Machine Learning-Driven Intrusion Detection for DDoS Attack Mitigation in Cyber-Physical Production Systems

Fathimath Naseera[1], Jayapriya K.N.[2]

[1,2] Department of Computer Science and Engineering, Kathir College of Engineering, Coimbatore, India

E-mail: [1]naseerareenup@gmail.com, [2]jayapriya@kathir.ac.in

Abstract

The research aims to create an intelligent Intrusion Detection System (IDS) for Cyber-Physical Production Systems (CPPS) that uses machine learning approaches to identify Distributed Denial-of-Service (DDoS) attacks. The proposed approach trains and compares the performance of Random Forest (RF) and Deep Neural Networks (DNN). To train the various models, the dataset is first pre-processed by feature selection, normalisation, and splitting. Fast classification and interpretability are enabled by the RF model, while deep feature learning is used by the Deep Neural Networks model to identify intricate attack patterns. The Random Forest and Deep Neural Networks models achieved high accuracy scores of 98.2 and 99.3%, respectively, and low false positive rates, according to experimental assessments on benchmark datasets. These results show that the Deep Neural Networks based Intrusion Detection System is a good option for real-time industrial security applications as it effectively protects CPPS from changing cyberthreats.

Keywords: Cyber-Physical Production Systems (CPPS), Distributed Denial of Service (DDoS), Machine Learning (ML), Intrusion Detection System (IDS), Smart Manufacturing, Network Security, Anomaly Detection, Industrial Cybersecurity

DOI: https://doi.org/10.36548/jitdw.2025.1.003
Received: 03-03-2025. received in revised form: 01-04-2025, accepted: 17-04-2025, published: 28-04-2025

1. Introduction

By integrating cutting-edge computer, communication, and control technology with physical production activities, Cyber-Physical Production Systems (CPPS) have completely transformed contemporary industrial processes [10]. CPPS are essential elements of Industry 4.0 that provide autonomous decision-making, real-time monitoring, and increased efficiency in intricately linked systems. These developments provide serious vulnerabilities to cybersecurity attacks, even while they have major operational and economic advantages. The dependability, security, and effectiveness of CPPS are seriously threatened by distributed denial of service (DDoS) attacks in particular. DDoS attacks may impede decision-making, stop production, and interrupt important communication by flooding networks with malicious traffic. This can have serious operational and financial repercussions.

Because DDoS attacks are dynamic and sophisticated, traditional cybersecurity tools like static rule-based Intrusion Detection Systems (IDS) often fail to identify and stop them in CPPS [11]. Diverse protocols, heterogeneous components, real-time data streams, and the need for high-speed communication are some of the particular difficulties that CPPS settings provide. The installation of strong defences is made more difficult by the attackers' changing behaviour. Protecting CPPS operations in this situation requires the capacity to quickly spot unusual patterns and adjust flexibly to new threats.

Because it can learn from and generalise from patterns in big datasets, machine learning (ML) has become a potent tool for intrusion detection in contemporary networks [12]. The requirements of CPPS are especially well-suited for ML-based Intrusion Detection Systems (IDS), which can efficiently identify known and new attack patterns by examining complicated traffic patterns and adjusting to changing attack vectors. These systems can identify cyber threats with high accuracy and low false-positive rates by using supervised, unsupervised, and reinforcement learning approaches.

This study aims at the development and deployment of an ML-driven intrusion detection system specifically designed to identify DDoS attacks in CPPS. The suggested framework analyses network data in real time and detects anomalous patterns suggestive of cyberattacks using sophisticated machine learning methods. Key CPPS security problems, such as scalability, efficiency, and adaptation to resource-constrained contexts, were taken into consideration while developing the system. Additionally, the study uses both benchmark

datasets and actual CPPS situations to assess the IDS performance across several measures, including detection accuracy, processing overhead, and responsiveness.

This study advances intelligent defence mechanisms in smart manufacturing ecosystems by tackling significant weaknesses in CPPS security, such as adaptations to evolving attack patterns, and provides higher accuracy and faster detection speeds in real time compared to previous works described in the next section. An important step towards robust CPPS operations is the implementation of ML-based intrusion detection, which will guarantee sustained industrial development despite changing cybersecurity threats.

2. Related Work

DDoS attacks have been detected and stopped using a variety of deep learning techniques. Hybrid models that integrate techniques and algorithms improved the identification of DDoS attacks in this research. Deep learning techniques for DDoS attack detection have been tested by several researchers. CNNs and RNNs are outperformed by basic neural networks in a CSE-CIC-IDS2018 study. The Heartbleed vulnerability, DDoS, penetration, brute force, and botnet problems were all simulated in the studies [1]. Overfitting of RNN and CNN models often leads to an increase in false positives and negatives. The accuracy and precision of the main neural network were 82% and 42%, respectively.

Intrusion detection systems (IDS) and DDoS detection have been enhanced by the use of deep learning algorithms in hybridization approaches. These methods deal with issues such as shifting attack patterns, the need for several classifiers to identify distinct attacks, and dynamic network traffic data. Data dimensionality may be decreased by using K-means clustering, wrapper feature selection, and a genetic method. The accuracy of detection is increased when secret data is analysed using a support vector machine. While blockchain and machine learning manage datasets and identify network intrusions, another approach combines the Bat algorithm and PCA to find characteristics [2].

In order to identify DDoS attacks in SDN networks, Sudar et al. [3] studied the use of Support Vector Machines (SVM) and Decision Trees (DTs). They used the KDD CUP dataset to test their suggested methodology. Nevertheless, their approach performed poorly; Decision Trees only received a 785 accuracy.

In order to identify several kinds of DDoS attacks, such as point attacks, flow table switch attacks, SDN controller attacks, and bandwidth attacks, Santos et al. [4] used Multiple Layer Perceptron (MLP), Random Forest Algorithm, SVM, and DTs. They used a realistic dataset to evaluate their methodology. However, with an accuracy rate of just 90%, the data showed that MLP and SVM performed poorly in classification when it came to identifying controller attacks.

Celesova et al. [5] proposed a technique that controls DDoS attacks and safeguards the data planes in SDN networks by using a Deep Neural Network (DNN). Nevertheless, they trained, tested, and evaluated their suggested system using the UNSW-NB15 dataset, which is not specifically made for the SDN network environment. As a result, the method's performance on calculation metrics was poor.

Hybrid deep learning models were created in some experiments. To identify an attack, Gadze et al., 2021 [6] suggested a model that included CNN and LSTM, two forms of deep learning. Mininet used Floodlight as an external controller and OpenFlow switches to create the dataset dynamically. According to the results, RNN LSTM achieved an accuracy of 89.63% in comparison to the scores of linear-based models such as SVM (86.85%) and Naive Bayes (82.61%). The KNN approach, which is based on linear models, had an even greater accuracy than their model, which had an accuracy of 99.4%. Furthermore, the model performed best when the data was divided into 70/30 train/test split ratios. MSCNN-LSTM-AE is the name of the hybrid autoencoder model developed by Singh and Jang-Jaccard (2022) [7]. This model combined an LSTM with a multi-scale convolutional neural network (MSCNN) to identify abnormalities in network traffic. Initially, the MSCNN autoencoder was used to assess the dataset's spatial properties. The temporal properties of the latent space features learnt by the MSCNN-AE were then identified using an LSTM-based autoencoder network. Using the UNSW-NB15 [8], NSL-KDD [9], and CICDDoS2019 tests, the authors examined their work. Their model (MSCNN-LSTM-AE) has a recall score of 92.26% and an accuracy score of 93.76%.

3. Proposed Work

3.1 Dataset

Analysing existing intrusion detection datasets, with an emphasis on those that have comprehensive network traffic data relevant to DDoS attacks, is the first step in the dataset gathering process. The feature sets, attack variety, and applicability to CPPS contexts of the public datasets NSL-KDD and CSE-CIC-IDS2018 are evaluated and used in this application. Because these datasets include labelled traffic examples, supervised learning techniques may be used to train the IDS. Data labels from these datasets can be seen in Table 1, with a breakdown of the per-class data samples. The total number of data samples used for this study is 282,683 from the NSL-KDD dataset and 11,025,262 from the CSE-CIC-IDS2018 dataset. The data has been split in an 80:20 ratio, using 80% of the sample for training and 20% testing the data models built.

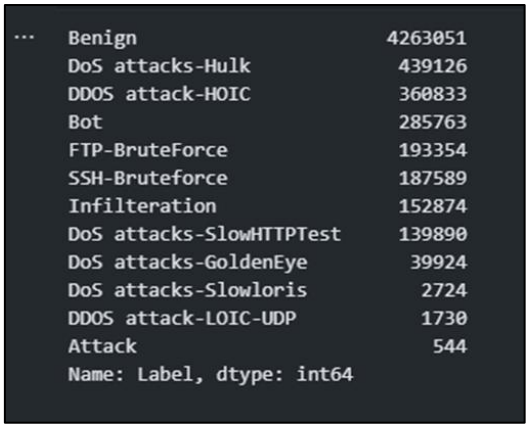
Dataset	Classification	Total
KDD	Benign	60,591
	Neptune Attack	58,001
	Smurf Attack	164,091
CSE-CIC-IDS 2018	Benign	9,108,759
	DoS-Hulk Attack	461,912
	DoS-SlowHTTPTest	139,890
	DoS-GoldenEye	41,508
	DoS-Slowloris	10,990
	DDoS-LOIC-HTTP	576,191
	DDoS-HOIC	686,012

Table 1. Dataset Labels of NSL-KDD and CSE-CIC-IDS2018 [16]

3.2 Preprocessing of Data and Feature Extraction

Data cleaning, which includes deleting incomplete or damaged entries, is the first step in the preparation process [13]. Network traffic statistics may have missing values due to network congestion, improper logging, or packet loss during collection. To overcome this, imputation approaches, such as substituting the median or mean of the associated feature distribution for missing data, have been used. To preserve data integrity, an entry is rejected when a significant portion of a record is missing. Furthermore, duplicate data, which could have been recorded more than once as a result of network monitoring equipment capturing the same event from several sources, have been found and eliminated. These thorough preparation procedures are used to convert the dataset into a high-quality, structured format that is

appropriate for ML-based IDS. This minimises false positives and negatives in CPPS systems and guarantees that the models trained on this dataset can properly and effectively identify DDoS attacks. A preview of the pre-processed CSE-CIC-IDS2018 dataset is shown in Figure 1.



```
...  Benign                      4263051
     DoS attacks-Hulk             439126
     DDOS attack-HOIC             360833
     Bot                          285763
     FTP-BruteForce               193354
     SSH-Bruteforce               187589
     Infilteration                152874
     DoS attacks-SlowHTTPTest     139890
     DoS attacks-GoldenEye         39924
     DoS attacks-Slowloris          2724
     DDOS attack-LOIC-UDP           1730
     Attack                          544
     Name: Label, dtype: int64
```

Figure 1. Preview of Pre-Processed CSE-CIC-IDS2018 Dataset

Feature extraction, which entails obtaining significant network properties from unprocessed packet data, comes after the dataset has been cleaned. The features extracted after preprocessing the data include packet arrival time intervals, flow length, source and destination IP address ports, protocol kinds, timestamps, packet sizes, and transmission speeds. Since DDoS attacks appear as anomalous traffic patterns. A sample of these extracted features from the CSE-CIC-IDS2018 dataset is shown in Figure 2. To capture differences between normal and attack traffic, high-level statistical characteristics are calculated, including the mean, standard deviation, and entropy of packet flows. Since time-based abnormalities often point to continuing attack activities, temporal factors are also taken into account.

	Dst Port	Protocol	Timestamp	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts
count	6.067402e+06	6.067402e+06	6.067402e+06	6.067402e+06	6.067402e+06	6.067402e+06	6.067402e+06
mean	2.236399e-01	3.914833e-01	3.956549e-01	1.046317e-01	6.394926e-02	4.106511e-02	6.450427e-02
std	3.510408e-01	3.061866e-01	3.596658e-01	2.312967e-01	1.464035e-01	1.013161e-01	1.644647e-01
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	8.087528e-04	3.529412e-01	4.877811e-02	6.050006e-06	3.229682e-06	0.000000e+00	0.000000e+00
50%	6.759850e-03	3.529412e-01	2.176439e-01	1.425583e-04	2.309469e-03	2.183406e-03	1.076763e-03
75%	5.570607e-01	3.529412e-01	7.947376e-01	3.251452e-02	2.078522e-02	1.612903e-02	2.101870e-02
max	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00

Figure 2. Extracted Features Sample From the CSE-CIC-IDS2018 Dataset

3.3 Random Forest Architecture

Because of its resilience, effectiveness, and capacity to manage high-dimensional network traffic data, the Random Forest (RF) method is chosen as one of the fundamental machine learning models for identifying Distributed Denial of Service (DDoS) attacks in Cyber-Physical Production Systems (CPPS) [14]. In order to improve intrusion detection performance, this technique describes the precise procedures needed to construct, train, and optimise the RF model.

In order to increase classification accuracy, the Random Forest model, an ensemble learning technique, builds many decision trees during training and combines their results as shown in Figure 3. A random feature pick and a portion of the training data are used to construct each decision tree. The model is very resistant to overfitting and is able to manage intricate attack patterns in CPPS settings since the final classification choice is decided by majority vote among the trees.

Choosing how many decision trees to include in the ensemble is the first stage in building the RF architecture. The "number of estimators," as this parameter is called, is essential for striking a balance between computational efficiency and performance. Although more trees often result in better classification accuracy, they also need more time and resources during training. An ideal number of trees is found using intensive testing and hyperparameter adjustment to get the greatest possible balance between computational cost and detection performance.

Each decision tree is built using a recursive procedure that repeatedly divides the dataset according to feature thresholds that optimise information acquisition. At each split, the purity of a node is assessed using the Gini impurity criterion, which guarantees that data points from the same class are clustered together as much as feasible.

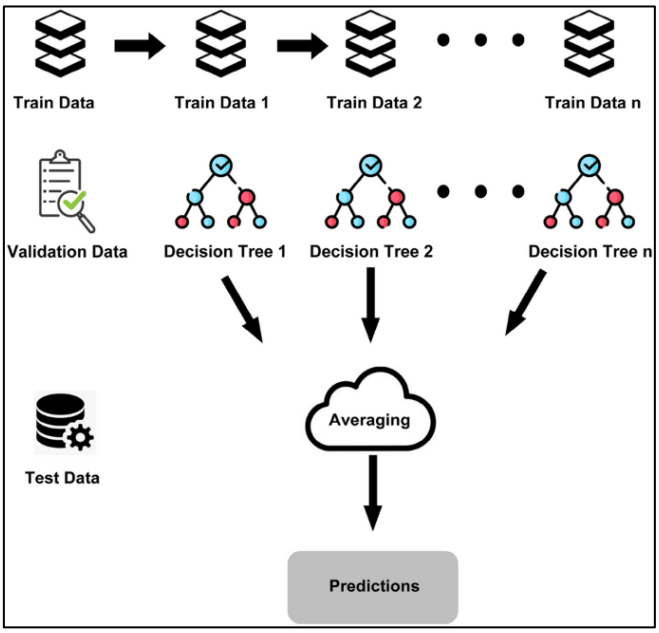


Figure 3. Random Forest Architecture for DDoS Detection [17]

The Python Scikit-learn module is used to construct and implement the Random Forest (RF) model. A predetermined number of decision trees that were individually trained on a randomly chosen portion of the dataset are used to initialise the model. By ensuring that every tree learns from a distinct sample, the bootstrapping process enhances generalisation.

Following training, the RF model is evaluated using the test dataset, and the trees vote by majority to decide which classifications to make. To assess the model's efficacy, performance evaluation methods, including accuracy, precision, recall, and F1-score, are calculated. The evaluation was done based on 20% of the dataset, which had been split for testing purposes using the TensorFlow library during the training process. Grid Search is used for hyperparameter tuning, which optimises parameters like minimum samples per split, maximum depth, and number of trees. The grid Search hyperparameter fitting and the parameters it uses are shown in Figure 4.

```
...  Fitting 5 folds for each of 5 candidates, totalling 25 fits

...  GridSearchCV(cv=5,
                  estimator=RandomForestClassifier(class_weight={0: 0.6351680989134786,
                                                                 1: 2.3495488359278136}),
                  n_jobs=-1, param_grid={'n_estimators': [50, 75, 100, 125, 150]},
                  verbose=1)
```

Figure 4. Grids Search Implementation

3.4 Deep Neural Networks Architecture

To improve the detection of Distributed Denial of Service (DDoS) attacks in Cyber-Physical Production Systems (CPPS), the Deep Neural Network (DNN) model makes use of its capacity to identify intricate patterns and high-dimensional correlations in network traffic data. Because DNNs can automatically build hierarchical feature representations, they are far more successful at detecting intrusions than typical machine learning models. To guarantee precise and effective detection of harmful traffic while reducing false positives and negatives, this technique describes the design, training, and optimization of the DNN architecture.

The network is made up of many completely linked hidden layers that come after the input layer. These layers are each intended to capture various degrees of abstraction in the data. Empirical analysis is used to balance model performance and computing economy while determining the number of hidden layers and neurons per layer. By applying the Rectified Linear Unit (ReLU) activation function to each hidden layer, the model gains non-linearity and can recognise intricate correlations in network traffic data. ReLU is preferred over more conventional activation functions like sigmoid or tanh because it may avoid the vanishing gradient issue, guaranteeing steady and effective training.

The output layer's softmax activation algorithm allocates probability to the attack and regular traffic classes. Two neurons that represent malicious and legitimate traffic are included in the output layer since DDoS detection is a binary classification problem. Because the softmax function guarantees that the model generates a probabilistic output, threshold-based decision-making is possible in practical applications. The overview of this architecture is seen in Figure 5.

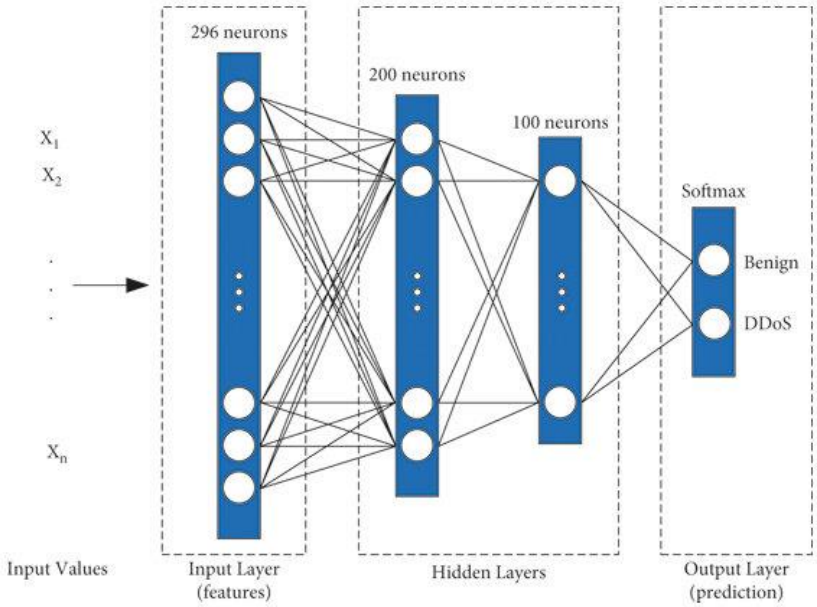


Figure 5. DNN Architecture for DDoS Detection [18]

To identify intricate DDoS attack patterns, the Deep Neural Network (DNN) model is constructed using TensorFlow and Keras, using their deep learning capabilities. A number of hidden layers, an output layer, and an input layer make up the architecture [15]. The pre-processed dataset's chosen characteristics are reflected in the number of neurons in the input layer.

The Adam optimiser is used to train the model, and the loss function is categorical cross-entropy. To iteratively update model weights, mini-batch gradient descent is used. By feeding the dataset into the model in batches, training efficiency is increased and memory utilisation is optimised. Grid Search and Random Search are used for hyperparameter tuning to find the ideal number of layers, neurons per layer, learning rate, dropout rate, and batch size. The training process, epoch by epoch, is previewed in Figure 6.

```
Epoch 1/10
493715/493715 [==============================] - 157s 318us/step - loss: 0.0608 - acc: 0.9785 - val_loss: 0.0144 - val_acc: 1.0000

Epoch 00001: val_acc improved from -inf to 1.00000, saving model to checkpoint-01.hdf5
Epoch 2/10
493715/493715 [==============================] - 126s 256us/step - loss: 0.0290 - acc: 0.9882 - val_loss: 0.0148 - val_acc: 0.9967

Epoch 00002: val_acc did not improve from 1.00000
Epoch 3/10
493715/493715 [==============================] - 134s 271us/step - loss: 0.0258 - acc: 0.9897 - val_loss: 0.0167 - val_acc: 1.0000

Epoch 00003: val_acc did not improve from 1.00000
Epoch 4/10
493715/493715 [==============================] - 108s 219us/step - loss: 0.0236 - acc: 0.9905 - val_loss: 0.0180 - val_acc: 1.0000

Epoch 00004: val_acc did not improve from 1.00000
Epoch 5/10
493715/493715 [==============================] - 94s 190us/step - loss: 0.0214 - acc: 0.9920 - val_loss: 0.0164 - val_acc: 0.9967
```

Figure 6. DNN Training Process

The method used to evaluate this model, based on the test dataset, which served to be the simulated data, is the evaluation of the trained DNN model's classification accuracy, recall, precision, F1-score, and detection latency. The model's capacity to dynamically identify ongoing attacks is further validated using real-time CPPS network settings. Matplotlib was used to plot an accuracy graph evaluating the performance of this data model, while TensorFlow was used to train and test the data model during the training process itself.

4. Results and Discussion

The results of this research demonstrate the effectiveness of the Machine Learning (ML)-based Intrusion Detection System (IDS) in accurately identifying Distributed Denial of Service (DDoS) attacks within Cyber-Physical Production Systems (CPPS). The evaluation is performed based on multiple performance metrics, including accuracy, precision, recall, F1-score, detection latency, and computational efficiency. The results are analyzed separately for the Random Forest (RF) and Deep Neural Network (DNN) models, followed by a comparative analysis to determine the best approach for real-world CPPS applications.

The RF model's ensemble learning strategy, which mixes many decision trees to enhance classification performance, allows it to achieve high detection accuracy. The RF model shows a good capacity to differentiate between normal and attack traffic, with an overall accuracy of almost 98.2% after training and testing on the pre-processed dataset.

To assess the model's capacity to accurately detect attack events while reducing false positives, precision and recall metrics are examined. With a precision of 97.8%, the RF model indicates that the majority of traffic classified as an attack. The algorithm accurately identifies

most attack events without ignoring important threats, as shown by the 96.5% recall. The model's overall robustness is confirmed by the F1-score of 97.1%, which strikes a compromise between accuracy and recall, as noted in Table 2.

Table 2. Performance Analysis of RF

Metric	**Random Forest (RF)**
Accuracy	98.2%
Precision	97.8%
Recall	96.5%
F1-score	97.1%
Detection Latency	8 ms
Training Time	Few minutes

The interpretability of the RF model is one of its main benefits. According to the feature importance study, characteristics including the number of unique source IP addresses, packet size distribution, and packet flow length are important in detecting DDoS attacks. Because of its interpretability, RF is a useful tool for security analysts as it offers information about the most pertinent aspects of attacks.

The RF model's computational efficiency is limited, despite its excellent detection performance. A typical computer system may complete the training process in a matter of minutes. However, with an average detection delay of 8 milliseconds per network transaction, real-time classification is a little slower than deep learning techniques because of the ensemble structure of the model which is also noted in Table 2. The majority of CPPS applications can tolerate this delay, however high-speed industrial settings that need ultra-low latency intrusion detection could find it unsuitable.

When it comes to identifying intricate DDoS attack patterns, the DNN model outperforms RF in terms of learning capabilities since it is built to capture intricate correlations among network data. Following a thorough training process on the dataset with optimised hyperparameters, the DNN model outperforms RF in detection performance, with an overall accuracy of 99.3% as seen in Figure 7.

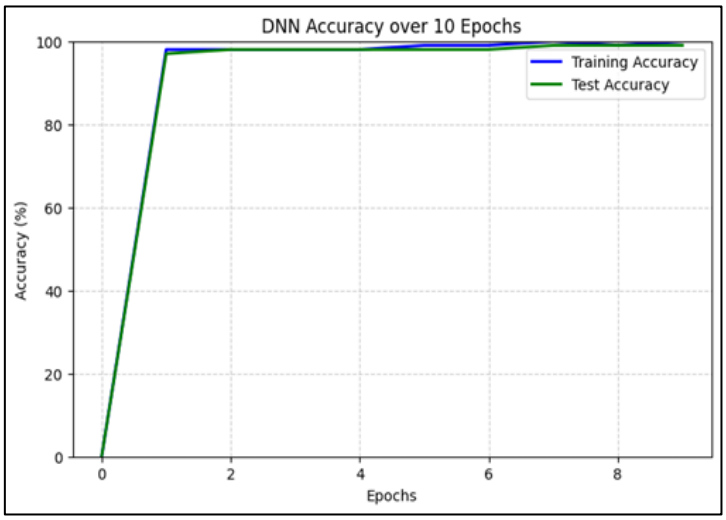


Figure 7. Performance of DNN Model Based on Various Metrics

The model is successful in reducing false positives and false negatives, as shown by its precision and recall scores. The DNN model achieves 99.1% accuracy, guaranteeing that almost all attack classifications are accurate. With a 98.8% recall rate, the model is quite dependable for practical applications as it can identify almost every attack incident. The model's resilience is further shown by its 98.9% F1-score as shown in Table 3. Using the network monitoring tool Wireshark, packet and flow data were recorded in real time. Python-based scripts were then used to convert the data into organised representations. The relevant characteristics needed for model input, including packet size, flow length, protocol types, and inter-arrival periods, were retrieved using these scripts. In future a Python-based API would be deployed in the real-time detection models as services, enabling the trained RF or DNN to classify incoming traffic characteristics.

Table 3. Performance Analysis of DNN

Metric	**Deep Neural Network (DNN)**
Accuracy	99.3%
Precision	99.1%
Recall	98.8%
F1-score	98.9%
Detection Latency	4 ms
Training Time	Several hours

The DNN model's benefit is that it can learn high-dimensional representations of network data and identify previously unobserved attack variants. In contrast to RF, which depends on characteristics that are explicitly chosen, DNN automatically extracts hierarchical features, improving generalisation. This feature is very helpful in CPPS settings where attackers are often changing their tactics.

Nevertheless, there are more computing requirements for the DNN model. The more resource-intensive training method requires hours of work on a high-performance computer system with a GPU. In spite of this, real-time classification inference time is much less than RF, with an average detection delay of 4 ms per network flow, which was calculated by timing how long it took for the system to provide a matching intrusion detection result after a network traffic instance arrived. This makes DNN a better option for CPPS settings that need to identify intrusions quickly and effectively.

To find the best method for CPPS intrusion detection, a comparison of the RF and DNN models is carried out. The main performance indicators for both models are compiled in Table 4 below.

Table 4. Performance Analysis of RF and DNN Models

Metric	**Random Forest (RF)**	**Deep Neural Network (DNN)**
Accuracy	98.2%	99.3%
Precision	97.8%	99.1%
Recall	96.5%	98.8%
F1-score	97.1%	98.9%
Detection Latency	8 ms	4 ms
Training Time	Few minutes	Several hours

According to the findings, both models provide good detection accuracy; however, DNN outperforms RF across the board. Given its interpretability and shorter training time, the RF model is still a solid contender and a optimal choice for settings with constrained computing resources. Nonetheless, the DNN model is the recommended option for real-time CPPS applications that need quick attack detection because of its better classification performance and reduced inference latency.

5. Conclusion

This study introduces a sophisticated Intrusion Detection System (IDS) based on Machine Learning (ML) that can identify Distributed Denial of Service (DDoS) attacks in Cyber-Physical Production Systems (CPPS). Both Random Forest (RF) and Deep Neural Networks (DNN) are used in the research to examine network data and spot harmful activity. With a detection delay of 8 milliseconds per network flow and an accuracy of 98.2%, the evaluation findings show that the RF model enables great classification performance and high interpretability. It has trouble adjusting to new attack patterns. However, in terms of accuracy (99.3%) and detection speed (4 milliseconds per network flow), the DNN model outperforms RF, which makes it more appropriate for real-time CPPS settings. The IDS's implementation and practical testing verify that it can operate effectively under a range of network demands. The technology minimises possible downtime and protects industrial operations from cyber risks by sending out instant notifications when it detects breaches.

References

[1] Hagar, Abdulnaser A., and Bharti W. Gawali. "Deep learning for improving attack detection system using CSE-CICIDS2018." NeuroQuantology 20, no. 6 (2022).

[2] Yaras, S.; Dener, M. IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. Electronics 2024, 13, 1053.

[3] Sudar, K.M.; Beulah, M.; Deepalakshmi, P.; Nagaraj, P.; Chinnasamy, P. Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–5.

[4] Santos, Reneilson, Danilo Souza, Walter Santo, Admilson Ribeiro, and Edward Moreno. "Machine learning algorithms to detect DDoS attacks in SDN." Concurrency and Computation: Practice and Experience 32, no. 16 (2020): e5402.

[5] Celesova, B.; Val'ko, J.; Grezo, R.; Helebrandt, P. Enhancing security of SDN focusing on control plane and data plane. In Proceedings of the 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal, 10–12 June 2019; pp. 1–6.

[6] Gadze, James Dzisi, Akua Acheampomaa Bamfo-Asante, Justice Owusu Agyemang, Henry Nunoo-Mensah, and Kwasi Adu-Boahen Opare. "An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers." Technologies 9, no. 1 (2021): 14.

[7] Singh, Amardeep, and Julian Jang-Jaccard. "Autoencoder-based unsupervised intrusion detection using multi-scale convolutional recurrent networks." arXiv preprint arXiv:2204.03779 (2022).

[8] The UNSW-NB15 Dataset|UNSW Research. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 6 February 2025).

[9] NSL-KDD|Datasets|Research|Canadian Institute for Cybersecurity|UNB. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 6 February 2025).

[10] Monostori, László. "Cyber-physical production systems: roots from manufacturing science and technology." at-Automatisierungstechnik 63, no. 10 (2015): 766-776.

[11] Zheng, Yu, Zheng Li, Xiaolong Xu, and Qingzhan Zhao. "Dynamic defenses in cyber security: Techniques, methods and challenges." Digital Communications and Networks 8, no. 4 (2022): 422-435.

[12] Liu, Hongyu, and Bo Lang. "Machine learning and deep learning methods for intrusion detection systems: A survey." applied sciences 9, no. 20 (2019): 4396.

[13] Ridzuan, Fakhitah, and Wan Mohd Nazmee Wan Zainon. "A review on data cleansing methods for big data." Procedia Computer Science 161 (2019): 731-738.

[14] Oyetoro, Amos, Joseph Mart, and Ugochukwu Amah. "Using Machine Learning Techniques Random Forest and Neural Network to Detect Cyber Attacks." ScienceOpen Preprints (2023).

[15] Han, Hyojoon, Hyukho Kim, and Yangwoo Kim. "Correlation between deep neural network hidden layer and intrusion detection performance in IoT intrusion detection system." Symmetry 14, no. 10 (2022): 2077.

[16] Kim, Jiyeon, Jiwon Kim, Hyunjung Kim, Minsun Shim, and Eunjung Choi. "CNN-based network intrusion detection against denial-of-service attacks." Electronics 9, no. 6 (2020): 916.

[17] Najar, Ashfaq Ahmad, and S. Manohar Naik. "DDoS attack detection using MLP and Random Forest Algorithms." International Journal of Information Technology 14, no. 5 (2022): 2317-2327.

[18] Ortet Lopes, Ivandro, Deqing Zou, Francis A. Ruambo, Saeed Akbar, and Bin Yuan. "Towards effective detection of recent DDoS attacks: A deep learning approach." Security and Communication Networks 2021, no. 1 (2021): 5710028.