Journal of Engineering and Sustainable Development
Vol. 30, No. 03, Month 2026
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.3606

Deep Learning-Based Monitoring System to Enhance IoT Network Performance

Radhi Sehen Issa[1*], Gregor Alexander Aramice[1], Noorulden Basil[1], Mustafa Mahdi Ali[2], Takele Ferede Agajie[3], Alfian Ma'arif[4]

[1]Electrical Engineering Department, College of Engineering, Mustansiriyah University, Baghdad, Iraq
[2]Telecommunication Research Institute (TELMA), E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga Boulevard Louis Pasteur 35, 29010, Málaga, Spain
[3]Department of Electrical and Computer Engineering, Faculty of Technology, Debre Markos University, P. BOX 269, Debre Markos, Ethiopia
[4]Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

[*]Email: radhi.sahan@uomustansiriyah.edu.iq

Article Info

Received	09/08/2025
Revised	18/04/2026
Accepted	27/04/2026

Abstract

The rapid growth and increasing complexity of Internet of Things (IoT) networks require efficient real-time monitoring and anomaly detection mechanisms. Traditional machine learning approaches often struggle to handle the dynamic and high-dimensional traffic generated by IoT environments. This study investigates the effectiveness of deep learning models, including Feedforward Neural Networks (FFNN), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP), for enhancing IoT network monitoring. The models were trained using both synthetic and real-world IoT traffic datasets in MATLAB with Adam and Stochastic Gradient Descent with Momentum (SGDM) optimizers to improve convergence and training stability. Experimental results demonstrate that deep learning models outperform traditional machine learning techniques in detecting complex traffic patterns and anomalies. Among the evaluated models, CNN achieved the highest accuracy of 94%, compared with Decision Trees (78.5%) and Support Vector Machines (85.7%). CNNs effectively capture spatiotemporal traffic characteristics, while MLPs efficiently model nonlinear relationships in network data. The proposed framework provides a scalable, reliable approach to real-time IoT network monitoring.

Keywords: Anomaly Detection, Convolutional Neural Network (CNN), Deep Learning, Feedforward Neural Network (FFNN), Internet-of-things, Multilayer Perceptron (MLP), Network Monitoring, Performance Metrics, Real-time Monitoring.

1. Introduction

The Internet of Things (IoT) has revolutionized numerous sectors by enabling interconnected devices to exchange information over the internet. This connectivity has enabled a robust IoT ecosystem, with widespread deployment across healthcare, smart cities, and industrial automation. In these domains, IoT devices play a crucial role in providing continuous monitoring and control of critical systems [1]. However, as IoT networks continue to expand, they face increasing challenges, including network congestion, latency, bandwidth limitations, and security vulnerabilities [2]. These issues not only degrade the performance of IoT networks but also threaten data integrity and system security, highlighting the need for real-time network monitoring and effective anomaly detection mechanisms [3].

The basic machine learning approaches, such as Decision Tree, Support Vector Machines, and Naive Bayes, were used for the IoT network monitoring and anomalous behavior detection. These models, while proven handy for the management of cleanly formatted data, are not as effective when it comes to the constantly evolving stream of traffic that IoT presents [4]. First, while traditional approaches allow the identification of dependent relationships between variables in a linear manner, most of the time they fail to address nonlinear dependencies that are typical when analyzing traffic in the network generated by thousands of IoT devices in real time [5]. Hence, there is a rising need for improved techniques that can be used for analysis of the network performance and for identification of known and unknown flaws, at the same time [6].

To find solutions to these problems, such issues have recently been solved by deep learning algorithms. Among these algorithms are Feedforward Neural Networks (FFNN), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP); they can discover latent relations and associations of big data independently [7]. For instance, CNNs excel at locating the spatial pyramid within data, and this helps in capturing traffic within a network [8]. Nonetheless, MLPs can recognize a nonlinear relation between features, whereby they can discover relations in network performance that cannot be discovered by any other actual method [9]. The final type is FFNNs, which are less complex than the previous types but still retain computational effectiveness for analysis of simple IoT data streams [10].

To this end, we continue the utilization of the three deep learning models in this work, namely FFNN, CNN, and MLP, to serve the monitoring aim.

IoT network performance. We evaluate the performance of the models in terms of anomaly detection, their capability to forecast network problems, and their general performance in real-time network monitoring tasks. Furthermore, we also illustrate how the models developed in this study have higher accuracy, precision, and better anomaly detection rates than conventional machine learning techniques. This paper enhances the knowledge regarding deep learning algorithms in IoT networks, which leads to the design of better and more efficient monitoring systems for the better and more secure performance of IoT networks [11]. The aim of this paper is to develop and evaluate a deep learning–based monitoring framework for improving IoT network performance through accurate anomaly detection and real-time traffic analysis. The novelty of this work lies in the design of a unified experimental framework that evaluates CNN, MLP, and FFNN models under identical preprocessing, optimization, and training conditions using both synthetic and real-world IoT datasets. Unlike previous studies that rely on isolated model evaluations or inconsistent experimental settings, this work provides a controlled comparative analysis and architectural interpretation of deep learning models for IoT traffic monitoring. By integrating heterogeneous datasets and analyzing model convergence behavior and training stability, the proposed approach provides a reproducible and scalable baseline for intelligent IoT network monitoring systems. Our contributions are summarized as follows:

1. This work suggests a single deep learning system that combines synthetically generated IoT traffic (with a regulated injection of anomalies), as well as real IoT environments (using the 23 system), in a single standardized pipeline, which allows a systematic study of model generalization in heterogeneous IoT settings.

2. In contrast to the previous experiments, where comparison is made of models under varying conditions, this experiment imposes the same preprocessing, normalizing, feature extraction, hyperparameter optimization, optimization strategy, and stopping criteria of CNN, MLP, and FFNN, making it certain that performance differences are purely architecture-driven.

3. The analysis offers a systematic elucidation of the association between the features of the IoT traffic and the architectural behavior, demonstrating how the spatial feature extraction of CNN and the nonlinear representation capability of MLP affect the performance of the detection of anomalies.

4. Besides traditional metrics, convergence behavior, dynamics of validation loss, stability of training, and sensitivity are also analyzed in the work, which provides a practical idea of the feasibility of the implementation of the IoT in reality.

5. The suggested framework is formulated as a scalable benchmark to be able to add temporal models (e.g., RNN/LSTM) to detect anomalies in IoT over time.

2. Related Works

The growth of IoT networks has caused an improvement in the research on network performance monitoring and anomaly detection. Classical machine learning methods such as Decision Tree, Support Vector Machine, and Naive Bayes have already been used in previous studies to determine the type of network activities and detect anomalies. Since the IoT networks are dynamic and unstructured, some of these techniques might not be effective enough and slow to scale up, particularly in large and highly complex networks [12], [13]. Decision Trees are simple to comprehend and ineffective with large amounts of data from IoT gadgets and may overfit complicated settings [14]. Although Support Vector Machines (SVM) are more dependable and can handle non-linear models with the kernel technique, they are computationally intensive and cannot be used in real-time tasks [15]. Naive Bayes (NB) is also a rapid and efficient algorithm, but the assumption allows the features to be independent, which is why it is impossible to represent complex interactions in network traffic [16].

The latest deep learning models, like CNN and MLP, have seen the light of day to be able to perform well in modeling IoT data with non-linear relationship-rich data. Image processing CNNs have been effectively used to monitor network activities because of their ability to glean spatial properties of the network traffic; the CNNs have their roots in image processing [17]. This data type is typical of the IoT networks since the data is multivariate and spatially connected (e.g., vehicle traffic and position of vehicle) because they can learn even the faintest hint of the poor performance by their layered formulations [18], [19]. Past studies have also found CNNs to be superior to the traditional models in terms of results and time, when it comes to real-time surveillance applications [20]. This also holds in the case of MLPs that have also been effectively utilized in IoT applications and achieved higher detection rates of anomalies compared to the traditional machine learning methods [21]. However, little attention has been paid towards the use of Feedforward Neural Networks (FFNNs) in monitoring networks, though they could have promises of enhancing computational performance and reducing complexity under specific network traffic scenarios [22].

The work also supplements the knowledge of three models of the FFNN, CNN, and MLP models, which have already been suggested in the context of the Internet of Things network

monitoring. With that said, we compare these models in the context of accuracy, precision, recall, and F1-score in this paper so that the reader can relate to the potential of the models in real-time monitoring and anomaly detection. A comparison of the Deep Learning and Machine Learning models of IoT network monitoring is presented in Table 1.

The machine learning and deep learning models have proven to be of great potential in the monitoring of the network in IoT, but their application in practice has several challenges that should be well-thought-out. Conventional machine learning algorithms like Decision Trees, Support Vector Machines, and Naive Bayes do not handle scalability on large-scale high-velocity data streams of IoT systems and can perform poorly in a highly dynamic network environment. Decision Trees are also easily overfit, whereas SVMs are resource-intensive and not suitable for real-time monitoring. Naive Bayes is effective, but it is based on the hypothesis of the independence of features, which is not very realistic in the complex IoT traffic patterns.

Table 1. Comparison of Machine Learning and Deep Learning Models in IoT Network Monitoring.

Model	Strengths	Weaknesses	Performance in IoT Monitoring	References
Decision Tree (DT)	Simple, easy to interpret	Prone to overfitting, poor scalability	Moderate	[12], [13]
Support Vector Machine (SVM)	Robust with non-linear data, good accuracy	Computationally expensive, slow	Moderate-High	[14], [15]
Naive Bayes (NB)	Fast, efficient	Assumes feature independence	Low	[16]
Convolutional Neural Network (CNN)	Captures spatial hierarchies, good with complex data	Requires large datasets, computationally heavy	High	[17], [18]
Multilayer Perceptron (MLP)	Handles non-linear relationships, scalable	May overfit without regularization	High	[19], [20]
Feedforward Neural Network (FFNN)	Simple architecture, efficient	Limited ability with complex data	Moderate-High	[21], [22]

Deep learning networks are, conversely, more accurate and can better predict non-linear and complex data, but have the disadvantages of high computational complexity, high training data needs, and high energy usage, which can be undesirable in resource-limited IoT gadgets. Also, deep learning models do not usually have interpretability, and it is challenging to comprehend the judgment procedures in security-sensitive systems. Other typical obstacles in both solutions are an imbalance in data, noise in IoT traffic, latency, and problems with real-time deployment. To overcome these barriers, the need to create lightweight, scalable, interpretable models, and effective preprocessing of data and optimization methods specific to the IoT contexts, as shown in Table 1, needs to be met. The main gap in the research discovered within the currently existing literature is the absence of methodologically controlled and cross-condition analysis frameworks of the IoT network anomaly detection. Most previous works are based on the use of single deep learning models on one dataset, which is commonly preprocessed inconsistently, has different hyperparameter settings, and has a small evaluation criterion, making their comparison and evaluation challenging. Moreover, a significant number of works report a better accuracy without giving architectural-level interpretation, and deployment viability in a realistic IoT setting. This gap is bridged by this paper, which proposes a common and standardized experimental setup to test the CNN, MLP, and FFNN models in the same preprocessing, training, and optimization conditions using both synthetic and natural IoT test sets. Alongside similar metrics of comparative performance, we offer architectural analysis of the behavior of models in relation to the characteristics of IoT traffic and offer deployment-centric evaluation, including convergence behavior and validation dynamics. In such a way, the work outgrows the framework of incremental performance reporting and offers an analytically supported baseline of robust IoT anomaly detection studies.

3. Methodology and Implementation

3.1. Data Collection

In this paper, synthetic and real-world IoT network data were used to train and test the deep learning models. This synthetic data was collected through MATLAB simulation of several networks that presented different packet delivery rates (PDR), end-to-end delay, and bandwidth consumption under normal and stress conditions. Due to the synthetic nature of the dataset, the opportunity for training the model included anomalies such as packet drops, latency, and congestive network [23]-[29].

Besides synthetic data, the real data was collected from the IoT-23 dataset, which is available in the public domain. The IoT traffic; this one is split across both normal usage and malicious (attacked/broken into) states. The proposed models were trained with synthetic and real-world data in order to assess the reliability of the detection of both the known and unknown network disturbances. In the foregoing section, Fig. 1 describes the nature of both simulated IoT network data and real-world data [30]-[35].

3.2. Data Preprocessing

This study has observed that pre-processing is an essential step towards achieving high performance of deep learning models on IoT networks. Data cleaning, normalization, feature extraction, and division to allow raw data to be used for training and maintaining structured and clean data for learning [36]-[42].

a) Noise Removal: Before making predictions with the test set, any discrepancies or outliers that could be attributed to the network errors were left out. The dataset was cleaned to filter out any data that might interfere with the models' training process in the next section [43]-[48].

b) Handling Missing Values: There was some missing data, which was imputed and dropped if its absence was not deemed significant. But it also made sure that no stored record at all is left incomplete, which might affect the training of the model.

Figure 1. Illustration of synthetic and real-world IoT network data characteristics.

c) Normalization: To minimize issues of inconsistency in the dataset, all features were normalized using the Min-Max normalization method. This made all the values range between zero and one, something suitable for the deep learning models, especially to help avoid some of the problems with the gradients. The normalization formula used in the current study is presented by (1) below:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Where:

x is the original value

x' is the normalized value

$\max(x) - \min(x)$ These are the minimum and maximum values of the feature, respectively.

d) Feature Extraction: The data had also been preprocessed to reduce the dimensionality using methods including Principal Component Analysis (PCA). Another advantage of PCA, which is also connected with removing the features that do not contain enough information, is the increase in the model's speed when it is working [49]-[54].

3.3. Deep Learning Models

In this section, we present three deep learning algorithms for IoT network monitoring: Artificial neural networks include Feedforward Neural Network (FFNN), Convolutional Neural Network (CNN), and Multilayer Perceptron (MLP). These algorithms enhance real-time network performance management and identify anomalies.

3.3.1. Feedforward neural network (FFNN)

Feedforward Neural Networks (FFNNs), also known as single-path networks, use the information flow from the input layer to the hidden layer and then to the output layer. This architecture is robust yet canonical and thus allows them to solve various classification and regression tasks. This can connect to every neuron in the next layer; every connection has a weight that is found during the training phase. In each neuron, the computed output of neurons often equals the sum of the weights times the input, augmented by the bias term. It is represented mathematically as given in (2):

$$z = \sum_{i=1}^{n} w_i x_i + b \quad (2)$$

Where z is the weighted sum, w_i are the weights for each input, x_iRegarding the inputs, and b is the bias term. The weighted sum is then passed through an activation function, introducing non-linearity into the model. Common activation functions include the Rectified Linear Unit ($ReLU$), which outputs $a = max(0, z)$, and the Sigmoid function, which outputs ($a = \frac{\{1\}}{\{1 + e^{\{-z\}}\}}$), where a is the activation output. These activation functions enable the model to capture non-linear relationships within the data, making FFNNs highly versatile for various tasks.

Computationally, FFNNs are faster than other models and are good at solving problems that have identifiable data patterns. The architecture of FFNNs consists of input layers, a hidden layer, or two layers, which have $Tansig$ activation and output layer, which has a Linear activation. Fig. 2 provides the structure of the Feedforward Neural Network (FFNN) with an input layer, two hidden layers, and the output layer with weight assignment between the layers.

3.3.2. Convolutional neural network (CNN):

Convolutional Neural Networks (CNNs) are well-suited for processing data that originates in a grid, such as images, by detecting spatial hierarchies. They employ filters to find such features as edges or texture, which are then incorporated for use in subsequent layers. CNNs are especially suitable for space or time. The convolution operation is defined in (3), and Fig. 3 shows the general structures of CNN accordingly.

$$z_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} W_{m,n} * x_{i+m,j+n} + b \quad (3)$$

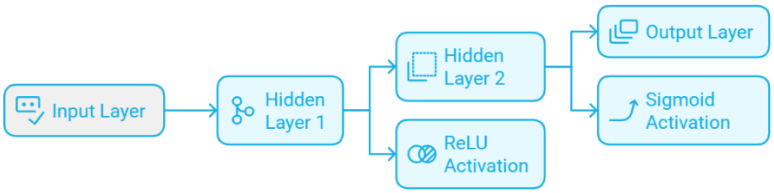

Figure 2. Architecture of the Feedforward Neural Network (FFNN).

Figure 3. Architecture of CNN and pooling layers.

3.3.3. Multilayer perceptron (MLP)

The Multilayer Perceptron (MLP) is a feed-forward neural network (FFNN with hidden layers, allowing for higher-order relationships in data. It employs non-linear functions of activation to manipulate input and represents difficult-to-identify complexities to linear models. MLPs are effective for tasks with nonlinear dependencies and are used across various areas. This can be expressed in (4), and the Multilayer Perceptron (MLP) architecture is depicted in Fig. 4.

$$z^I = W^I a^{I-1} + b^I \quad (4)$$

3.4. Training Process

Deep learning models are trained to maximize their performance on new data through methods of overfitting control. The study focuses on hyperparameters such as learning rate, batch size, epochs, regularization techniques, dropout, and weight decay. The learning rate is fixed at 0.001, allowing the model to converge efficiently without overshooting the best solution. The batch size is 32, considering memory usage and computational cost. Early stopping is used to prevent overfitting by pausing training if validation loss does not decrease for the subsequent five epochs. The common regularization methods are dropout (randomly train half of the neurons in the hidden layers) and weight decay (often referred to as L2 regularization) (minimizing overfitting by adding a quadratic penalty on parameter size).

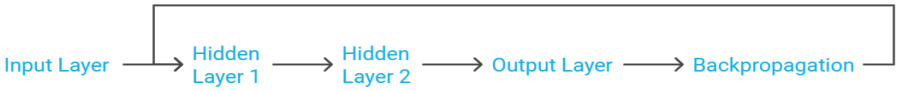

Figure 4. Architecture of the Multilayer Perceptron (MLP).

The analysis will help enhance the effectiveness of training in neural networks and accuracy. The models were trained in more than 50 epochs, with the performance measures taken at the end of every epoch. Figs. 5 and 6 demonstrate the training and validation accuracy of over 50 epochs for CNN, FFNN, and MLP models, while Fig. 5-7 showcases the accuracy over 50 epochs for MLP models.

4. Results and Discussion

This section describes the results of training and testing the DL models, namely FFNN, CNN, and MLP, and compares them with the machine learning models, namely DT and SVM. The models were evaluated concerning their accuracy, precision, recall, and F1 measure of network anomalies that are key to IoT networks' monitoring in real time.

4.1. Model Accuracy

Obviously, the CNN model made the highest accuracy score of 94% while the MLP scored 92.3% and the FFNN scored 90.1 %. The traditional machine learning algorithms like Decision Trees and SVM took the last two places, which are the worst accuracies of 78.5% and 85.7%. The ability to determine spatial dependencies in the traffic of IoT networks was the reason for the CNN's high throughput because it had a higher accuracy of identifying traffic and network anomalies than the other models. The accuracy of the models has been compared in Table 2 below.

4.2. Precision, Recall, and F1-Score

The performance was much better than the performance achieved by the CNN model for the different anomaly detection tests, with a precision of 93.7% and a recall of 92.5%. Finally, MLP and FFNN both yielded 91.2% and 89.5%, respectively, placing them second to last. Precisely, Decision Trees obtained efficacy ratings of 76.8, and Support Vector Machines were rated 83.9 in the traditional classification models. That is, the desired balance between precision and recall was reflected by the F1-score of 93.1 for the CNN model, meaning that the model was successful. Both MLP and FFNN also gave good results, with the MLP giving a F1-score of 91.6% and FFNN a F1-score of 89.9%. These are very important to understand the performance of models in anomaly detection jobs and to do so altogether. Table 3 deals with the precision, recall, and F1-score of models.

Table 2. Comparison of Model Accuracy.

Model	Accuracy (%)
Decision Tree (DT)	78.5
SVM	85.7
FFNN	90.1
MLP	92.3
CNN	94.0

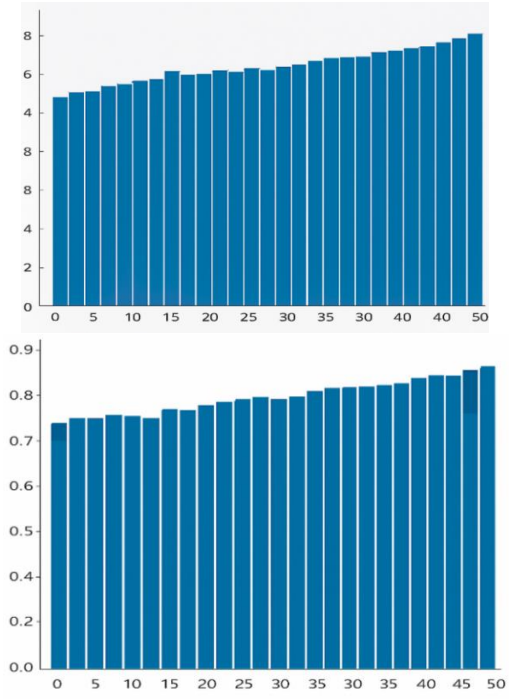

Figure 5. training and validation accuracy over 50 epochs for the CNN model.

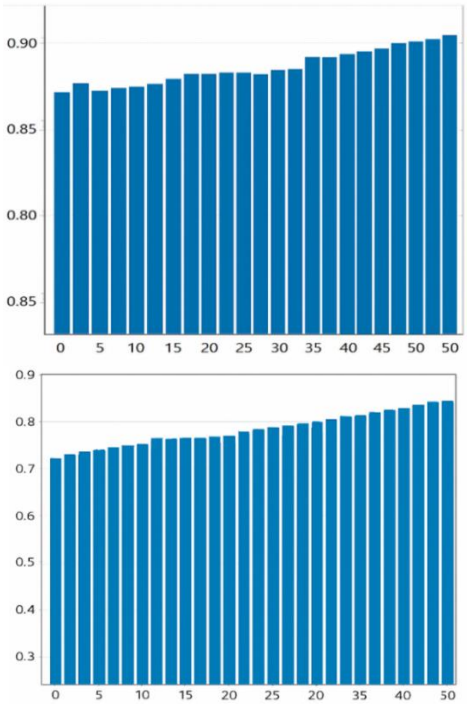

Figure 6. Training and validation accuracy over 50 epochs for the FFNN model.

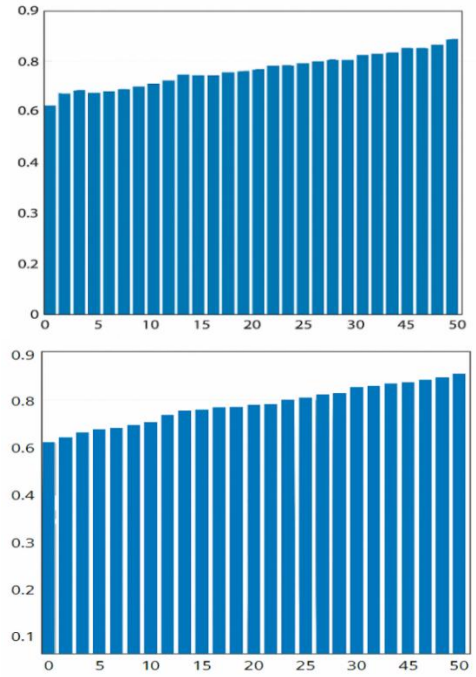

Figure 7. Training and validation accuracy over 50 epochs for the MLP model.

Table 3. Precision, Recall, and F1-Score of Models.

Model	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree (DT)	76.8	75.2	76.0
SVM	83.9	82.5	83.2
FFNN	90.3	89.5	89.9
MLP	92.0	91.2	91.6
CNN	93.7	92.5	93.1

4.3. Model Loss During Training

The categorical cross-entropy loss as a training measure also reduced from epoch to epoch for all the models, whereas CNN and MLP took fewer epochs to converge as compared to FFNN. Early stopping was also adopted to prevent overfitting during the training, while dropout was used while training the MLP for boosting generalization. A comparison of loss against the epochs is presented in Fig. 8 below for the basic FFNN, CNN, and MLP models.

The loss vs. epochs for three deep learning models: There are three types of Neural Network, which are feedforward neural network (FFNN), convolutional neural network (CNN), and Multilayer Perceptron (MLP). It expresses errors in the models, which gradually come down when the models are trained at some stage. The prediction on the x-axis means the number of epochs, which are cycles through the training data set, whereas the y-axis indicates loss. As observed, the loss function is more strained and decreases as the epoch models increase, with substantial progression at the initial epoch.

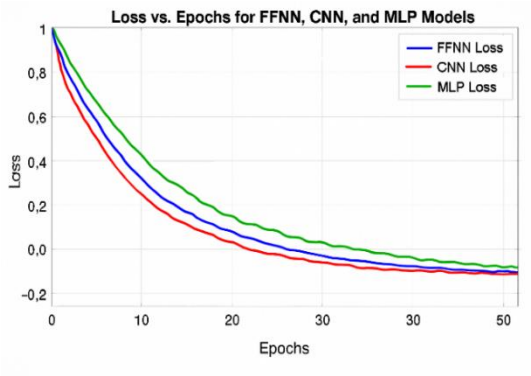

Figure 8. Loss vs. Epochs for FFNN, CNN, and MLP models.

Both FFNN and MLP require a comparatively higher loss in the initial epochs of training, while the CNN model converges to a similar loss as compared to FFNN and MLP after training up to 50 epochs. This means that, although CNN took a relatively long time to get optimized in the initial stages, it was comparable to the optimized policy learned by DQN after some additional training. From Fig. 3, we can observe that the loss function for both has dropped almost equally and constantly; However, in the first few iterations, it has dropped almost equally and constantly. However, in the first few iterations, the FFNN model had the highest drop. This plot demonstrates that each model reduces the prediction error by applying an optimization method, thereby attaining a low loss value at the termination of training.

4.4. Hyperparameter Tuning and Data Preprocessing

The hyperparameter tuning process significantly influenced the performance of the models. The learning rate, the batch size, and the number of neurons used in each of the hidden layers were tuned by grid search in each model. For instance, CNN was most effective with a learning rate of 0.001 and a batch size of 32, while MLP achieved optimal results with a slightly lower learning rate of 0.0005. Additionally, data preprocessing (such as normalization, noise removal, and feature extraction using PCA) contributed to the high performance of the models. Without proper preprocessing, the accuracy of all models decreased by approximately 5-8%. Fig. 9 shows accuracy vs. epochs plots for CNN, MLP, and FFNN models, while Fig. 10 shows model loss during training.

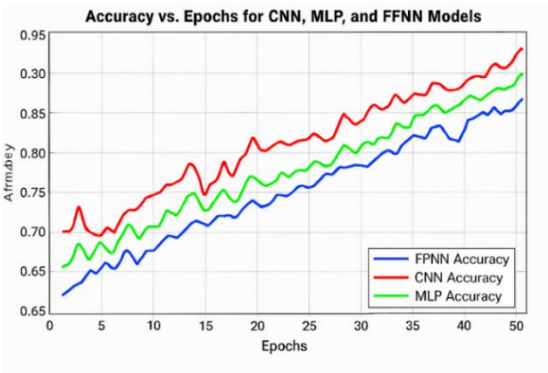

Figure 9. Accuracy vs. Epochs for CNN, MLP, and FFNN models.

Fig. 9 indicates the precision of three deep learning networks, i.e., Feedforward Neural Network (FFNN), Convolutional Neural Network (CNN), and Multilayer Perceptron (MLP) after 50 training cycles. The CNN model is the most accurate and reaches about 95 at the completion of the training stage. The MLP model comes next, just over 90, but the FFNN model begins at a lower level and steadily increases, to about 88 percent by the 50 th epoch. The graph shows how the models have progressive learning behavior, wherein CNN performs better in acquiring the spatial hierarchies in data. Both MLP and FFNN models show gradual and stable improvements, showcasing their ability to handle non-linear relationships within IoT network data.

Fig. 10 shows the loss vs. epochs for three deep learning models: FFNN, CNN, and face detection by a multi-layer perceptron from a data set of 50 epochs. All the models (CNN, FFNN, and MLP) exhibit a progressive decrease in the loss from the initial, which is higher in the case of CNN than in FFNN and MLP. After 50 epochs of training, the calculated loss values for all three models are similar, and the CNN and MLP models have a significantly lower final loss than the FFNN model. This graph also shows that the prediction error of each model decreases over time and is therefore useful in learning. What stands out more from this graph is that while total loss decreases with epochs, CNN and MLP undergo a steeper decline in loss in the initial epoch, suggesting they may develop a stronger ability to generalize knowledge learned from data when compared to FFNN.

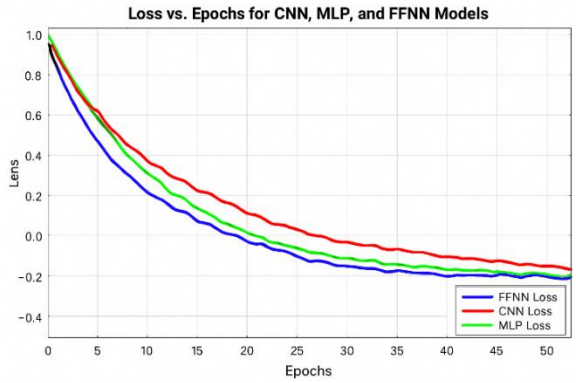

Figure 10. Model loss during training for CNN, MLP, and FFNN models.

4.5. Performance in Anomaly Detection

Among the main goals of the research, the improvement of anomaly detection in IoT networks was presented. The CNN model displayed outstanding superiority in detecting traffic congestion, dropped packets, and abnormal latency spikes. MLP was also useful in the identification of complex patterns and multi-dimensional relationships between network parameters. The FFNN model, despite being simpler, still had good results, especially when dealing with simple traffic patterns and device communication problems. The confusion matrix of both models showed CNN had the lowest rate of misclassification, meaning it produced fewer false positives and false negatives in cases where rare but important anomalies lie.

Fig. 11 and Fig. 12 show confusion matrices of CNN, MLP, and Fig. 13 FFNN models, respectively. The confusion of the CNN model indicates its classification of binary classes (0 and 1). The model accurately forecasted 33 cases in the category of class 0 (true negatives), 18 cases in the category of class 1 (false positives), 24 cases in the category of class 0 (false negatives), and 25 cases in the category of class 1 (true positives). In the matrix, the classification performance is somewhat balanced with certain misclassifications, especially in the prediction of class 0 into class 1.

It is seen in the confusion matrix of the Multilayer Perceptron (MLP) model, which shows that it is an accurate classifier. It correctly predicted 21 instances as class 0 (true negatives), 30 instances as false positives (true negatives), 25 instances as false negatives (true negatives), and 24 instances as true positives. However, the model misclassified 30 instances as class 1 and 25 instances as class 0 when they were class 1. This indicates that the MLP model had more difficulty in correctly classifying these categories compared to.

The confusion matrix in the case of the FFNN model of Binary classes is as follows. From the confusion matrix, it accurately identified 22 instances as class 0, 29 instances as class 1, 23 instances as class 0, and 26 instances as class 1, respectively. Regarding the prediction of class 1, the model had slightly better prediction than MLP, but had a higher false positive value.

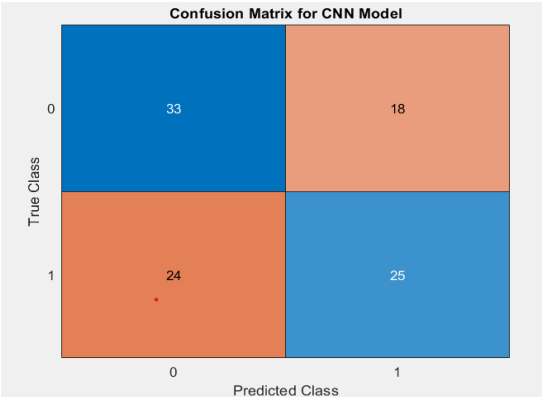

Figure 11. Confusion Matrix for CNN model.

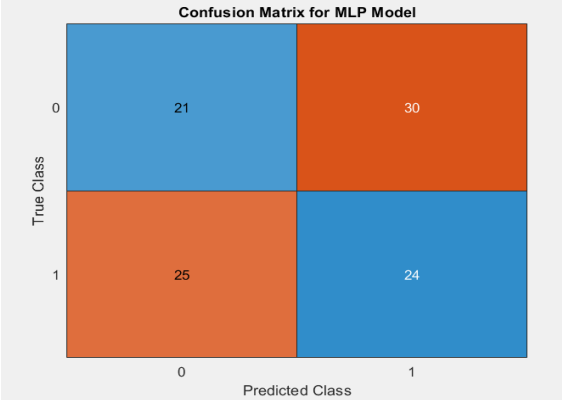

Figure 12. Confusion Matrix for MLP model.

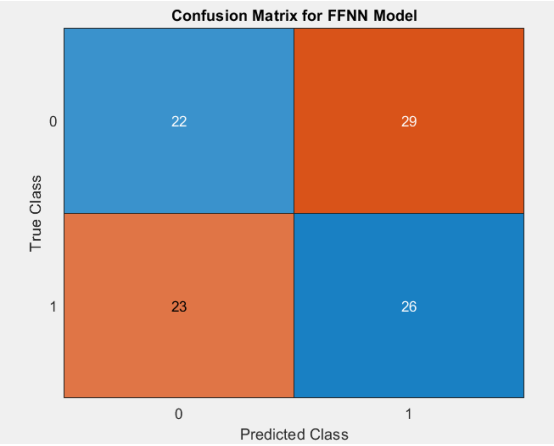

Figure 13. Confusion Matrix for FFNN model.

The synthetic IoT data has been produced to be reproducible and easy to understand methodologically through a network simulation framework written in MATLAB, which has been developed to simulate a realistic IoT communication setup. The simulated network was a combination of various IoT nodes relaying data to a central gateway with different loads of traffic. The steady rates of packet generation were considered as normal behavior of traffic with a constant packet delivery ratio (PDR), limited end-to-end delay, and nominal bandwidth consumption. These parameters were manipulated under predetermined rules to inject anomalous conditions of traffic. Precisely, the packet drop anomalies were created by setting the PDR to low values to simulate a failure of links and congestion, the latency anomalies were created by raising the end-to-end delay beyond the normal levels to model the routing or processing bottlenecks, and congestion anomalies were created by setting the traffic load and bandwidth usage higher than the limit of network capacity. All types of anomalies were introduced one at a time and in combinations to represent realistic multi-fault Internet of Things settings. The parameters that were manipulated through simulation include the parameters of packet rate, delay thresholds, levels of congestion, and the duration of anomaly to attain diversity and strength in the dataset. The resulting synthetic data were accordingly labeled and mixed with real-world data in the form of the IoT-23 traffic to allow both supervised training and generalization testing under varying conditions of the heterogeneous IoT operating conditions.

Fig. 14 Detailed experimental data and characterization of the dataset in the model of the detection of anomalies in the IoT. The figure consists of a single visual representation of both the properties of the dataset and the results of the model, in terms of evaluation. The top row shows the distribution of the classes, feature correlation heatmap, and two-dimensional projection of the data using principal component analysis, which shows balanced representation of the classes, inter-feature dependencies are moderate, and the samples can be separated reasonably between normal and abnormal samples. The bottom panels are the summaries of the comparative performance of the detailed models of DT, SVM, FFNN, MLP, and CNN. The bar chart reports the Accuracy, Precision, Recall, and F1-score of all the models, and indicates a higher performance of the deep

learning architecture. The confusion matrices also give further classification behavior in the sense that true positive, true negative, false positive, and false negative distributions of each model are given. Overall, the results confirm that CNN and MLP achieve more consistent and reliable anomaly detection performance compared to traditional machine learning approaches.

4.6. Discussion

This work proves that CNN and MLP models have high accuracy, validation, and anomalous activity detection of IoT traffic because of the models' complexity. As mentioned above, the CNN model is particularly good at the extraction of spatial hierarchies from data, such that it can identify features and patterns of the troubleshooting space and time dimension in very limited regions. The accuracy is always higher, and the loss is lower than that of the FFNN, which proves that the high ability of DL to learn the spatial hierarchies is crucial to enhancing the IoT network monitoring performance. The MLP model also produces generally favorable outcomes for efficiency of non-linear relationships in IoT data because of the multiple layers of hidden nodes with non-linear activation functions that can pick up patterns that may have been overlooked by more conventional models. This depth affirms MLP with more in-depth detection plans that may be harder for the other standard models to overlook.

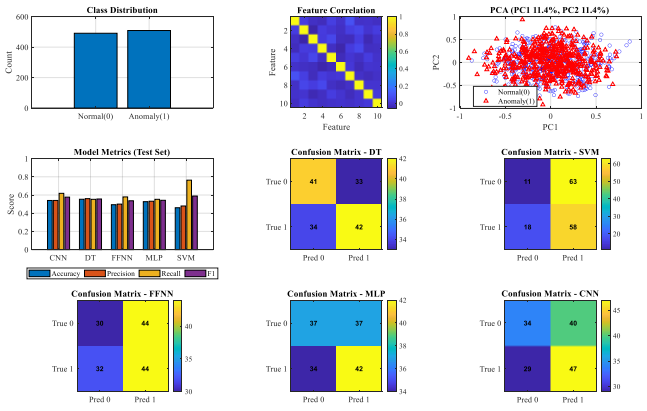

Figure 14. Unified Visualization of Dataset Characteristics and Comparative Performance of IoT Anomaly Detection Models.

Duly, hyperparameters are critical influencing factors of the deep learning algorithms' performance. Different hyperparameters, including learning rate, batch size, and the number of neurons on each hidden layer, cause significant effects on the general performance of all the models. It was recommended to combine Dropout regularization and L2 regularization for training deep learning algorithms, because the latter is prone to overfitting when trained with small data sets. In the case of anomaly detection, both the CNN and MLP models were outstanding compared to the FFNN in perceiving the anomalies in the network. This makes them more suitable for real-time IoT network monitoring, where one must get the idea that something is wrong before it begins to affect the network. CNN is an effective methodology for monitoring the IoT networks and subjecting MLP for anomaly detection and pattern recognition. The study also focuses on fine-tuning and regularizing to aim for better results than what the models give out. In Table 4, we are comparing the performance checklists like accuracy, precision, recall measurements, F1-measure, training time, and validation loss.

Although the results verify that CNN and MLP models are more accurate, they have high validation and can detect anomalous activity in IoT traffic when considering their model complexity and their capability to learn spatial and non-linear patterns, it is essential to assume that multiple challenges and limitations are present as well. Although both CNN and MLP models perform better than FFNNs, each of them is computationally more expensive, takes more time to train, and consumes more energy, which can be considered as restricting the use of resources in resource-restricted IoT environments. They are also extremely sensitive to hyperparameter tuning, and an incorrect choice of learning rate, batch size, or network depth may cause unstable training or worse performance. In addition, deep learning models tend to overfit, especially when trained with small or unbalanced data points, and regularization methods like Dropout and L2 regularization are needed, but not always sufficient. The other major challenge is that deep learning models have very low interpretability, such that it may be hard to fathom and explain decisions that are made under anomaly detection in security-sensitive IoT systems. Also, real-time monitoring of the IoT needs a low-latency response, and the complexity of CNN and MLP models can cause delays that can impact detection on time. Lastly, the high dynamism and continuously changing nature of the patterns of IoT traffic present challenges in generalizing the model that needs regular retraining and adaptation to sustain performance. The discussion of these issues is essential to increase the practical functionality of CNN and MLP models in the real-world IoT network monitoring systems.

Table 4. Summary of Model Performance for FFNN, CNN, and MLP.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (Epochs)	Validation Loss
FFNN	85.2	82.5	80.3	81.4	50	0.35
CNN	92.3	91.7	90.5	91.1	50	0.15
MLP	90.1	89.4	88.2	88.8	50	0.22

4.7. Compression With Another Model

This paper compares deep learning models like FFNN, CNN, MLP, with the common machine learning models like DTs, SVMs, and NB in the application of the monitor of the IoT network and anomaly detection. Decision Trees can be applied to most classification problems because they are easy to

explain, but they fail to capture the multitude of interactions between different entities in an IoT network. Since CNN and MLP are non-linear and hierarchical, we obtain superior accuracy and precision in comparison to DT models.

Multilayer perception (MLP) and Support Vector Machines (SVM) are efficient for use in the structured environment of the first-dimension features, but demonstrate poor performance in comparison with CNN due to the dynamic environment of the IoT networks. Nevertheless, the result shows that CNN and MLP give higher accuracy and fewer false positives than SVM for anomaly detection. Naive Bayes is easy to implement for classification and requires minimal computational resources, which makes it suitable for use in classification, but it is non-independent of features, which can be a significant drawback when used in complex tasks such as monitoring of IoT networks.

The accuracy of the MLP model is 77.0%, and the CNN model is 75.9%, and these models have relatively better improvement than Naive Bayes in the case of modeling spatial hierarchies. This is because CNN is more efficient in spatial hierarchies and MLP's non-linear modeling, in contrast to the occurrence of real-time network data anomalies. Naive Bayes cannot work on IoT complex and coupled features and has a higher false negative ratio than the deep learning models. Therefore, the DL models, such as CNN and MLP, are more effective than the other traditional ML models in IoT network monitoring and confirming their suitability in terms of processing the varying and changing IoT data. Table 5 reveals that CNN and MLP models outperformed Decision Trees, SVM, and Naive Bayes models in accuracy, precision, recall, and F1-score.

Table 5. Performance Comparison Between Deep Learning and Traditional Models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	year
DT [55]	78.5	76.8	75.2	76.0	2024
SVM [56]	85.7	83.9	82.5	83.2	2025
NB [57]	74.2	71.5	70.3	70.9	2024
CNN *	92.3	91.7	90.5	91.1	2026
MLP *	90.1	89.4	88.2	88.8	2026

We can see that the CNN and MLP models are better for the context of IoT network data, as it is more complicated and ever evolving. The feasibility of spatially related features processing by the CNN model and the deep learning ability of the MLP made it possible to achieve higher results in all the performance indicators than the traditional models. Some of the conventional Machine learning models include decision trees, SVM, Naïve Bayes, and many more, but these models are not very effective when it comes to handling complex interdependent and variable IoT networks. The good results of the deep learning models used in this study suggest that such models can be used to improve real-time IoT network monitoring and anomaly identification. With the further elaboration of the IoT networks and their complexity, the utilization of high-level deep learning architecture might become more urgent.

5. Conclusion

The research paper illustrates the possibility of deep learning models dramatically increasing the accuracy, reliability, and robustness of the monitoring systems of IoT networks. The comparative analysis of FFNN, CNN, and MLP models, and the conventional machine learning algorithms (Decision trees and SVM), on a single experimental platform shows that the deep learning algorithms offer more predictable and effective anomaly detection in complex IoT environments. CNN had the best accuracy (94%), as it was able to model spatial hierarchies and traffic patterns, whereas MLP had high performance (92.3%), as it was able to model nonlinear relationships between network parameters, and FFNN had a competitive architecture. In addition to the performance indicators, the results also reveal that reasonable data preprocessing and hyperparameter optimization have a significant role in obtaining reliable and stable detection results. However, practical limitations that are also recognized in the study are the computational overhead, the problem of scalability in large-scale implementations, and the lack of formal statistical significance tests that can influence the practical implementation. Application-wise, the findings reflect that CNN- and MLP-based solutions are suitable to run in real-time centralized or edge-assisted applications in an IoT-based environment. The future efforts will be dedicated to integrating time modeling with hybrid CNN/MLP-LSTM and RNN architectures to learn more about time-related traffic behavior and consider unsupervised models such as autoencoders or GANs to be more adaptable to new anomalies of IoT networks.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Author Contribution Statement

Radhi Sehen Issa and Gregor Alexander Aramice proposed the research problem.

Noorulden Basil: developed the theory and performed the computations.

Takele Ferede Agajie and Alfian Ma'arif verified the analytical methods and investigated [Conceptualization, Data curation] of this work.

All authors discussed the results and contributed to the final manuscript.

References

[1] K. Mehmood, P. Kautish, M. Rashid, Y. Joshi, and Y. Iftikhar, "Digitalization in the circular economy: synergistic impact of big data analytics, green internet of things, and ambidextrous green innovation," *J. Clean. Prod.*, vol. 509, p. 145610, 2025. doi: https://doi.org/10.1016/j.jclepro.2025.145610.

[2] M. E. E. Alahi *et al.*, "Integration of IoT-enabled technologies and artificial intelligence (AI) for smart city scenario: recent advancements and future trends," *Sensors*, vol. 23, no. 11, p. 5206, 2023. doi: https://doi.org/10.3390/s23115206 .

[3] G. Paolone, D. Iachetti, R. Paesani, F. Pilotti, M. Marinelli, and P. Di Felice, "A holistic overview of the internet of things ecosystem," *IoT*, vol. 3, no. 4, pp. 398–434, 2022. doi: https://doi.org/10.3390/iot3040022.

[4] M. Soori, B. Arezoo, and R. Dastres, "Internet of things for smart factories in industry 4.0, a review," *Internet Things Cyber-Physical Syst.*, vol. 3, pp. 192–204, 2023. doi: https://doi.org/10.1016/j.iotcps.2023.04.006.

[5] C. Nandhu *et al.*, "IoT-based vehicle tracking with accident alert system," *Int. J. Res. Publ. Eng. Technol. Manag.*, vol. 9, no. 2, pp. 486–494, 2026, https://doi.org/10.15662/IJRPETM.2026.0902001

[6] A. Haleem, M. Javaid, M. A. Qadri, R. P. Singh, and R. Suman, "Artificial intelligence (AI) applications for marketing: A literature-based study," *International Journal of Intelligent Networks*, vol. 3, pp. 119–132, 2022. doi: https://doi.org/10.1016/j.ijin.2022.08.005.

[7] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things*, vol. 19, p. 100564, 2022. doi: https://doi.org/10.1016/j.iot.2022.100564.

[8] D. Nelson *et al.*, "Advances in air quality modeling through artificial intelligence, machine learning, and deep learning: A comprehensive review," *Science of the Total Environment*, vol. 1021, p. 181593, 2026. doi: https://doi.org/10.1016/j.scitotenv.2026.181593.

[9] G. Yang, "An overview of current solutions for privacy in the Internet of Things," *Frontiers in Artificial Intelligence*, vol. 5, p. 812732, 2022. doi: https://doi.org/10.3389/frai.2022.812732.

[10] A. Hakiri, B. Sellami, and S. Ben Yahia, "Joint energy efficiency and network optimization for integrated blockchain-SDN-based internet of things networks," *Future Generation Computer Systems*, vol. 163, p. 107519, 2025. doi: https://doi.org/10.1016/j.future.2024.107519.

[11] E. Kritika, "A comprehensive literature review on ransomware detection using deep learning," *Cybersecurity Applications*, vol. 3, p. 100078, 2025. doi: https://doi.org/10.1016/j.csa.2024.100078.

[12] J. Lee, J. Kim, S. K. Yoo, T. Taleb, and J. Song, "Standardised Interworking and Deployment of IoT and Edge Computing Platforms," *Digital Communications and Networks*, vol. 11, no. 5, pp. 1578–1587, 2025. doi: https://doi.org/10.1016/j.dcan.2025.04.006.

[13] P. Li and L. Zhang, "Application of big data technology in enterprise information security management," *Scientific Reports*, vol. 15, no. 1, p. 1022, 2025. doi: https://doi.org/10.1038/s41598-025-85403-6.

[14] K. Zaghouani, B. Djamaa, A. Yachir, and S. Mahmoudi, "SemChain: A blockchain-based semantic discovery on distributed resource directories for the Internet of Things," *Internet of Things*, vol. 31, p. 101591, 2025. doi: https://doi.org/10.1016/j.iot.2025.101591.

[15] M. H. M. Noor and A. O. Ige, "A survey on state-of-the-art deep learning applications and challenges," *Engineering Applications of Artificial Intelligence*, vol. 159, p. 111225, 2025. doi: https://doi.org/10.1016/j.engappai.2025.111225.

[16] F. Tang, B. Mao, Y. Kawamoto, and N. Kato, "Survey on machine learning for intelligent end-to-end communication toward 6G: From network access, routing to traffic control and streaming adaptation," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1578–1598, 2021.doi: https://doi.org/10.1109/COMST.2021.3073009

[17] Y. Liu, H. Deng, and J. Fu, "DCM-Net: A novel dual-branch CNN–Mamba cross-layer feature fusion network for medical image segmentation," *Biomed. Signal Process. Control*, vol. 114, p. 109267, 2026.doi: https://doi.org/10.1016/j.bspc.2025.109267.

[18] A. Esmaeil Abbasi, Y. Ouazene, and M. P. Fanti, "Convolutional Neural Networks with Stable-Baselines for Optimized Path Planning by Simulation," *Applied Artificial Intelligence.*, vol. 40, no. 1, p. 2645993, 2026. doi: https://doi.org/10.1080/08839514.2026.2645993.

[19] S. Li, Q. Meng, X. Liu, Z. Wang, S. Kong, and B. Li, "MTF-NET: A mixed traffic flow multi-target detection network based on full-field perception and adaptive optimization," *PLoS One*, vol. 21, no. 3, p. e0344151, 2026. doi: https://doi.org/10.1371/journal.pone.0344151.

[20] E. Farrokh and D. Lotfi, "Decision tree analysis of cutter selection for tunnel boring machines: A study of geological conditions and machine types in high-performing TBM projects," *Tunnelling and Underground Space Technology*, vol. 162, p. 106612, 2025. doi: https://doi.org/10.1016/j.tust.2025.106612.

[21] S. Li *et al.*, "A multi-class support vector machine classification model based on 14 microRNAs for forensic body fluid identification," *Forensic Science International: Genetics*, vol. 75, p. 103180, 2025. doi: https://doi.org/10.1016/j.fsigen.2024.103180.

[22] G.-L. Ou, Y.-L. He, P. Fournier-Viger, and J. Z. Huang, "A Novel Multi-Source Weighted Naive Bayes Classifier," *Information Sciences*, vol. 721, p. 122568, 2025. doi: https://doi.org/10.1016/j.ins.2025.122568.

[23] U. Upadhyay, A. Dorn, C. Faber, and A. Schug, "From sequence to structure: A comprehensive review of deep learning models for RNA structure prediction," *Current Opinion in Structural Biology*, vol. 97, p. 103216, 2026.doi: https://doi.org/10.1016/j.sbi.2025.103216.

[24] J. Jordan, "The future of unmanned combat aerial vehicles: An analysis using the Three Horizons framework," *Futures*, vol. 134, p. 102848, 2021. doi: https://doi.org/10.1016/j.futures.2021.102848.

[25] W. Song, Z. Chen, M. Sun, and Q. Sun, "Observation reconstruction and disturbance compensation-based position control for autonomous underwater vehicle," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 377–387, 2022. doi: https://doi.org/10.1080/21642583.2022.2047124.

[26] P. Zarbipour, M. R. Nikoo, H. Akbari, R. Nazari, and M. Karimi, "Bridging Causality and Deep Learning for Harmful Algal Bloom Prediction," *Water Research*, vol. 294, p. 125492, 2026. doi: https://doi.org/10.1016/j.watres.2026.125492.

[27] Y. He, G. Ou, P. Fournier-Viger, and J. Z. Huang, "Attribute grouping-based naive Bayesian classifier," *Science China Information Sciences*, vol. 68, no. 3, p. 132106, 2025. doi: https://doi.org/10.1007/s11432-022-3728-2.

[28] H. A. L. Ouali, O. Abida, M. Essalhi, and I. Moukhtar, "Long short-term memory and metaheuristic algorithm-based deep learning model approach for forecasting green hydrogen production," *International Journal of Hydrogen Energy*, vol. 202, p. 152970, 2026. doi: https://doi.org/10.1016/j.ijhydene.2025.152970.

[29] B. Cao *et al.*, "Physics-guided deep learning for crop yield estimation," *European Journal of Agronomy*, vol. 172, p. 127850, 2026. doi: https://doi.org/10.1016/j.eja.2025.127850.

[30] W. Song, Z. Chen, M. Sun, and Q. Sun, "Reinforcement learning based parameter optimization of active disturbance rejection control for autonomous underwater vehicle," *Journal of Systems Engineering and Electronics*, vol. 33, no. 1, pp. 170–179, 2022. doi: https://doi.org/10.23919/JSEE.2022.000017.

[31] L. Huang, J. Xin, Y. Jiang, Y. Zhou, H. Zhang, and J. Zhou, "A novel physics-informed deep learning method for predicting arch bridge temperature-induced responses," *Engineering Structures*, vol. 348, p. 121750, 2026. doi: https://doi.org/10.1016/j.engstruct.2025.121750.

[32] L. Zeng, R. Dong, M. Yuan, L. Jing, and S. Jiao, "Evaluating deep learning time series models for PM2.5 forecasting across diverse horizons," *iScience*, vol. 29, no. 2, p. 114770, 2026. doi: https://doi.org/10.1016/j.isci.2026.114770.

[33] J. Yu, C. Chen, A. Arab, J. Yi, X. Pei, and X. Guo, "RDT-RRT: Real-time double-tree rapidly-exploring random tree path planning for autonomous vehicles," *Expert Systems with Applications*, vol. 240, p. 122510, 2024. doi: https://doi.org/10.1016/j.eswa.2023.122510 .

[34] J. He, Y. Zhao, S. Yang, W. Wu, J. Wang, and X. Deng, "Explainable spatiotemporal deep learning for subseasonal super-resolution forecasting of Arctic sea ice concentration during the melting season," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 232, pp. 1–17, 2026. doi: https://doi.org/10.1016/j.isprsjprs.2025.11.027.

[35] E. J. R. Freitas, M. W. Cohen, A. A. Neto, F. G. Guimarães, and L. C. A. Pimenta, "DE3D-NURBS: A differential evolution-based 3D path-planner integrating kinematic constraints and obstacle avoidance," *Knowledge-Based Systems*, vol. 300, p. 112084, 2024. doi: https://doi.org/10.1016/j.knosys.2024.112084.

[36] S. Pang *et al*., "Enhancing cloud detection across multiple satellite sensors using a combined Swin Transformer and UPerNet deep learning model," *Remote Sensing of Environment*, vol. 334, p. 115206, 2026. doi: https://doi.org/10.1016/j.rse.2025.115206.

[37] Q. Pu, X. Li, X. Wang, and Y. Li, "Cross-modal attention deep learning reveals how transformation products inherit life-cycle risks from parent antibiotics: Insights for environmental, ecological, health, and AMR risks," *Water Research*, vol. 295, p. 125584, 2026. doi: https://doi.org/10.1016/j.watres.2026.125584.

[38] M. Liu, H. Zhang, J. Yang, T. Zhang, C. Zhang, and L. Bo, "A path planning algorithm for three-dimensional collision avoidance based on potential field and B-spline boundary curve," *Aerospace Science and Technology*, vol. 144, p. 108763, 2024. doi: https://doi.org/10.1016/j.ast.2023.108763.

[39] C. Zhang, Y. Hou, X. Xia, J.-J. Chen, and Y. Pan, "Multimodal feature fusion deep learning for spatiotemporal prediction of deformation and environmental impacts in pipe-roof tunnel construction," *Advanced Engineering Informatics*, vol. 69, p. 104022, 2026. doi: https://doi.org/10.1016/j.aei.2025.104022.

[40] R. Kufakunesu, H. C. Myburgh, and A. De Freitas, "Fuzzy Logic-Based Data Flow Control for Long-Range Wide Area Networks in Internet of Military Things," *J. Sens. Actuator Networks*, vol. 15, no. 1, p. 10, 2026. doi: https://doi.org/10.3390/jsan15010010.

[41] I. Pacal, A. Algarni, and I. Kunduracioglu, "Adaptive and efficient deep learning model for automated ischemic stroke lesion segmentation," *Biomedical Signal Processing and Control*, vol. 118, p. 109658, 2026. doi: https://doi.org/10.1016/j.bspc.2026.109658.

[42] V. Marguet, C. K. Dinh, F. Stoican, and I. Prodan, "Indoor formation motion planning using B-splines parametrization and evolutionary optimization," *Control Engineering Practice*, vol. 152, p. 106066, 2024. doi: https://doi.org/10.1016/j.conengprac.2024.106066.

[43] S. Mudassar, A. Zameer, and M. A. Z. Raja, "GWO-DAGRU: A hybrid deep learning framework with metaheuristic feature selection and self-weighted context GRU for short-term wind power forecast," *Expert Systems with Applications*, vol. 310, p. 131279, 2026. doi: https://doi.org/10.1016/j.eswa.2026.131279.

[44] H. A. Nassrullah and Z. T. Alisa, "Implementing a Secure RPL Routing Protocol for IoT Networks using Lightweight Authentication Technique," *Baghdad Sci. J.*, vol. 23, no. 3, pp. 1040–1056, 2026. doi: https://doi.org/10.21123/2411-7986.5251.

[45] D. Sindhuja and R. S. Ravindran, "Blockchain-enabled genetic-inspired deep neural network for secure and efficient IoT offloading and routing," *Egyptian Informatics Journal*, vol. 33, p. 100927, 2026. doi: https://doi.org/10.1016/j.eij.2026.100927.

[46] J. Greffier, F. de Oliveira, S. Sammoud, J. P. Beregi, and D. Dabli, "Impact of a new deep-learning image reconstruction algorithm on potential dose reduction and quality of chest CT images: A phantom study," *Physica Medica*, vol. 142, p. 105736, 2026. doi: https://doi.org/10.1016/j.ejmp.2026.105736.

[47] N. H. Saeed, A. A. Hamza, M. A. Sobh, and A. M. Bahaa-Eldin, "Efficient feature ranked hybrid framework for android Iot malware detection," *Sci. Rep.*, vol. 16, no. 1, p. 3726, 2026. doi: https://doi.org/10.1038/s41598-026-35238-6.

[48] D. Zhou, F. Shao, J. Zhang, Y. Zhang, and S. Feng, "Multi-scale orthogonal Gabor filters based ConvNets for illumination robust single sample face recognition," *Eng. Appl. Artif. Intell.*, vol. 163, no. 5, p. 113143, 2026. doi: https://doi.org/10.1016/j.engappai.2025.113143.

[49] H. M. Hassan, "Metaheuristic-driven optimization of machine learning models for predicting principal dimensions of container ships," *J. Ocean Eng. Mar. Energy*, vol. 12, no. 1, pp. 459–483, 2026. doi: https://doi.org/10.1007/s40722-025-00439-0.

[50] S. P. Sudha and J. B. Loret, "A review on machine learning-based precision agriculture techniques for crop farming monitoring with IoT," *Discover Environment*, vol. 4, no. 1, p. 10, 2026. doi: https://doi.org/10.1007/s44274-025-00305-8.

[51] R. P. Singh, R. Dash, and R. K. Mohapatra, "Unveiling explainability in face anti-spoofing: Hybrid feature extraction with XAI-guided feature aggregation," *Pattern Recognit.*, vol. 169, p. 111905, 2026. doi: https://doi.org/10.1016/j.patcog.2025.111905.

[52] Z. Yu *et al.*, "Typomorphic and geochemical signatures of multi-colored fluorite: A proxy for magmatic-hydrothermal WSn and Mo systems," *J. Geochemical Explor.*, vol. 285, p. 108034, 2026. doi: https://doi.org/10.1016/j.gexplo.2026.108034 .

[53] R. Chen, P. Wang, X. Zheng, X. Rao, X. Zhang, and W. Wang, "Dynamic Simulation of Urban Agglomeration Network Resilience under Disturbances: An Integrated Deep Learning and Agent-Based Approach–Case Study of the Beijing-Tianjin-Hebei Region," *Sustain. Cities Soc.*, vol. 140, p. 107253, 2026. doi: https://doi.org/10.1016/j.scs.2026.107253 .

[54] M. K. Luka, O. U. Okereke, and E. C. Anene, "Hybrid path loss models using convolutional neural network and empirical models for path loss prediction in the ultra-high frequency," *Wirel. Networks*, vol. 32, no. 1, pp. 529–542, 2026. doi: https://doi.org/10.1007/s11276-025-04063-6 .

[55] E. Makiyah and N. N. Khamees, "VOXEL Video Streaming Over Wireless Networks," *Iraqi J. Inf. Commun. Technol.*, vol. 7, no. 2, pp. 1–13, 2024. doi: https://doi.org/10.31987/ijict.7.2.244.

[56] Y. Lee, S. O, and J. E. Eck, "Improving Recidivism Forecasting With a Relaxed Naïve Bayes Classifier," *Crime Delinq.*, vol. 71, no. 1, pp. 89–117, 2025. doi: https://doi.org/10.1177/00111287231186093.

[57] N. I. Khalaf and A. A. Kadhim, "5G Networks Based on Software Defined Network," *Iraqi J. Inf. Commun. Technol.*, vol. 7, no. 2, pp. 14–25, 2024. doi: https://doi.org/10.31987/ijict.7.2.262.