NSL-KDD	• Accuracy: 91.6 • Recall: 94 • F1-score: 91.6
RF [22]	NSL-KDD	• Accuracy: 99.3

(Continued)

Table 1 (continued)

Proposed techniques	Dataset	Model performance metrics (%)
XGB [23]	TON_IoT	• Accuracy: 99.3 • Precision: 98.4 • Recall: 99.1
KNN [24]	TON_IoT	• Accuracy: 98.2
DNN [25]	NSL-KDD	• Accuracy: 81 • Precision: 96 • Recall: 70 • F1-score: 81
DL [26]	Real time dataset	• Accuracy: 97.36 • Recall: 98.46 • Precision: 97.81 • F1-score: 98.31
Ensemble learning [27]	TON_IoT	• Accuracy: 97.9
DaRaNN [28]	TON_IoT	• Accuracy: 99.14
Ensemble stacking [29]	NSL-KDD, UNSW-NB15	• Accuracy: 95.15 • Precision: 96.47 • Sensitivity: 95.93 • Specificity: 93.76 • F1-score: 96.20
CNN [30]	NID	• Accuracy: 99.51
CNN-DT [31]	NSL-KDD	• Accuracy: 99.49
DRaNN_PSO [32]	TON_IoT	• Accuracy: 99.57
LSTM [33]	UNSW-NB15	• Accuracy: 90.66
CNN [34]	CSE-CIC-IDS2018	• Accuracy: 98.8 • Recall: 98.1 • Precision: 99.5
CNN [35]	NSL-KDD	• Accuracy: 99.3
XGB [36]	TON_IoT	• Accuracy: 99.0
ANN [37]	UNSW-NB15	• Accuracy: 99.26
CNN, RNN, DNN [38]	IoT malware dataset	• Accuracy: 99.98 • Recall: 99.97 • Precision: 99.96 • F1-score: 99.94
BLSTM [39]	NSL-KDD	• Accuracy: 84.25
RaNN [40]	UNSW-NB15	• Accuracy: 99.54
RF [41]	Testbed dataset	• Accuracy: 99.99 • MCC: 96.81 • Sensitivity: 97.44
IG-PCA [42]	NSL-KDD	• Accuracy: 98.24 • Recall: 98.2

While existing research explores IDS using individual or an ensemble of ML/DL models, there remains a need to enhance their performance metrics, such as accuracy and recall. This study addresses this gap by proposing a novel stacking ensemble approach that combines two CNNs with an XGB model. To the best of our knowledge, this specific combination has not previously been explored in the literature.

3 Methodology

In this section, an ensemble stacking based model called Deep Stacked Neural Network (DSNN) was proposed to develop IDS for a better attack detection rate and accuracy in a smart industrial environment. The proposed DSNN based IDS architecture, CNN base learners' architecture and its hyperparameters are discussed in this section. The TON_IoT dataset characteristics and preprocessing steps are also described.

3.1 Proposed Ensemble Stacking Based Intrusion Detection System

An ensemble stacking based IDS was proposed with DSNN for of normal and attack classification in network traffic of smart industrial environments. We have proposed heterogeneous stacking model which consists of two different CNNs as base learners, including CNN-1 and CNN-2, which consist of 4 and 6 trainable layers. We used XGB as the meta learner, which is an implementation gradient boosted decision tree and a widely used algorithm, due to its efficiency to achieve better performance in large datasets. The DSNN was developed with k-fold cross validation technique and evaluated with TON_IoT testing subset. We used various performance metrics for model evaluation, comparing them with baseline and existing models. The proposed IDS with DSNN architecture are shown in Fig. 2.

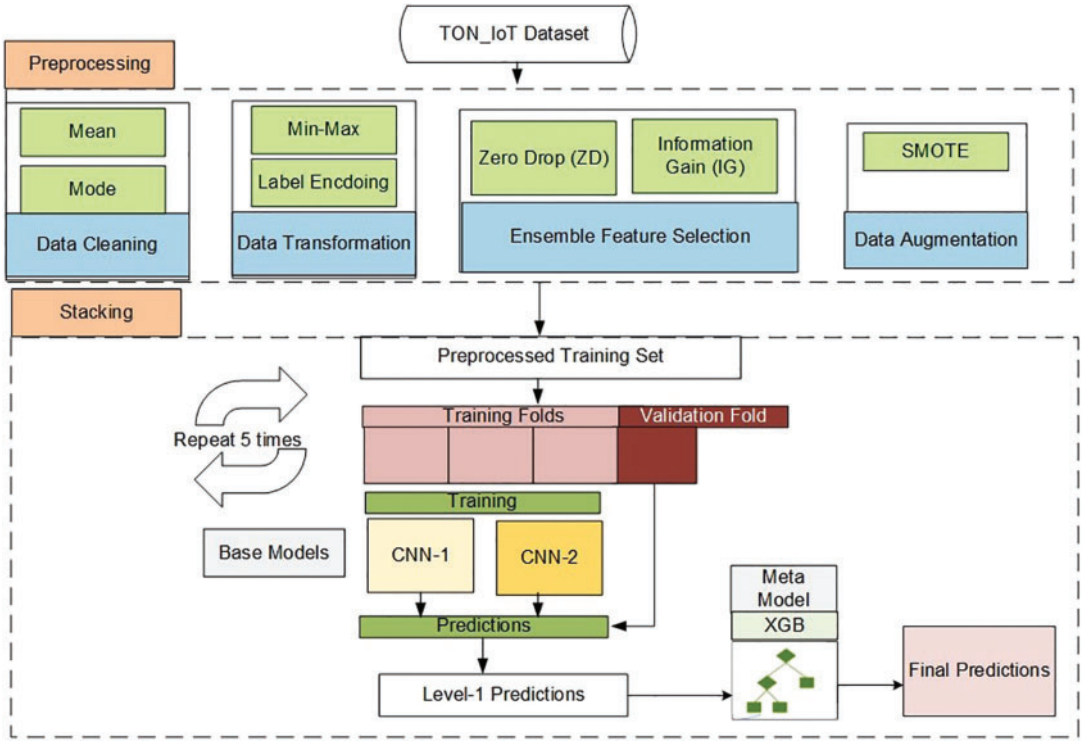

Figure 2: Proposed IDS with DSNN architecture for a smart industrial environment

Hyperparameters of the proposed DSNN are given in Table 2.

Table 2: The proposed DSNN and its base learner hyperparameters

Hyperparameter	Value
No. of base learners	2
Base learners	CNN-1 with 9 layers and CNN-2 with 13 layers
Meta learner	XGB
Activation functions	ReLU, Sigmoid
Dropout rate	0.5
Optimizer	Adam
Learning rate	0.001
No. of epochs	100
Loss function for binary classification	Binary_crossentropy

3.2 Convolutional Neural Network (CNN) as Base Learner

CNN is a prominent DL algorithm primarily used for computer vision problems. We proposed two CNN models with different architecture as the base learner for DSNN. We proposed both CNNs after an iterative process of hyperparameter optimization of CNNs. Both CNNs' hyper parameters are the same, except for the number of layers. The CNN-1 consists of a total of 9 layers, of which 4 are trainable and 3 are non-trainable. The CNN-2 model consists of a total of 13 layers, of which 6 are trainable and 7 are non-trainable. We optimized CNNs with an Adam optimizer of learning rate 0.001 and used ReLU and Sigmoid activation functions. Binary Cross Entropy loss function was used to measure the loss between actual and predicted labels. The CNN architecture is illustrated in Figs. 3 and 4, where $X_1, X_2, \ldots, X_n$ represents features in input layer, and 0/1 represents normal/attack class.

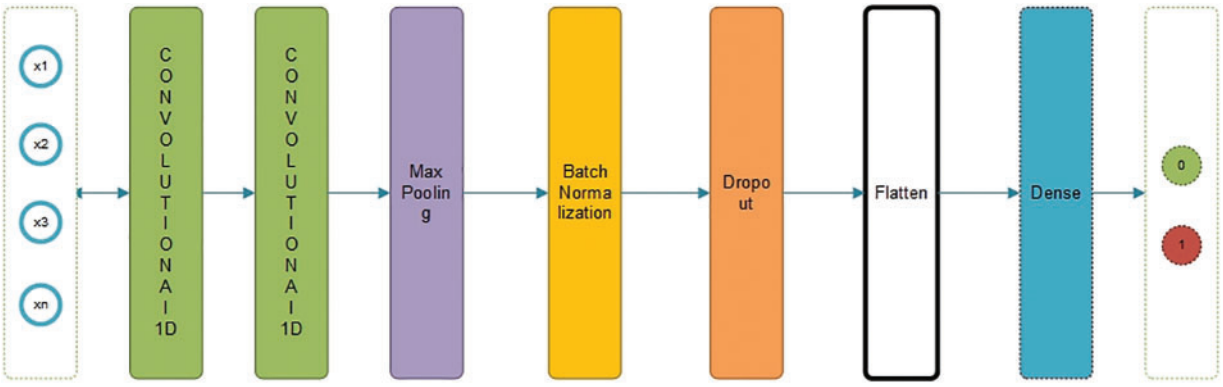

Figure 3: Proposed CNN-1 9-layer architecture

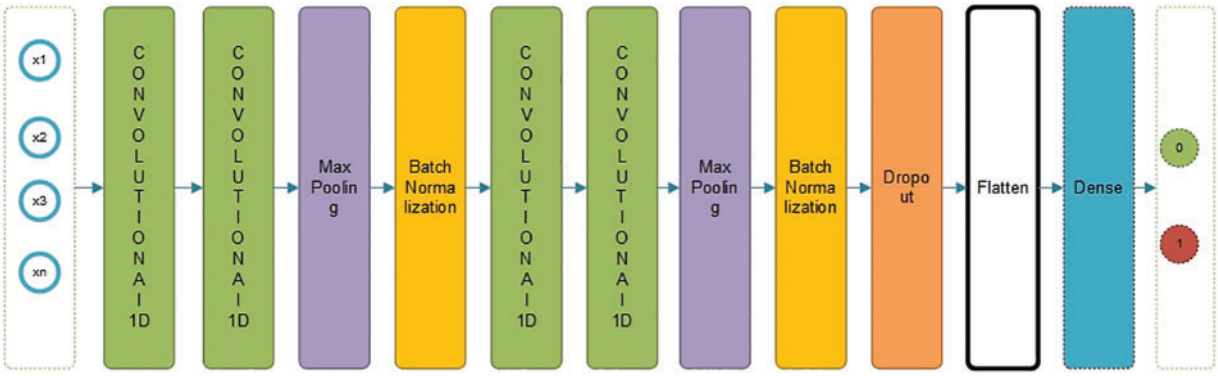

Figure 4: Proposed CNN-2 13-layer architecture

3.3 Overfitting in Development of DSNN

We resolved overfitting issues in DSNN development in various ways. We proposed the ensemble feature selection method to select important features which reduces the overfitting problem because when training with important features, DSNN demonstrate better results. We also used various regularization layers in base learners, such as max pooling, batch normalization and dropout layers, to reduce overfitting.

3.3.1 Max Pooling Layer

The max pooling layer was used for the down sampling operation to reduce the parameters. Max pooling is used to decrease overfitting, training time and increase model performance.

3.3.2 Batch Normalization Layer

The batch normalization layer was used to normalize the layers' input. It is used between training layers and helps to decrease training time and increases performance using layer normalization.

3.3.3 Dropout Layer

The dropout layer was used to remove neurons for the down sampling operation, which was used to reduce model overfitting. We used a 0.5 dropout value for both the CNN-1 and CNN-2 models.

3.4 Experimental Setup

Our experiments were conducted on a readily available laptop with an Intel Core i5 (5th Gen), processor, 2.30 GHz processor, 8 GB RAM and 750 GB HDD, which provided sufficient processing power for our chosen software stack (Python, libraries including TensorFlow, Keras, scikit-learn, matplotlib, and mlxtend). These libraries offered robust functionalities for data manipulation, model training, evaluation, and streamlining the development of our DL based IDS. We evaluated the system using both the original imbalanced dataset (with all features) and subsequently augmented the dataset with selected features. All the DL models (CNN-1, CNN-2 and DNN) used for developing our proposed DSNN employed the same learning rate, dropout rate, and Adam optimizer. Details of the baseline DL model parameters are provided in Table 3.

Table 3: Experimental design with various parameters for development of proposed DSNN

Model	Parameters	Epochs
DNN	523	100
CNN-1	2464	100
CNN-2	1395	100

3.5 TON_IoT Network Dataset

TON_IoT dataset contains telemetry network traffic data from various sources consisting of emerging cyber-attack events such as MiTM and ransomware, which is why we selected that dataset for our research study. There is a need to be familiar with the dataset because it will be used for training and testing the model. The dataset description is used to explore the data and its features. The dataset consists of 45 features: out of 45, 43 are training features and 2 features are labels, i.e., Normal/Attack. Total events are 461,043 which includes 300,000 normal, while 161,043 are attack events. The graphical representation of class normal and attack examples is given in Fig. 5.

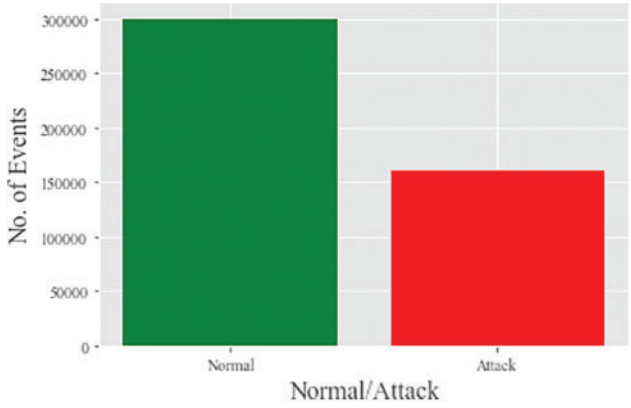

Figure 5: TON_IoT network dataset normal and attack training examples

DL models require enough data for training and subsets for evaluation. We therefore divided the TON_IoT dataset into two subsets: 80% for training and 20% for testing. The training subset was used to train the model which will learn the pattern in the data. The testing subset consists of those records which have never been seen by the trained model. This subset is used to predict unknown attacks for model evaluation. The training and testing subset statistical description is shown in Table 4.

Table 4: Training and testing subset description

TON_IOT characteristic	Subset description	
Dataset type	Training subset	Testing subset
Normal events	240,027	59,973
Attack events	128,807	32,236
Total events	368,834	92,209

In Fig. 5 and Table 4, the TON_IoT network dataset exploration shows that the whole dataset and training subset have imbalanced class distribution. Normal events are higher than attack events, which will result in an imbalanced learning problem in model training. We have tackled the problem of imbalanced learning with the oversample SMOTE technique. The SMOTE technique is further discussed in the data augmentation section.

3.6 Preprocessing

Preprocessing is the major step to prepare data for the development of ML based IDS. If the data is clean, normalized, and relevant it will result in being an efficient model. The preprocessing element consists of data cleaning, feature transformation, feature selection and data augmentation.

3.6.1 Data Cleaning and Transformation

Data cleaning is a process to clean a dataset which may contain insignificant information, such as null and zero values. TON_IoT does not contain null values, but features contain zero and dash "-" values, which we have replaced with zero to make a general value; we then tackled zero with our proposed zero drop feature selection technique and mean/mode. Features with zeros have been replaced with mean if it is a numeric feature; zeros in categorical features were replaced with mode.

Data transformation was used to transform data into useful forms for model learning. Feature normalization and categorization were used to transform the TON_IoT data into a numeric form which was required by the models. Data normalization was used to transform the large numeric value into a smaller form. We used the min-max scalar technique to transform data between 0 and 1. Data normalization helps to reduce training time and increases performance. The Min-Max scalar formula is shown in Eq (1).

$$y = \frac{(X - min)}{(max - min)} \tag{1}$$

where X represents features and min, max represents features range.

Categorical features need an alternative technique to transform the categorical values into the required form. We have used a label encoding technique for encoding categorical values such as service (FTP, HTTP); these values are encoded into numbers (1, 2) where each number represents a feature value.

3.7 Ensemble Feature Selection

Feature selection is an important step to select relevant features and drop irrelevant features. We can estimate features' relevance and select the most important features using the feature selection method. Data with high dimensions can increase model performance; however, it can also increase overfitting, so it can be tackled with feature selection which improves model performance and reduces overfitting. We proposed a novel ensemble feature selection method which consists of two techniques: one is zero drop (ZD), and the second is information gain (IG). The TON_IoT dataset has many features which contain important information and may contain zero values. We proposed the ensemble feature selection technique which selects the features with respect to both issues. The first training subset was given the zero-drop technique, which returns a reduced training subset; after that the reduced features training subset was given as the input to IG. The proposed ensemble feature selection method is further illustrated in Fig. 6.

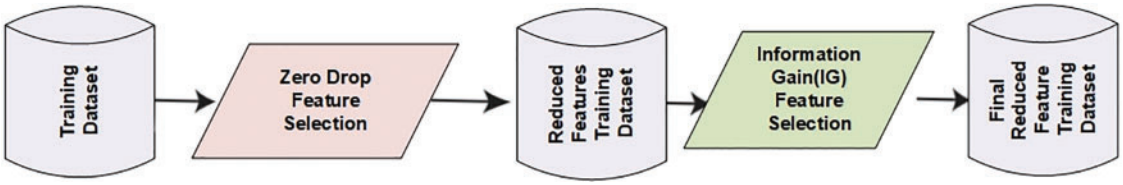

Figure 6: Hybrid ensemble feature selection method

3.7.1 Zero Drop Feature Selection

TON_IoT contains various features with a quantity of zeros. Mostly, feature selection techniques use various threshold scores or association rules to use select features, while features with zero values are removed or replaced with mean or mode. We proposed a novel zero drop technique to deal with zero values and select important features based on zero drop (ZD) value. ZD value is the percentage (%) of the number of zero records with the total number of records in the feature. The equation of zero drop is given in Eq. (2).

$$ZD = \frac{z}{x} \times 100 \tag{2}$$

where z represents % of number of zeros in feature and x represents total number of records in feature.

The zero-drop technique works in that flow, i.e., if the ZD value is less than 95% then it is a significant feature, otherwise it is not significant and will be dropped. So, we dropped those features where the ZD value was greater than 95%. ZD feature selection reduced the training set from 43–22 features. Zero drop feature selection is further illustrated in Fig. 7.

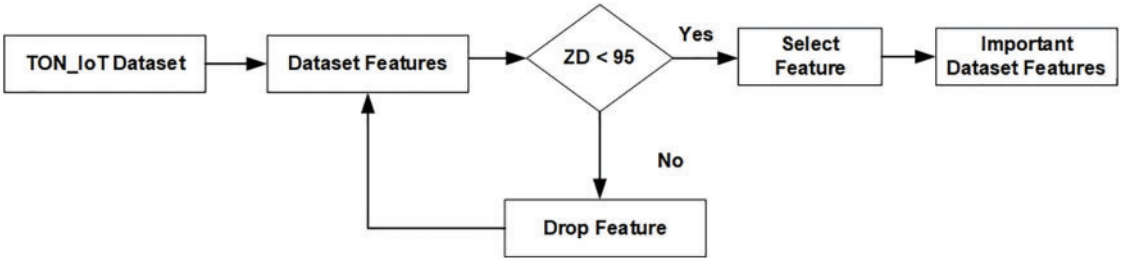

Figure 7: Zero drop feature selection

3.7.2 Information Gain (IG) Feature Selection

IG is a prominent feature selection technique to select the most important features. We have given a reduced training set of zero drop technique to IG. The IG equation is given below:

$$H(X) = \sum_{i}^{k} \mathrm{P}(\mathrm{x_i}) \log \mathrm{P}(\mathrm{x_i}) \tag{3}$$

$$H(Y|X) = \sum_{x,y} \mathrm{P}(\mathrm{x}, \mathrm{y}) \log \left(\frac{1}{P(y|x)} \right) \tag{4}$$

$$IG(Y|X) = H(X) - H(Y|X) \tag{5}$$

Using substitution of Eqs. (3) and (4) in Eq. (5), the IG Eq. (5) will be

$$IG\,(Y|X) = \sum_{i}^{k} \mathrm{P}(\mathrm{x_i}) \log \mathrm{P}(\mathrm{x_i}) - \sum_{x,y} \mathrm{P}\,(\mathrm{x}, \mathrm{y}) \log \left(\frac{1}{P\,(y|x)} \right) \tag{6}$$

where $IG\,(Y|X)$ represents information gain, X represents features, Y represents class, $P(X)$ represents probability, $H(X)$ is entropy and $H(Y|X)$ represents conditional entropy.

We chose the value 0.01 to select the final training subset features for the model's training. IG technique reduced the training subset, which consisted of 17 features, which was the final training subset. The final training subset was achieved using the ensemble feature selection method shown in Fig. 8.

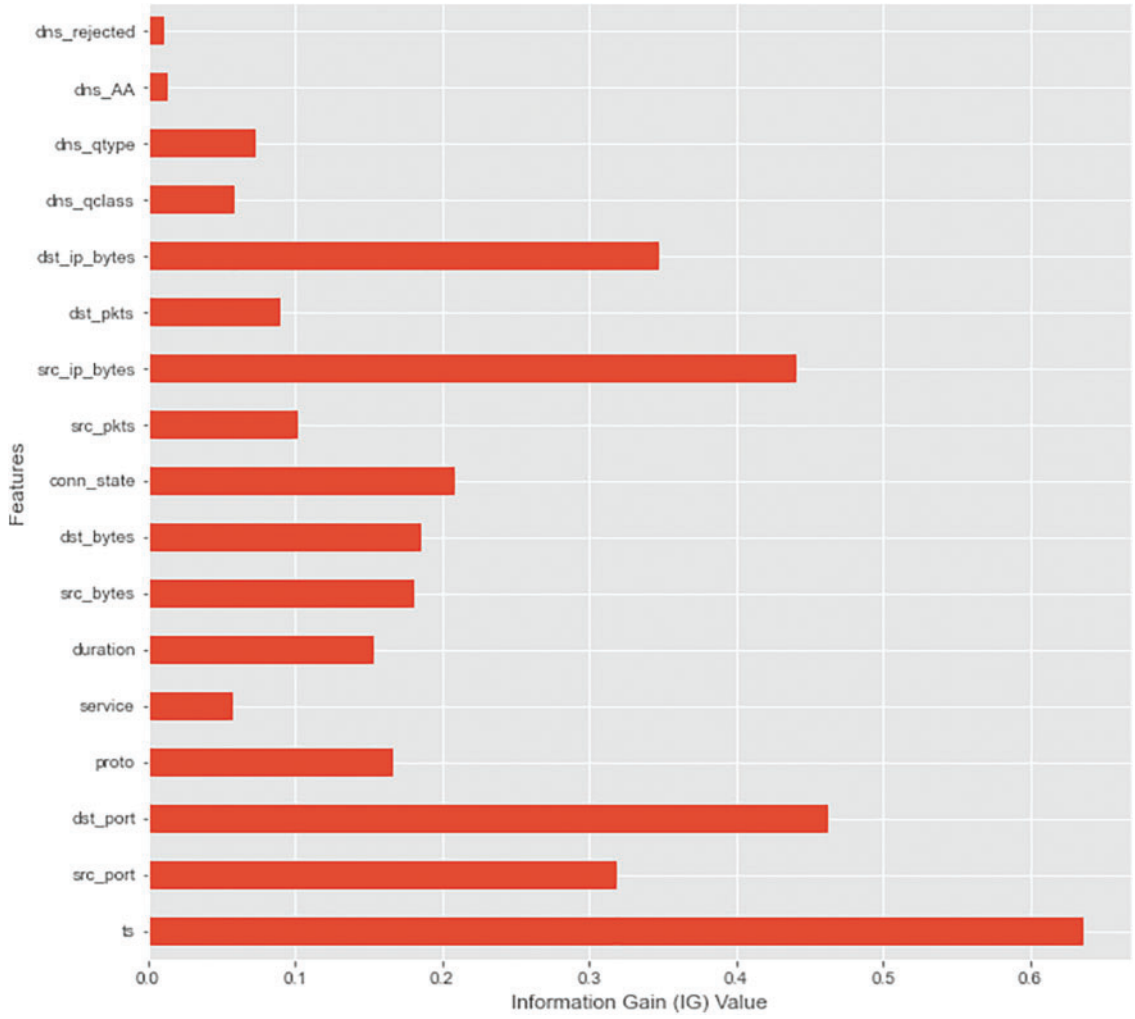

Figure 8: Training subset final 17 features with IG feature importance

3.8 Data Augmentation for Imbalanced Learning

Data augmentation is a method to generate synthetic data samples of imbalanced classes to tackle the problem of imbalanced learning. Imbalanced learning increases bias towards the majority class,

which also reduces model performance. In TON_IoT, normal events are higher than attacks, which indicates that normal is majority while attack is minority class, so the Synthetic Minority Over-Sampling Technique (SMOTE) [43] as an over-sampling technique was proposed to overcome that problem. The SMOTE technique was applied on the training subset, which made both class events balance. Normal and attack events distribution in the original training subset after applying SMOTE is shown in Table 5.

Table 5: Description of TON_IoT training subset

Training subset events	TON_IoT original	TON_IoT after SMOTE
Normal	2,40,027	2,40,027
Attack	1,28,807	2,40,027
Total	3,68,834	4,80,054

3.9 Performance Evaluation Metrics

We used the following metrics to evaluate the performance of the model:

Accuracy is the percentage of correctly classified records over the total number of records.

$$\text{Accuracy} = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{7}$$

Precision is defined as the ratio of the number of true positives (TP) divided by the sum of true positives (TP) and false positives (FP).

$$\text{Precision} = \frac{(TP)}{TP + FP} \tag{8}$$

Recall is calculated as the ratio of the number of TP divided by sum of TP and FN.

$$\text{Recall} = \frac{(TP)}{TP + FN} \tag{9}$$

F1-score is the weighted harmonic mean of the precision and recall and reflects the balance between Precision and Recall.

$$\text{F1} - \text{Score} = 2 \times \frac{(Recall \times Precision)}{Recall + Precision} \tag{10}$$

False positive rate (FPR) is calculated as the ratio of the number of FP divided by the sum of FP and TN.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{11}$$

False negative rate (FNR) is calculated as the ratio of the number of FN divided by the sum of FN and TP.

$$\text{FPR} = \frac{FN}{FN + TP} \tag{12}$$

MCC is used to find the correlation between true and predicted classes.

$$\text{MCC} = \frac{TP \times TN + FP + FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \tag{13}$$

Receiver Operating Characteristic Curve (ROC) is used to graphically evaluate the performance of the model. It is plotted between true positive rate (TPR) and FPR. **Confusion matrix** is a table which shows actual and predicted classes. It consists of TP, TN, FP and FN.

We selected various performance metrics to quantitatively demonstrate model efficiency and to visually depict its performance from different perspectives. Accuracy measures overall correctness, while recall and precision focus on specific portions, like TP and FP rates. Additionally, we utilise the ROC curve to depict the trade-off between TPR and FPR across various thresholds. Furthermore, the confusion matrix provides a detailed breakdown of the model's prediction, offering valuable insights into its strengths and weakness.

4 Results and Discussions

This section presents and discusses binary classification experimental results on the TON_IoT dataset. The proposed DSNN outperformed more in terms of accuracy and other performance metrics than individual ML and DL models. The proposed DSNN model was compared with individual CNN-1, CNN-2, DNN, logistic regression, Naïve Bayes, and XGB. We further compared the DSNN with related work. To evaluate our model's performance, we selected various quantitative metrics such as accuracy, recall, precision, F1-score, MCC, FPR, FNR. Additionally, the ROC curve and confusion matrix is used for visualization of model performance. The evaluation results of various performance metrics of the proposed DSNN and baseline models are shown in Table 6.

Table 6: Performance metrics for evaluation

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	MCC (%)	FPR	FNR
CNN-1	97.01	96.49	95.04	95.76	93.46	0.02705	0.03515
CNN-2	99.31	99.66	98.39	99.02	98.50	0.00875	0.00335
DNN	99.03	98.38	98.85	98.61	97.87	0.00617	0.01616
LR	88.98	96.16	77.65	85.92	78.23	0.14873	0.03837
Naïve Bayes	81.14	94.56	66.09	77.8	75.76	0.26080	0.05100
XGB	97.16	93.42	98.38	95.84	93.76	0.00825	0.06580
DSNN	**99.61**	**99.62**	**99.28**	**99.45**	**99.15**	**0.00390**	**0.00385**

Quantitative analysis of DSNN performance was completed with the confusion matrix (see Fig. 9), which showed that DSNN can detect a more accurate attack with a lower FR and FN. The ROC curve also visually unveiled that DSNN performance was more satisfactory than baseline models, as shown in Fig. 10.

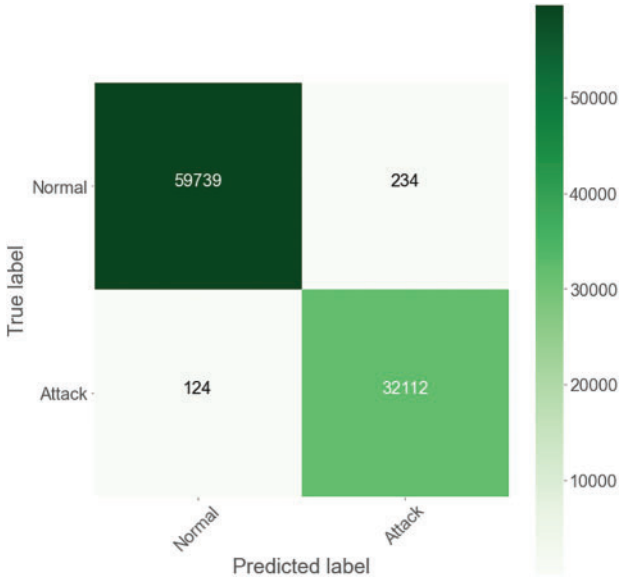

Figure 9: DSNN confusion matrix

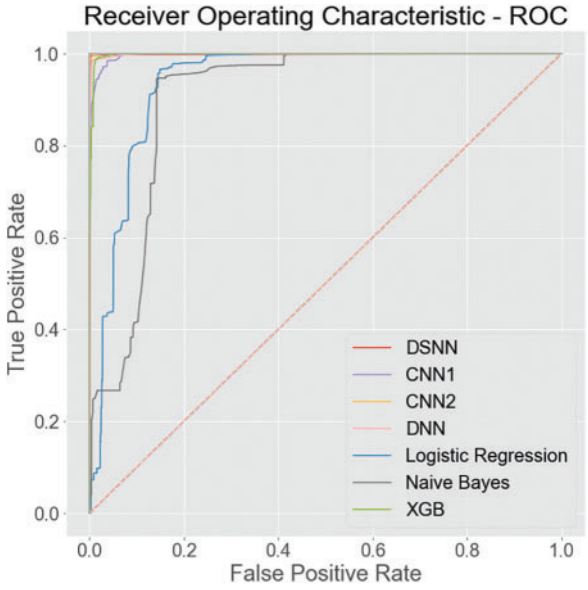

Figure 10: ROC curve of DSNN and baseline models

The DSNN model provides features to select different hyperparameters such as the selection of base and meta learners, increasing the number of base learners which can be selected and optimized with heuristic techniques, i.e., trial and error. We have selected those hyperparameters using the trial-and-error technique and achieved better performing results than on all the baseline DL and ML models. In Table 6, experimental results showed that the DSNN model achieved better accuracy, recall, precision, F1-score, MCC, FPR and FNR scores than all baseline models, except for CNN-2, whose recall and FNR value was 99.66 and 0.00335, which is slightly higher than the DSNN model. Other performance metrics, however, were lower than the DSNN. CNN-2 achieved a good score due to its

deeper structure than CNN-1, and we selected this as the base learner for the DSNN, which enhanced the DSNN performance metrics more than CNN-2 and CNN-1. The DSNN confusion matrix shows that that the quantity of TP and TN are high, and FP and FN are low, therefore the FPR of the DSNN is lowest of all the baseline models. The ROC curve graph also showed that that the DSNN quality was better than the other baseline models. After the proposed DSNN, individual DL models achieved better results and showed that DL models are more efficient in intrusion detection. If we compare ML models, XGB, which is also a type of ensemble ML model, achieved a higher performance than all the baseline ML models. The Naïve Bayes and logistic regression performance is lower, which makes them less effective than all the baseline models.

The DSNN's strong performance can be attributed to several factors. CNNs are adept at learning from a large amount of data, effectively extracting and learning features. Additionally, we employed various regularization layers to the model's parameter size, further enhancing its performance. Ensemble stacking, which combines the predictions of CNN-1 and CNN-2 (both with good individual performances) as inputs to train the XGBoost model (another ensemble method), which contributed to the DSNN's high accuracy. Therefore, the DSNN with ensemble of CNN-1, CNN-2 and XGB, achieves superior performance metrics compared to all the baseline models. A comparison of proposed work with similar research work available in the literature is presented in Table 7.

Table 7: Comparison of proposed work with previous work

Reference	Dataset	Technique(s)	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)	FPR
[7]	NSL-KDD	LSTM	88.13	–	–	99.58	–
[8]	TON_IoT	ANN	94.17	93	94	93.5	–
[9]	TON_IoT	GRU	95.69	96.99	91.23	94.02	–
[14]	TON_IoT	LSTM	96.35	96	98.4	97.19	–
[15]	TON_IoT	DL	99.15	–	–	98.83	–
[16]	NSL-KDD	DNN, QL	91.4	90.2	92.8	–	–
[22]	NSL-KDD	RF	98.8	96.2	96.8	96.5	–
[23]	TON_IoT	XGB	92.2	99.60	95.4	97.5	0.026
[24]	TON_IoT	KNN	98.2	–	–	–	–
[28]	TON_IoT	RaNN	99.14	99.07	99.23	99.27	–
[32]	TON_IoT	DraNN	99.57	99.59	–	–	–
[35]	NSL-KDD	CNN	99.3%	–	–	–	–
[36]	TON_IoT	KNN	98.2	98.9	95.9	97.4	0.023
[42]	NSL-KDD	IG-PCA	98.24	98.2	–	–	0.017
Proposed	**TON_IoT**	**DSNN**	**99.61**	**99.62**	**99.28**	**99.45**	**0.003902**

In Table 7, in comparison with existing related work, it also showed that the proposed DSNN performed better than existing models. The authors used LSTM [7] for IDS and the DL model results are not up to potential as compared to DSNN. ANN was evaluated with TON_IoT for attack detection in IoT network; the ANN efficiency, as compared to DSNN and DSNN, has promising results [8]. The GRU model was proposed for IDS and evaluated with TON_IoT [9], but it achieved lower accuracy, recall, precision, and F1-score. LSTM was again proposed by [14] for intrusion detection, and achieved a lower performance metrics score than our proposed DSNN model. Oseni et al. [15] proposed the DL model, but it achieved better performing results on TON_IoT.

5 Conclusion and Future Work

This paper addresses the problem of improved network IDS for threat mitigation in a smart industrial environment. We investigated the performance of the DSNN model which is ensemble stacking of CNN models and XGB. The proposed model has achieved higher performance results than baseline models. This research study improves the impact of network IDS and ensemble stacking techniques with CNN models, which strengthens industries to deploy such systems to mitigate their networks. CNN as a base learner has increased model accuracy and predicted attacks with a lower FPR than individual DL and ML models. DSNN-based Network IDS can be used inline to detect emerging attacks and reduce threat risks in real-world smart industrial environments. In a real-world smart industrial environment, DSNN-based network IDS can be deployed and trained with real-time data to improve its performance for threat mitigation. There may be challenges such as underfitting and overfitting in real-time implementation of DSNN; however, DSNN provides the opportunity for hyperparameter optimization of CNN base learners, such as adding or removing various layers.

The limitations and future work directions are highlighted in our work. Our proposed zero drop technique is not useful for those datasets that do not contain huge numbers of zero values, so only individual IG techniques can be used for feature selections. DSNN is proposed to detect intrusions at the network level; however, there are also challenges to developing IDS for hosts. The generalized model can be developed with ensemble methods for threat mitigation at both host and network stages. The proposed ensemble feature selection method can be compared with existing feature selection methods on IoT datasets for a comparative evaluation of model performance. The transfer learning method can be used with the individual and ensemble models to further extend our work. Furthermore, graph-based deep learning models, such as graph convolutional and graph attention networks, can be used as a third-base learner to develop IDS. Graph neural networks-based ensemble learning can be used to improve attack detection performance. A federated learning-based approach can be used to develop a more robust ensemble stacking-based IDS. DSNN may not be able to detect a zero-day attack, or most advanced attacks, such as advanced persistent attacks, spyware, or polymorphic attacks, so further datasets can be used to train models for the detection of the most advanced cyber-attacks.

Acknowledgement: This research was made possible thanks to the support and facilitation provided by the Higher Education Commission (HEC) of Pakistan. The authors gratefully acknowledge their contribution.

Funding Statement: This research study was funded by the Higher Education Commission (HEC) of Pakistan.

Author Contributions: All authors have contributed equally to this research study.

Availability of Data and Materials: The dataset used in this study is openly available at: https://research.unsw.edu.au/projects/toniot-datasets.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Castagnoli, G. Büchi, R. Coeurderoy, and M. Cugno, "Evolution of Industry 4.0 and international business: A systematic literature review and a research agenda," *Eur. Manag. J.*, vol. 40, no. 4, pp. 572–589, 2022. doi: 10.1016/j.emj.2021.09.002.

[2] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot co-working," *Procedia Computer Science*, vol. 158, pp. 688–695, 2019. doi: 10.1016/j.procs.2019.09.104.

[3] M. Soori, B. Arezoo, and R. Dastres, "Internet of things for smart factories in Industry 4.0, a review," *Internet Things Cyber-Phys. Syst.*, vol. 3, 2023. doi: 10.1016/j.iotcps.2023.04.006.

[4] M. Humayun, "Industry 4.0 and cyber security issues and challenges," *Turk. J. Comput. Math. Educ.*, vol. 12, no. 10, pp. 2957–2971, 2021.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.

[6] T. Althiyabi, I. Ahmad, and M. O. Alassafi, "Enhancing IoT security: A few-shot learning approach for intrusion detection," *Mathematics*, vol. 12, no. 7, pp. 1055, Mar. 2024. doi: 10.3390/math12071055.

[7] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, 2023. doi: 10.1016/j.comcom.2022.12.010.

[8] A. Jamal, M. F. Hayat, and M. Nasir, "Malware detection and classification in IoT network using ANN," *Mehran Univ. Res. J. Eng. Technol.*, vol. 41, no. 1, pp. 80–91, 2022. doi: 10.22581/muet1982.2201.08.

[9] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, pp. 113249, 2022. doi: 10.1186/s42400-021-00103-8.

[10] N. Magaia, R. Fonseca, K. Muhammad, A. H. F. N. Segundo, A. V. Lira Neto and V. H. C. de Albuquerque, "Industrial internet-of-things security enhanced with deep learning approaches for smart cities," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6393–6405, 2021. doi: 10.1109/JIOT.2020.3042174.

[11] A. M. Al Tobi and I. Duncan, "KDD 1999 generation faults: A review and analysis," *J. Cyber Secur. Tech.*, vol. 2, no. 3–4, pp. 164–200, 1999. doi: 10.1080/23742917.2018.1518061.

[12] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. N. Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, no. September, pp. 165130–165150, 2020. doi: 10.1109/ACCESS.2020.3022862.

[13] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities," *Appl. Soft Comput.*, vol. 155, no. 4, pp. 111434, Apr. 2024. doi: 10.1016/j.asoc.2024.111434.

[14] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Eng. J.*, vol. 14, no. 10, pp. 102211, 2023. doi: 10.1016/j.asej.2023.102211.

[15] A. Oseni *et al.*, "An explainable deep learning framework for resilient intrusion detection in IoT-enabled transportation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1000–1014, 2023. doi: 10.1109/TITS.2022.3188671.

[16] V. Sujatha, K. L. Prasanna, K. Niharika, V. Charishma, and K. B. Sai, "Network intrusion detection using deep reinforcement learning," in *2023 7th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Erode, India, 2023, pp. 1146–1150. doi: 10.1109/ICCMC56507.2023.10083673.

[17] N. Karyemsetty, T. Syamsundarao, S. Venkata Kishore Babu, D. Suresh Kumar, J. Goddu and B. Samatha, "Intrusion detection system in vehicular network using deep learning approach," in *2023 5th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Tirunelveli, India, 2023, pp. 999–1003. doi: 10.1109/ICSSIT55814.2023.10060872.

[18] H. C. Altunay and Z. Albayrak, "A hybrid CNN + LSTM based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol., Int. J.*, vol. 38, pp. 101322, Feb. 2023. doi: 10.1016/j.jestch.2022.101322.

[19] V. Hnamte and G. S. Member, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," *IEEE Access*, vol. 11, pp. 37131–37148, 2023. doi: 10.1109/ACCESS.2023.3266979.

[20] S. Srinivasan and D. P, "Enhancing the security in cyber-world by detecting the botnets using ensemble classification based machine learning," *Meas.: Sens.*, vol. 25, no. 5, pp. 100624, 2023. doi: 10.1016/j.measen.2022.100624.
[21] A. Sohail, B. Ayisha, I. Hameed, M. M. Zafar, H. Alquhayz and A. Khan, "Deep neural networks based meta-learning for network intrusion detection," arXiv preprint arXiv:2302.09394, 2023.
[22] I. H. Hassan, M. Abdullahi, M. M. Aliyu, S. A. Yusuf, and A. Abdulrahim, "An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection," *Intell. Syst. Appl.*, vol. 16, no. 2020, pp. 200114, 2022. doi: 10.1016/j.iswa.2022.200114.
[23] A. R. Gad, M. Haggag, A. A. Nashat, and T. M. Barakat, "A distributed intrusion detection system using machine learning for IoT based on ToN-IoT dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, 2022. doi: 10.14569/issn.2156-5570.
[24] A. Sharma, H. Babbar, and A. Sharma, "TON-IoT: Detection of attacks on Internet of Things in vehicular networks," in *2022 6th Int. Conf. Electr., Commun. Aerosp. Technol.*, Coimbatore, India, 2022, pp. 539–545. doi: 10.1109/ICECA55336.2022.10009070.
[25] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system," *SN Comput. Sci.*, vol. 3, no. 2, pp. 2227, 2022. doi: 10.1007/s42979-022-01031-1.
[26] W. Wang, F. Harrou, B. Bouyeddou, S. M. Senouci, and Y. Sun, "A stacked deep learning approach to cyber-attacks detection in industrial systems: Application to power system and gas pipeline systems," *Cluster Comput*, vol. 25, no. 1, pp. 561–578, 2022. doi: 10.1007/s10586-021-03426-w.
[27] C. A. Fadhilla, M. D. Alfikri, and R. Kaliski, "Lightweight meta-learning BotNet attack detection," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8455–8466, 2022. doi: 10.1109/JIOT.2022.3229463.
[28] S. Latif *et al.*, "Intrusion detection framework for the Internet of Things using a dense random neural network," *IEEE Trans Industr Inform.*, vol. 18, no. 9, pp. 6435–6444, 2022. doi: 10.1109/TII.2021.3130248.
[29] R. Soleymanzadeh, M. Aljasim, M. W. Qadeer, and R. Kashef, "Cyberattack and fraud detection using ensemble stacking," *AI*, vol. 3, no. 1, pp. 22–36, 2022. doi: 10.3390/ai3010002.
[30] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, no. 5, pp. 107810, Apr. 2022. doi: 10.1016/j.compeleceng.2022.107810.
[31] J. Simon, N. Kapileswar, P. K. Polasi, and M. A. Elaveini, "Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm," *Comput. Electr. Eng.*, vol. 102, no. 4, pp. 108190, Sep. 2022. doi: 10.1016/j.compeleceng.2022.108190.
[32] J. Ahmad, S. A. Shah, S. Latif, F. Ahmed, Z. Zou and N. Pitropakis, "DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8112–8121, 2022. doi: 10.1016/j.jksuci.2022.07.023.
[33] W. Choukri, H. Lamaazi, and N. Benamar, "Abnormal network traffic detection using deep learning models in IoT environment," in *2021 3rd IEEE Middle East North Africa COMMun. Conf. (MENACOMM)*, Agadir, Morocco, 2021, pp. 98–103. doi: 10.1109/MENACOMM50742.2021.9678276.
[34] H. C. Altunay and Z. Albayrak, "Network intrusion detection approach based on convolutional neural network," *Eur. J. Sci. Technol.*, 2021. doi: 10.31590/ejosat.954966.
[35] Q. A. Al-Haija, C. D. McCurry, and S. Zein-Sabatto, "Intelligent self-reliant cyber-attacks detection and classification system for IoT communication using deep convolutional neural network," *Lect. Notes Netw. Syst.*, 2021. doi: 10.1007/978-3-030-64758-2_8.
[36] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021. doi: 10.1109/ACCESS.2021.3120626.
[37] A. M. Aleesa, M. Younis, A. A. Mohammed, and N. M. Sahar, "Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques," *J. Eng. Sci. Technol.*, vol. 16, no. 1, pp. 711–727, 2021.

[38] D. Vasan, M. Alazab, S. Venkatraman, J. Akram, and Z. Qin, "MTHAEL: Cross-architecture IoT malware detection based on neural network advanced ensemble learning," *IEEE Trans. Comput.*, vol. 69, no. 11, pp. 1654–1667, 2020. doi: 10.1109/TC.2020.3015584.

[39] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020. doi: 10.1109/ACCESS.2020.2972627.

[40] S. Latif, Z. Idrees, Z. Zou, and J. Ahmad, "DRaNN: A deep random neural network model for intrusion detection in industrial IoT," in *2020 Int. Conf. UK—China Emerg. Technol. (UCET)*, Glasgow, UK, 2020, pp. 1–4. doi: 10.1109/UCET51115.2020.9205361.

[41] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, 2019. doi: 10.1109/JIOT.2019.2912022.

[42] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Comput. Netw.*, vol. 148, no. November, pp. 164–175, 2019. doi: 10.1016/j.comnet.2018.11.010.

[43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.