International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Enhanced IoT Network Security with Machine Learning Techniques for Anomaly Detection and Classification

Noman Abid*

1120 Mac Arthur Dr apt 1203 Carrollton tx 75007

Received 26 Nov 2023, Accepted 05 Dec 2023, Available online 15 Dec 2023, **Vol.13, No.6 (Nov/Dec 2023)**

Abstract

Network anomaly detection systems have grown in popularity and usefulness as a means to identify assaults, intrusions, and abnormalities in the ever-increasing volume of data sent by the internet and smart devices. Unusual patterns in network traffic may be reasonably anticipated using machine learning techniques. Nevertheless, the majority of prior efforts have been on anomaly detection inside conventional ML frameworks. Focussing on the IoT-23 dataset, which contains both harmless and malicious network traffic, this research explores the use of ML techniques for identifying and categorising anomalies in network security. Data preparation, feature engineering, model execution, and assessment are all a part of the technique. Models such as CNN, DT, LR, and SVM are employed to classify network anomalies, with performance evaluated using accuracy, precision, recall, and F1-score. Achieving a balance across parameters including F1-score, recall, and precision, the CNN model surpasses other models with an accuracy of 98.69%. Comparing the results, it can be concluded that CNN performs well aimed at anomaly detection, whereas Decision Trees offer a high level, and Logistic Regression and SVM are less accurate and stable. Since CNN is applied in deep learning, the study shows the effectiveness of deep learning models for network security in terms of anomaly detection with a suggestion to approach model optimisation to avoid overfitting and enhance generalisation.

Keywords: *Cybersecurity, IoT, Machine Learning, Network Security, Malicious Activities, Anomaly detection, Classification.*

Introduction

A rapid growth of the IoT has led to transformative changes across various industries, health care, transportation, manufacturing, and agriculture [1]. IoT integrates devices, sensors or systems by creating links that allow for the sharing of data and instant decision-making [2]. These interconnections serve productivity enhancement, process efficiency, as well as opportunity generation for new businesses [3]. On the other hand, network security is becoming more important as the number of IoT devices increases[4]. The larger the population of devices connected to the Internet, the larger the risk, which is why secure measures are becoming more critical [5][6][7].

Anomaly detection is an important function in establishing security within connected IoT networks [8]. Due to the number of devices and systems used, it is vitally important to provide detection of some abnormalities in data streams to notice potential threats or failures [9]. These anomalies may point to cybercrimes or attacks [10][11], or operational problems, such as system breakdowns [12].

*Corresponding author's ORCID ID: 0000-0000-0000-0000
DOI: https://doi.org/10.14741/ijcet/v.13.6.5

The quickly changing world of IoT threats may be too much for traditional security methods to handle. Because of this, sophisticated methods—especially those based on ML—have drawn a lot of interest due to their capacity to efficiently identify and categorise abnormalities in dynamic IoT environments[13].

The problems with IoT network security have a potential answer in ML algorithms [14]. ML models outperform more traditional approaches in detecting new risks, organising abnormalities, and identifying patterns in the massive amounts of data produced by IoT devices[15][16]. Supervised learning methods require examples of known threats to learn from[17], while unsupervised learning does not depend on examples of malicious behaviour [18][19][20]. The objective here is to evaluate the viability of these methods for improving IoT network security and early threat identification [21].

Aim and Contribution of Paper

Using the IoT-23 dataset to detect and reduce possible risks in IoT settings, this study attempts to improve IoT network security by creating and assessing ML methods for efficient anomaly detection and

categorisation. The study's contributions are as follows:

Uses the IoT-23 dataset to analyse network traffic, both malicious and benign.

Advanced preprocessing and feature extraction techniques are applied to enhance a relevance and efficiency of data, ensuring robust model performance.

Employing advanced encoding techniques, such as label encoding to transform categorical variables into numerical data.

Makes sure all features have an equal impact on the model by applying conventional scalar normalisation to feature scaling.

To implement the various models like CNN, Decision Tree, Logistic Regression, and SVM for anomaly detection.

Model evaluation with like accuracy, precision, recall, and F1-score ensures a granular understanding of model efficacy.

Developing scalable and reliable anomaly detection systems, enhancing IoT network security and resilience against evolving cyber threats.

Structure of paper

The following is the structure of the document's remaining sections: An examination of anomaly detection and classification's historical context is presented in Sections 2 and 3. The approach is described in Section 4. Section 4 compares the studies, analyses, and discussions. Section 5 presents the study's results as well as suggestions for further research.

Literature Review

In recent years, researchers have been more interested in the use of ML methods for anomaly detection and classification in order to improve the security of IoT networks. The following contains a few background studies:

This study, Roshan and Zafar, (2022) seeks to identify and elucidate abnormalities in networks using the XAI and kernelSHAP methods. This strategy is used to improve the f-score, accuracy, recall, and precision of the network anomaly detection model. The most recent CICIDS2017 dataset is used for the experiment. Two models, Model 1 and OPT Model, are built and then compared. After being trained in an unsupervised manner, the OPT Model achieves an overall accuracy of 0.90 and an F-score of 0.76 [22].

This study, Akoto and Salman, (2022) investigates the efficacy of ML and DL models in identifying and categorising breaches. When training and testing ML and DL models, make use of the publicly accessible CICIDS-2017 dataset. Three conventional ML models—LR, RF, and KN—as well as three deep learning models—1-D CNN, RNN, and a two-staged model that combines an ANN for classification with an unsupervised Dense Autoencoder (DAE) for pre-training—are used. Our findings show that among ML models, RF has the highest detection accuracy at 99.5%, while among DL models, DAE-ANN has the greatest performance at 98.7%. Lastly, find that RF achieves a higher detection rate of 91.35% compared to DAE-ANN's 84.66% [23].

In this study, Alqurashi, Shirazi and Ray, (2021) research how well a deep learning method called MLP can identify suspicious activity in ICS network data. They zero in on typical reconnaissance assaults that target ICS networks. An adversary's primary goal in such an assault is to learn as much as possible about the targeted network. A statistical ML method called isolation Forest (i Forest) is compared to MLP in order to assess our method. Our suggested deep learning method outperforms i Forest, which only achieves 75% accuracy, by more than 99%. All the more reason to believe in the potential of deep learning algorithms for detecting anomalies [24].

In this study, Malaiya et al., (2019) develop and evaluate deep learning models built using FCNs, VAEs, and Sequence-to-Seq (Seq2Seq) architectures. An anomaly detection network based on deep learning is feasible, according to our experimental findings, which show enhanced performance over traditional learning methods. In instance, the Seq2Seq with LSTM detection model shows great promise, as it reliably achieves 99% accuracy in identifying network abnormalities across all datasets used for assessment [25].

In this study, Ran, Ji and Tang, (2019) a ladder network-based deep learning method was suggested, which could properly classify attacks and identify network abnormalities by learning its own characteristics. And improving the model's discriminative capacity to categorise challenging data by employing focused loss as a loss function. Studies on the publicly accessible AWID have found four different types of network records: flooding assaults, injection attacks, impersonation attacks, and normal records. The overall accuracy of this study was 98.54%, with classification accuracies of 99.77%, 82.79%, 89.32%, and 73.41% for the four different categories of records [26].

This study Atefi, Hashim and Kassim, (2019) use the most recent dataset available for intrusion detection assessment, CICIDS-2017, to do anomaly analysis for categorisation purposes. This study used Deep Learning (DL) and KNN for ML and DNN, respectively, to perform anomaly analysis for classification purposes. An MCC-based classification performance for ML and DL is shown in one of the outcomes. When comparing the two classifiers, DNN stands head and shoulders above KNN with a score of 0.9293%. One of the most important things to do is use this research as a reference for creating IDS to better secure networked systems [27].

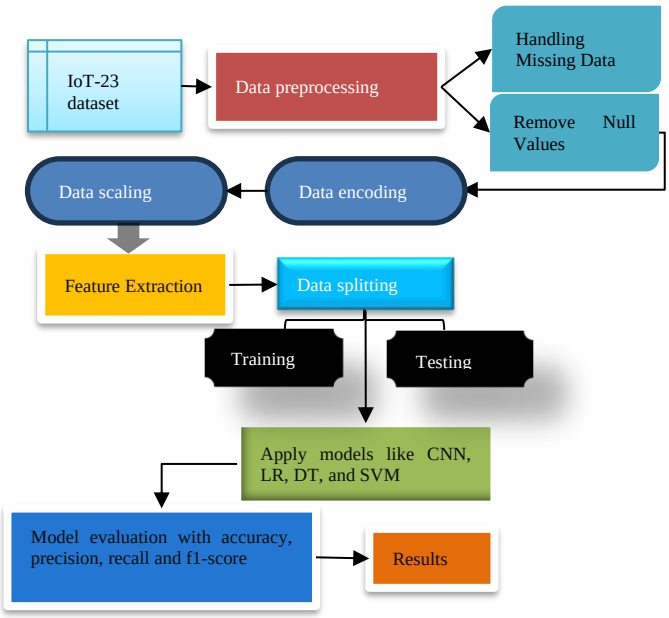
Table I summarises various studies on ML and DL approaches for IDS and network anomaly detection. It outlines the methods used, performance metrics, and key findings of each study. Additionally, it highlights the limitations and future directions for each research, suggesting areas for further exploration to improve detection accuracy and model efficiency.

Table 1 Summary of background study on Network Anomaly Detection using ML and DL approaches

Author	Methods	Data	Performance	Limitation/future study
Roshan and Zafar (2022)	XAI, kernelSHAP	CICIDS2017	OPT_Model: Accuracy = 0.90, F-score = 0.76	• Explore other XAI methods for improved accuracy. • Investigate model performance with larger datasets. • Test for real-time anomaly detection
Akoto and Salman (2022)	Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbor (KNN), 1-D CNN, RNN, DAE-ANN	CICIDS-2017	RF: 99.5% accuracy, DAE-ANN: 98.7% accuracy, RF better in categorisation (91.35% vs 84.66%)	• Focus on hybrid models. • Explore other deep-learning techniques for better performance. • Investigate multi-class classification in more detail.
Alqurashi, Shirazi, and Ray (2021)	Multi-Layer Perceptron (MLP), Isolation Forest (iForest)	ICS Network Traffic	MLP: >99% accuracy, iForest: 75% accuracy	• Compare MLP with other advanced deep learning techniques. • Test MLP on different datasets. • Investigate model scalability for large-scale data.
Malaiya et al. (2019)	Fully Connected Networks (FCN), Variational AutoEncoder (VAE), Seq2Seq with LSTM	Public datasets (varied)	Seq2Seq with LSTM: >99% accuracy for network anomaly detection	• Focus on real-time network anomaly detection. • Improve model efficiency for large-scale data. • Investigate hybrid architectures for enhanced accuracy.
Ran, Ji, and Tang (2019)	Ladder Network with focal loss	Aegean Wi-Fi Intrusion Dataset (AWID)	Overall accuracy: 98.54%, Classification accuracies: 99.77% (normal), 82.79% (injection), 89.32% (impersonation), 73.41% (flooding)	• Explore focal loss in other network attack datasets. • Improve classification of difficult samples. • Investigate efficiency in real-time scenarios.
Atefi, Hashim, and Kassim (2019)	K-Nearest Neighbors (KNN), Deep Neural Network (DNN)	CICIDS-2017	DNN: MCC = 0.9293, KNN: MCC = 0.8824	• Combine KNN and DNN for improved accuracy. • Explore the impact of different deep learning architectures. • Focus on cross-dataset generalisation.

Research Methodology

Data collecting, preprocessing, feature engineering, model implementation, and assessment are all part of the systematic approach that is used in the research methodology for Network Security using ML Techniques for Anomaly Detection and Classification. This data is mostly derived from the IoT-23 dataset, which includes both harmful and benign Internet traffic. Scaling features to guarantee consistency and relevance, encoding categorical data, resolving missing values, and deleting duplicates are all part of the preprocessing stages. Feature extraction techniques are employed to reduce dimensionality while retaining essential information. The dataset is split into training and testing subsets, with 80% for model training and 20% for performance evaluation. ML models like CNN, DT, LR, and SVM are implemented to classify network anomalies. Anomaly detection and classification are made robust and trustworthy by evaluating models using measures like Precision, accuracy, recall, and F1-score, with insights from confusion matrices. The following process of system implementation is shown in Figure 1.

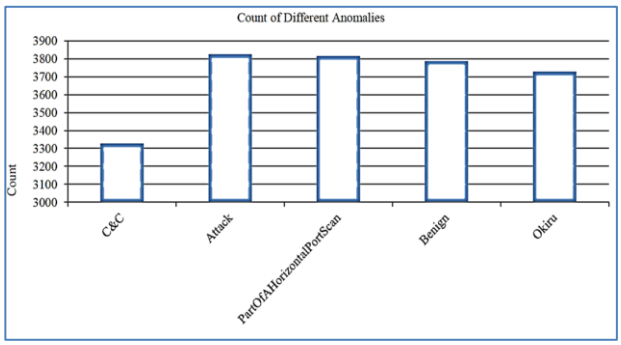


Flowchart Anomaly Detection and Classification

The following Figure 1 shows methodology steps and phases for anomaly detection that are explained below:

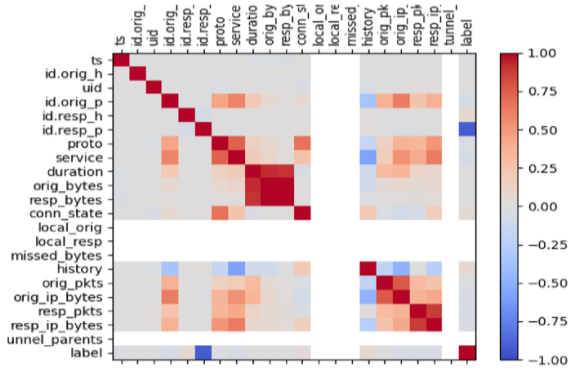
Data Collection

Data on network traffic from IoT devices has recently been compiled into the IoT-23 dataset. There is a total of twenty malware grabs, and three benign anomaly captures in the collection, all taken from various IoT PCs. The following anomalies are distributed in different category are shown in Figure 2.



Count of Dataset Anomalies

Figure 2 shows the Count of Dataset Anomalies, which illustrates the distribution of anomaly types within the dataset. The "Attack," "Part of A horizontal PortScan," and "Benign" categories dominate with the highest counts of 3800 each, followed closely by "Okiru" with 3700 occurrences. In contrast, "C&C" anomalies are the least frequent, with a count of 3300, highlighting a significant disparity in the representation of anomaly types in the dataset.



Correlation matrix of data

The correlation matrix in Figure 3 highlights strong positive relationships between id.orig_h and id.orig_p, as well as id.resp_h and id.resp_p, indicating close host-port associations. orig_bytes and resp_bytes correlate moderately with each other and with duration, suggesting longer connections involve more data transfer. missed_bytes shows a strong negative correlation with orig_bytes and resp_bytes, reflecting fewer missed bytes during higher traffic. Features like label, tunnel_parents, and history exhibit low correlations, implying limited impact on anomaly classification. These insights are useful for feature selection and dimensionality reduction, though further analysis is recommended.

Data Preprocessing

Preprocessing data involves cleansing the raw data from its unstructured state and transforming it into a well-structured dataset ready for further investigation [28]. "Data preprocessing" means cleaning the raw, unstructured data so it may be used in future research in an ordered and neat manner [29]. This section lays out the necessary pre-processing steps:

Remove null values: This process typically includes assessing the presence of null or NaN values in each column and deciding on appropriate strategies for imputation or removal.

Check Duplicate value: To provide accurate analysis findings and fair models, it is crucial to identify and eliminate duplicates.

Data Encoding

There is a need to transform categorical variables into numerical values as ML, and DL models operate on numerical quantities. Improved model performance is achieved by numerically representing categorical and special character values [30]. Use of label encoding and one-hot encoding techniques allows for the transformation of categorical data kinds into numerical information types [31]. A distinct numerical label is assigned to each category in label encoding, which transforms categorical data into numerical data. Label Encoder is a package provided by sklearn that may be used to convert numerical input from categories.

Data scaling

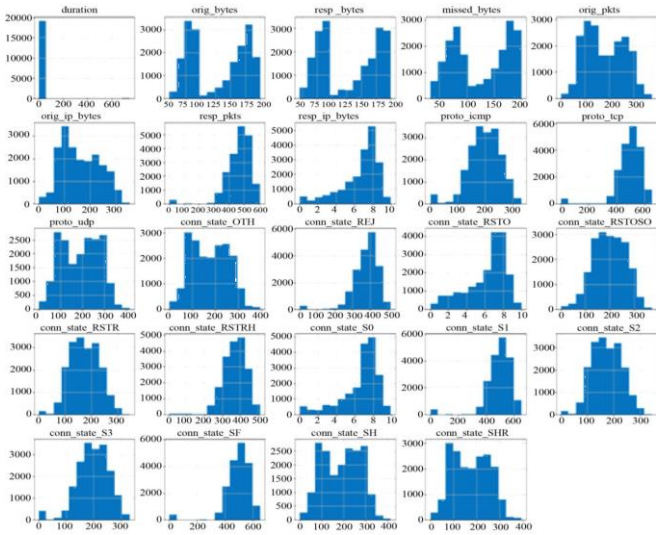
Making ensuring that no one feature has an excessive influence on the outcomes is the fundamental goal of feature scaling. It maintains the connection between each feature's lowest and maximum values [32]. Therefore, conventional scalar normalisation is used to rescale the features into a given scale. The features were rescaled using a typical scalar normalisation of 0 mean and 1 standard deviation [33]. A normalised value for the feature is obtained using the Equation in (1).

$$x_n = \frac{x-u}{\sigma} \quad (1)$$

where xn = normalized value, x = original value, μ = mean of data, and σ = data standard deviation.

Feature Extraction

The process of converting unprocessed input into numerical characteristics that ML algorithms may utilise is known as feature extraction in DL [34]. Effective data processing requires reducing the data's dimensionality, which feature extraction helps with. Feature extraction, then, is the process of developing new features that more effectively extract the relevant information from the source data while preserving its key characteristics[35].



Feature distribution of data

The histograms in Figure 4 reveal that most features in the dataset are right-skewed, with lower values being more frequent, which aligns with typical network traffic patterns where short-livv vvvved connections and smaller data transfers dominate. Protocol-related features (proto_tcp, proto_udp, proto_icmp) show TCP as the most common protocol, followed by UDP and ICMP. Connection state features vary in distribution, with conn_state_OTH displaying a more uniform spread, while states like conn_state_S1 are right-skewed, reflecting differences in connection behaviour.

Data Splitting

There are two sets of preprocessed data: one is used for training, and the other is for testing. The model is trained using the training set, which makes up 80% of the data, and its performance is evaluated using the testing set, which makes up 20% of the data.

Convolutional Neural Network (CNN) Models

CNNs are deep learning algorithms that identify patterns in pictures by using ANNs. There are often employed in image processing and identification applications [36]. A CNN is composed of layers, such as fully connected, pooling, and convolutional layers. An image's convolutional layer transforms it into numerical values, while the pooling layer lowers the input's parameter count. Among the deep learning algorithms, CNNs are especially well-suited for image processing and recognition applications [37]. The many layers that make it up include fully connected, pooling, and convolutional layers [38]. CNNs are highly adapted to identifying hierarchical patterns and spatial connections in images, and its design is inspired by visual processing in the human brain. It enables us to calculate a convolutional layer's output size [39][40]. The output in this instance has a length of 5. The output's length often follows (2),

$$\text{Output size} = nx = 2P - nhS + 1, \quad (2)$$

Output size = n x + 2 P − n h S + 1, where the input signal's length is denoted by nx. and nh represents the filter's length.

Mathematical operations like the convolution operation (Conv_Op) find widespread use in computer vision, signal processing, and image processing. A third signal, weighted by the form of the second signal, representing the effect of the first signal on the second, is produced by combining two signals or functions using it. The use of CNNs for feature extraction in computer vision is commonplace. As a mathematical operation, the convolution is defined as Eq. (3):

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] \quad (3)$$

Here, two possible continuous or discrete functions are f and g, and the output signal's location or time index is denoted by n. "∗" is the symbol for the convolution operation. The above Equation may be rewritten as (4) when dealing with discrete input signals:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]\Delta m \quad (4)$$

Here, the output signal's location or time index is denoted by n, and functions f and g might be either discrete or continuous. The convolution operation symbol is ∗. The above Equation may be rewritten as (5) when dealing with discrete input signals:

$$(f * m)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \quad (5)$$

where the output signal's time index is denoted by t.

Evaluation metrics

The prediction model concludes with this phase [41]. Some of the assessment metrics that can be used to evaluate the prediction findings in this section are; Classification accuracy, Recall, confusion matrix precision and F1 score. TN, FP, TP, and FN are the four metrics that rely on statistical data that emerge from a confusion matrix. Figure 5 displays a confusion matrix.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Representation of confusion metrics

True positive (TP): is a total number of successfully detected real positives.
True negative (TN): represents a total number of properly detected negatives

False positive (FP): represents a total number of false positives that were really ruled out.
False negative (FN): equals the sum of all the FP that were later shown to be false negatives.
Accuracy: Accuracy means the right predictions made in relation to the overall total number of predictions are produced to give an evaluation of the model. Here is the formula: (6):

$$Accuracy = \frac{TP+TN}{TP+Fp+TN+FN} \quad (6)$$

Precision: The category of the measure of precision is focused on the models' capabilities in defining their portion of FP as actual. Its Equation is (7):

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall: Recall is another measure of the performance of the model and relates especially to the quotient of detected true positive instances. Here is its formula: (8):

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

F1 score: The actual measures that are most commonly used are the harmonic mean of recall and precision or the F1 score. Here is the F1 Score range: [0, 1]. It tells you how accurate and reliable your classifier is. It is expressed mathematically as (9)-

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (9)$$

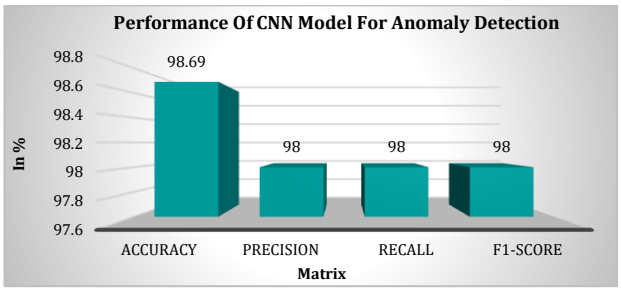
All these measures, when combined, explain levels of accuracy of the selected model in relation to the target variable.

Results and Discussion

For consistent computational performance, Google Colaboratory and Microsoft Windows 10 are chosen for this research. The configuration comprises an Intel Core i7 6850K processor running at 3.60 GHz with 12 cores, and an NVIDIA GeForce GTX 1080 Ti GPU equipped with 2760 4MB memory. Table II also shows the performances of the experiments made on IoT-23 dataset using various models including CNNs. A few of the evaluation matrices utilised in these studies involved f1-score matrix, precision matrix, accuracy matrix, and recall matrix. This section also compares the model performance with existing models like DT[42], LR[43], and SVM[44].

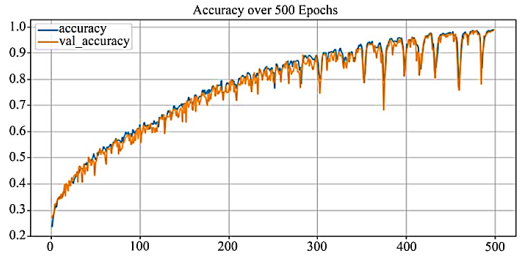
CNN model efficiency across performance matrix

Performance matrix	Convolutional Neural Network (CNN)
Accuracy	98.69
Precision	98
Recall	98
F1-score	98



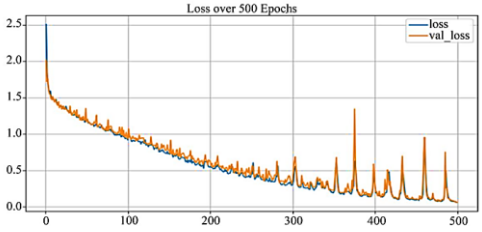
CNN model performance on IoT-23 dataset

The CNN model performance is shown in Table II and Figure 6. The correct result overall makes the model highly accurate with the average of 98.69% and recognising data anomalies. A CNN with a precision and recall of 98% is able to reliably detect the majority of real abnormalities and make accurate positive predictions. The F1 of 98 shows the best outcome where the model is able to perform equally well on precision and recall values; hence, it is good for trading off between the two. This means that the CNN could be best suited for cases where one wants to identify infrequent or obscure patterns with datasets, a premise that makes the CNN ideal for applications such as anomaly detection.



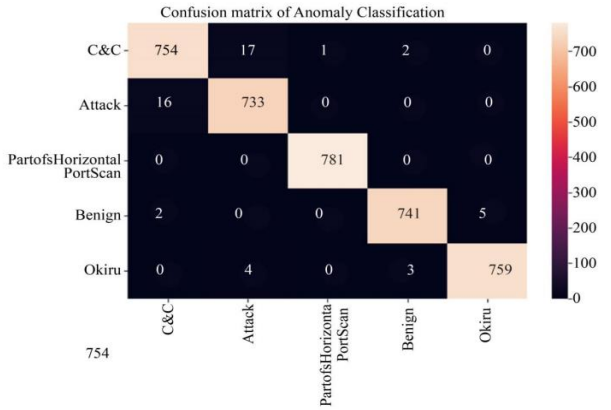
Accuracy curves for the CNN model

The blue line in Figure 7 shows the development of accuracy across 500 training epochs, whereas the orange line represents validation accuracy. Both measures show an initial spike, which indicates a period of fast learning. Subsequently, the accuracy continues to improve gradually, reaching a plateau of around 300 epochs. The validation accuracy, however, starts to diverge after this point, indicating potential overfitting. There has been no change in the model's performance on the training data, but it is becoming worse at generalising to new data (the validation set). As a result, methods for regularisation or early halting may be necessary to avoid overfitting and achieve superior generalisation.



Loss curves for the CNN model

The blue line in Figure 8 shows the training loss, and the orange line in Figure 8 shows the validation loss of a model across 500 epochs. Both metrics decrease rapidly in the initial epochs, indicating effective learning, and then gradually plateau as the model converges. The training loss and validation loss closely track each other for most of the training process, suggesting minimal overfitting. However, toward the later epochs (around 400–500), the validation loss shows periodic spikes, possibly due to fluctuations in model generalisation or data inconsistencies. Overall, the trend demonstrates effective training with some instability in validation performance at later stages.



Confusion Matrix for the CNN

The following Figure 9 shows a confusion matrix visualising the performance of an anomaly classification model across five classes: "C&C," "Attack," "PartofHorizontalPortScan," "Benign," and "Okiru." Each cell represents the number of instances classified as the predicted class (columns) compared to the actual class (rows). The diagonal cells containing high values indicate correct classifications: 754 for "C&C," 733 for "Attack," 781 for "PartofHorizontalPortScan," 741 for "Benign," and 759 for "Okiru." Off-diagonal values represent misclassifications, such as 17 "Attack" samples misclassified as "C&C" or 5 "Benign" samples misclassified as "Okiru." The overall matrix demonstrates strong classification performance with relatively low misclassification rates.

Comparison between ML and DL models for Anomaly Detection and classification

Model	Accuracy	Precision	Recall	F1-score
CNN	98.69	98	98	98
DT	96.3	92.7	96.3	94.5
LR	75	73	75	72
SVM	67	60	67	59

The following Table III provides the comparative analysis for anomaly detection. In terms of accuracy (98.69%) and balanced precision, recall, and F1-score (98 each), the CNN model stands head and shoulders above the competition, proving its resilience in capturing intricate patterns. The DT model follows with a commendable accuracy of 96.3% and a slightly lower precision of 92.7%, though it maintains a high recall of 96.3% and F1-score of 94.5%, indicating reliable performance but less precision than CNN. LR, with an accuracy of 75% and moderate precision, recall, and F1-score (73, 75, and 72, respectively), performs adequately but lags significantly behind the top models. The SVM model exhibits the lowest metrics, with 67% accuracy, 60% precision, 67% recall, and 59% F1-score, reflecting its limited ability to generalise in this context compared to the other algorithms.

Conclusion and Future Study

Anomaly Detection (AD) is an ML and data mining technique for identifying patterns. Behaviours or instances in data that are different or unusual from most other data. The goal is to discover samples that are inconsistent with expected behaviour, which may be anomalies or outliers. Non-conformance analysis can be very effective as it enables organisations to alert early losses or even potential risks. The results obtained illustrate that CNN outperforms other ML algorithms such as DT, LR, and SVM with an accuracy of up to 98.69% and an exceptional balance of precision, recall, and F1-score consistently. These results demonstrate that CNN is highly effective for detecting complex network anomalies. However, the main findings of the analysis are the possible problems of overfitting, which can be observed in training and validating accuracy curves after the epoch number. This makes it apparent that there is a bigger need to apply what they can call higher-level properties, such as regularisation or early stopping, to get even better generalisation. However, the use of CNN in network security applications is still highly effective due to the following general performance of CNN. Such studies could bring out other models and more tactics in order to increase the stability of the model in a real-world setting.

References

[1] J. Thomas, K. V. Vedi, and S. Gupta, “The Effect and Challenges of the Internet of Things (IoT) on the Management of Supply Chains,” *Int. J. Res. Anal. Rev.*, vol. 8, no. 3, pp. 874–879, 2021.

[2] R. Goyal, “The role of business analysts in information management projects,” *Int. J. Core Eng. Manag.*, vol. 6, no. 9, pp. 76–86, 2020.

[3] S. Zeadally and M. Tsikerdekis, “Securing Internet of Things (IoT) with machine learning,” *Int. J. Commun. Syst.*, 2020, doi: 10.1002/dac.4169.

[4] M. Gopalsamy, “Scalable Anomaly Detection Frameworks for Network Traffic Analysis in cybersecurity using Machine Learning Approaches,” *Int. J. Curr. Eng. Technol.*, vol. 12, no. 06, pp. 549–556, 2022, doi: : https://doi.org/10.14741/ijcet/v.12.6.9.

[5] K. V. V. and S. G. Jubin Thomas , Piyush Patidar, “An analysis of predictive maintenance strategies in supply chain management,” *Int. J. Sci. Res. Arch.*, vol. 06, no. 01, pp. 308–

317, 2022, doi: DOI: https://doi.org/10.30574/ijsra.2022.6.1.0144.
[6] R. Arora, S. Gera, and M. Saxena, "Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based ERP Applications," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2021, pp. 458–463.
[7] B. Patel, V. K. Yarlagadda, N. Dhameliya, K. Mullangi, and S. C. R. Vennapusa, "Advancements in 5G Technology: Enhancing Connectivity and Performance in Communication Engineering," *Eng. Int.*, vol. 10, no. 2, pp. 117–130, 2022, doi: 10.18034/ei.v10i2.715.
[8] R. Bishukarma, "Adaptive AI-Based Anomaly Detection Framework for SaaS Platform Security," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 07, pp. 541–548, 2022, doi: https://doi.org/10.14741/ijcet/v.12.6.8.
[9] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-Learning-Based Anomaly Detection for IoT Security Attacks," *IEEE Internet Things J.*, 2022, doi: 10.1109/JIOT.2021.3077803.
[10] Mani Gopalsamy, "An Optimal Artificial Intelligence (AI) technique for cybersecurity threat detection in IoT Networks," *Int. J. Sci. Res. Arch.*, vol. 7, no. 2, pp. 661–671, Dec. 2022, doi: 10.30574/ijsra.2022.7.2.0235.
[11] Mani Gopalsamy, "Enhanced Cybersecurity for Network Intrusion Detection System Based Artificial Intelligence (AI) Techniques," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 12, no. 01, pp. 671–681, Dec. 2021, doi: 10.48175/IJARSCT-2269M.
[12] G. Casolla, S. Cuomo, V. S. Di Cola, and F. Piccialli, "Exploring Unsupervised Learning Techniques for the Internet of Things," *IEEE Trans. Ind. Informatics*, 2020, doi: 10.1109/TII.2019.2941142.
[13] S. Krishnan, A. Neyaz, and Q. Liu, "IoT Network Attack Detection using Supervised Machine Learning," *Int. J. Artif. Intell. Expert Syst.*, 2021.
[14] V. S. Thokala, "Integrating Machine Learning into Web Applications for Personalized Content Delivery using Python," *Int. J. Curr. Eng. Technol.*, vol. 11, no. 06, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.9.
[15] M. Gopalsamy, "Artificial Intelligence (AI) Based Internet-of- Things (IoT) -Botnet Attacks Identification Techniques to Enhance Cyber security," *Int. J. Res. Anal. Rev.*, vol. 7, no. 4, pp. 414–419, 2020.
[16] K. Patel, "Quality Assurance In The Age Of Data Analytics: Innovations And Challenges," *Int. J. Creat. Res. Thoughts*, vol. 9, no. 12, pp. f573–f578, 2021.
[17] M. Gopalsamy, "Advanced Cybersecurity in Cloud Via Employing AI Techniques for Effective Intrusion Detection," *Int. J. Res. Anal. Rev.*, vol. 8, no. 01, pp. 187–193, 2021.
[18] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Appl. Soft Comput. J.*, 2018, doi: 10.1016/j.asoc.2018.05.049.
[19] M. S. Rajeev Arora, "Applications of Cloud Based ERP Application and how to address Security and Data Privacy Issues in Cloud application," *Himal. Univ.*, 2022.
[20] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic ," *Zenodo*, 2020.
[21] F. Cauteruccio *et al.*, "A framework for anomaly detection and classification in Multiple IoT scenarios," *Futur. Gener. Comput. Syst.*, 2021, doi: 10.1016/j.future.2020.08.010.
[22] K. Roshan and A. Zafar, "Using Kernel SHAP XAI Method to Optimize the Network Anomaly Detection Model," in *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom 2022*, 2022. doi: 10.23919/INDIACom54597.2022.9763241.
[23] J. Akoto and T. Salman, "Machine Learning vs Deep Learning for Anomaly Detection and Categorization in Multi-cloud Environments," *Proc. - 2022 IEEE Cloud Summit, Cloud Summit 2022*, pp. 44–50, 2022, doi: 10.1109/CloudSummit54781.2022.00013.
[24] S. Alqurashi, H. Shirazi, and I. Ray, "On the Performance of Isolation Forest and Multi Layer Perceptron for Anomaly Detection in Industrial Control Systems Networks," in *2021 8th International Conference on Internet of Things: Systems, Management and Security, IOTSMS 2021*, 2021. doi: 10.1109/IOTSMS53705.2021.9704986.
[25] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "An Empirical Evaluation of Deep Learning for Network Anomaly Detection," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2943249.
[26] J. Ran, Y. Ji, and B. Tang, "A semi-supervised learning approach to IEEE 802.11 network anomaly detection," in *IEEE Vehicular Technology Conference*, 2019. doi: 10.1109/VTCSpring.2019.8746576.
[27] K. Atefi, H. Hashim, and M. Kassim, "Anomaly analysis for the classification purpose of intrusion detection system with K-nearest neighbors and deep neural network," in *Proceeding - 2019 IEEE 7th Conference on Systems, Process and Control, ICSPC 2019*, 2019. doi: 10.1109/ICSPC47137.2019.9068081.
[28] S. Bauskar, "BUSINESS ANALYTICS IN ENTERPRISE SYSTEM BASED ON APPLICATION OF ARTIFICIAL INTELLIGENCE," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 04, no. 01, pp. 1861–1870, 2022, doi: DOI : https://www.doi.org/10.56726/IRJMETS18127.
[29] M. R. S. and P. K. Vishwakarma, "An Efficient Machine Learning Based Solutions for Renewable Energy System," *Int. J. Res. Anal. Rev.*, vol. 9, no. 4, pp. 951–958, 2022.
[30] V. N. Boddapati *et al.*, "Data migration in the cloud database: A review of vendor solutions and challenges," *Int. J. Comput. Artif. Intell.*, vol. 3, no. 2, pp. 96–101, Jul. 2022, doi: 10.33545/27076571.2022.v3.i2a.110.
[31] M. R. Kishore Mullangi, Vamsi Krishna Yarlagadda, Niravkumar Dhameliya, "Integrating AI and Reciprocal Symmetry in Financial Management: A Pathway to Enhanced Decision-Making," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 5, no. 1, pp. 42–52, 2018.
[32] V. V. Kumar, S. R. Yadav, F. W. Liou, and S. N. Balakrishnan, "A digital interface for the part designers and the fixture designers for a reconfigurable assembly system," *Math. Probl. Eng.*, 2013, doi: 10.1155/2013/943702.
[33] S. K. R. Anumandla, V. K. Yarlagadda, S. C. R. Vennapusa, and K. R. V Kothapalli, "Unveiling the Influence of Artificial Intelligence on Resource Management and Sustainable Development: A Comprehensive Investigation," *Technol. \& Manag. Rev.*, vol. 5, no. 1, pp. 45–65, 2020.
[34] V. K. Y. Nicholas Richardson, Rajani Pydipalli, Sai Sirisha Maddula, Sunil Kumar Reddy Anumandla, "Role-Based Access Control in SAS Programming: Enhancing Security and Authorization," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 6, no. 1, pp. 31–42, 2019.
[35] J. R. Sunkara, S. Bauskar, C. Madhavaram, E. P. Galla, and H. K. Gollangi, "Data-Driven Management: The Impact of Visualization Tools on Business Performance," *https//iaeme.com/Home/journal/IJM 1290 Ed. Int. J. Manag.*, vol. 12, no. 3, pp. 1290–1298, 2021.
[36] A. P. A. Singh, "Streamlining Purchase Requisitions and Orders : A Guide to Effective Goods Receipt Management," *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 5, pp. g179–g184, 2021.
[37] V. V. Kumar, F. W. Liou, S. N. Balakrishnan, and V. Kumar, "Economical impact of RFID implementation in

remanufacturing: a Chaos-based Interactive Artificial Bee Colony approach," *J. Intell. Manuf.*, 2015, doi: 10.1007/s10845-013-0836-9.

[38] S. A. and A. Tewari, "AI-Driven Resilience: Enhancing Critical Infrastructure with Edge Computing," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 02, pp. 151–157, 2022, doi: https://doi.org/10.14741/ijcet/v.12.2.9.

[39] P. Khare and S. Srivastava, "The Impact of AI on Product Management : A Systematic Review and Future Trends," vol. 9, no. 4, 2022.

[40] M. Z. Hasan, R. Fink, M. R. Suyambu, and M. K. Baskaran, "Assessment and improvement of intelligent controllers for elevator energy efficiency," in *IEEE International Conference on Electro Information Technology*, 2012. doi: 10.1109/EIT.2012.6220727.

[41] M. Z. Hasan, R. Fink, M. R. Suyambu, M. K. Baskaran, D. James, and J. Gamboa, "Performance evaluation of energy efficient intelligent elevator controllers," in *IEEE International Conference on Electro Information Technology*, 2015. doi: 10.1109/EIT.2015.7293320.

[42] D. R. Thamaraiselvi and S. Anitha Selva Mary, "Attack and Anomaly Detection in IoT Networks using Machine Learning," *Int. J. Comput. Sci. Mob. Comput.*, vol. 9, no. 10, pp. 95–103, 2020, doi: 10.47760/ijcsmc.2020.v09i10.012.

[43] L. Gotsev, M. Dimitrova, B. Jekov, E. Kovatcheva, and E. Shoikova, "A cybersecurity data science demonstrator: Machine learning in IoT network security," in *25th World Multi-Conference on Systemics, Cybernetics and Informatics, WMSCI 2021*, 2021.

[44] N. A. Stoian, "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set," *Univ. Twente*, 2020.