SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I8P119

Volume 11 Issue 8, 185-199, August 2024

Original Article

Deep Learning Model with Normalized Bayesian Optimizer for Anomaly Classification in IoT Security

Jisha Jose[1], J. E. Judith[2]

[1,2]*Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Tamil Nadu, India*

[1]*Corresponding Author : jishajose24@outlook.com*

Received: 12 June 2024 Revised: 26 July 2024 Accepted: 13 August 2024 Published: 31 August 2024

Abstract *- The Internet of Things (IoT) stands as a pivotal facilitator of technology, garnering considerable attention from the global scientific community. Nevertheless, the proliferation of IoT devices and the vast amounts of data they accumulate pose a significant vulnerability to an array of security threats and susceptibilities. The growing adoption of IoT infrastructure has given rise to challenges, including node failures, heightened threats, and increased susceptibility to attacks, anomalies, and potential security breaches. Addressing and mitigating these issues constitute a pivotal domain within the overarching realm of IoT. The current study presents a novel approach for anomaly classification in IoT Security by employing a Convolutional Neural Network (CNN) with feature optimization via the Normalized Bayesian Optimization Algorithm (NBOA). This research strives to enhance anomaly detection accuracy beyond the limitations of conventional CNN models. The proposed CNN is trained using meticulously optimized features extracted from the extensive IoT23 dataset, which is provided in CSV format. The utilization of this dataset results in notably superior performance, contributing to the overall effectiveness of the proposed anomaly detection approach. The proposed model attains an impressive accuracy of 98.69%, outperforming standard CNN models. The methodology entails leveraging NBOA for fine-tuning feature selection, thereby augmenting the model's ability to discern anomalies based on input values.*

Keywords *- IoT, Deep Learning model (DL), Bayesian optimization, Anomaly detection, Classification.*

1. Introduction

The IoT refers to a network of interconnected devices and objects that communicate and share data seamlessly over the internet [1]. These devices collect and exchange information to enable intelligent and automated decision-making [2]. IoT technology spans various domains, including smart homes [3], healthcare [4], industrial automation [5], and more, revolutionizing how to interact with the surroundings and enhancing efficiency through the integration of digital connectivity. It has emerged as a transformative technology, capturing substantial attention from the global scientific community. Its impact on daily life is profound, linking physical and virtual devices to deliver remarkable advantages, including enhanced automation, control, heightened productivity, real-time information access, and increased efficiency.

Figure 1 displays various applications associated with the IoT. The widespread adoption of IoT devices presents new challenges, particularly the susceptibility to anomaly attacks [6]. Anomalies in IoT data streams, indicating malicious activities or system malfunctions, underscore the critical need for early detection to ensure the security of IoT ecosystems. Conventional detection techniques, including Machine Learning (ML) with both supervised [7] and unsupervised algorithms [8], are instrumental in identifying anomalies. Supervised learning uses labelled datasets [9], while unsupervised learning adapts to evolving threats without predefined labels.

Anomaly classification [10] is crucial because of the complex as well as the dynamic behaviour of the IoT environment, providing a nuanced understanding of irregularities and enabling timely responses to potential security threats. This classification is essential for organizations aiming to mitigate risks and maintain the seamless operation of their IoT networks.

In this study, a methodology was proposed for anomaly classification in IoT security, employing a CNN [11]. Firstly, a multiclass classifier is proposed tailored to categorize anomalies within the IoT Security domain, leveraging a dataset comprising 24 features (IoT23 Dataset). Then, DL techniques are applied through the use of a CNN, allowing the model to discern intricate patterns within the data. Next, the features fed into the CNN are optimized with precision through the implementation of the NBOA [12].

Fig. 1 Applications of IoT

This optimization step improves the model's capacity to grasp and comprehend the distinctive attributes of IoT data, contributing to a more effective and accurate anomaly classification system. Through this innovative combination of CNN architecture and NBOA, the major aim is to advance anomaly detection capabilities in IoT Security, providing a robust solution to the challenges posed by evolving security threats. The proposed work on classification in the complex and dynamic realm of IoT security has three main contributions:

- An efficient DL model for anomaly classification in IoT security which addresses the challenge of detecting and classifying novel network attacks that evade traditional detection methods.
- The efficiency associated with the recommended strategy in attaining precision and effective classification of typical and attack data is highlighted through contrasting the offered approach with prior studies.
- It identifies research gaps in the literature and provides future scopes for further investigation, particularly in

exploring diverse data preparation methods, feature extraction techniques, and deep learning approaches.

The remaining sections of the document are organized as follows: In Section 2, the literature review and research gaps are provided. The methodology is highlighted in Section 3. The detailed results of the proposed method and the comparative study with existing research work are presented in Section 4. Finally, the conclusion and future scopes are discussed in Section 5.

2. Literature Review

Chakraborty et al. [13] launched a rule-based deep neural network for detecting and classifying novel network attacks, specifically addressing challenges posed by polymorphic techniques. The model, tested on the CICIDS 2017 dataset, showed impressive results. However, a limitation is the absence of testing across diverse network traffic classes and novel attack types, leaving the evaluation of its attack prediction capability incomplete. The study also did not explore the application of RNN/LSTM for handling anomalies in time series IoT traffic data, limiting the comprehensiveness of the anomaly detection methodology. Vishwakarma et al. [14] Outlined is a two-stage Intrusion Detection System (IDS) for IoT, employing Naive Bayes for data categorization and an unsupervised elliptic envelope for subsequent classification. The model attained an accuracy of 97% on the NSL-KDD dataset and 86.9% on the UNSW_NB15 dataset. However, limitations such as multiclass classification challenges, feature engineering issues, and the absence of real-time deployment with preventive action capabilities may hinder the practical applicability of the proposed IDS in complex and time-sensitive security scenarios.

Shah et al. [15] suggested a system paradigm for IoT security based on AI, employing LSTM, GRU, and ANN on the X-IIoT dataset, showcasing a comprehensive methodology and robust results. A limitation of this study is the absence of an analysis of the AI model's security performance against adversarial attacks on the IoT system, leaving potential vulnerabilities unexplored and unaddressed in the current research. Shami et al. [16] brought up an innovative anomaly detection technique designed for IoT devices at the host level, leveraging system call data and dynamic Markov chains to overcome segmentation and fixed threshold challenges. Evaluated on UNM and PiData datasets, the approach demonstrated impressive results with low false positives, high accuracy, and a notable F1 score, providing an efficient solution for syscall-based anomaly detection in IoT. However, limitations include the lack of exploration into the effectiveness of predetermined probability thresholds and fixed-length segmentation, with the study lacking empirical evidence or evaluation results. Additionally, the evaluations mention a lack of specific details on the datasets, potentially limiting the findings' generalizability.

Awajan et al. [17] suggested an IoT IDS based on DL with a fully connected, four-layer network, achieving impressive results: accuracy, precision, recall and an F1-score of 93.74%, 93.71%, 93.82% and 93.47%, respectively. It effectively addressed network heterogeneity, offering a scalable, platform-independent framework. However, limitations include the absence of a developed platform-independent framework from the system and a lack of exploration into the efficiency of a lightweight version, leaving potential improvements unexplored. Bhavsar et al. [18] presented the PCC-CNN model for IoT IDS, outperforming traditional ML in NSL-KDD, CICIDS-2017, and IOTID20 datasets. Limitations include the absence of real-time datasets, exploration of anomaly detection pipelines, and overlooking imbalanced attacks and data handling techniques. Vigoya et al. [19] proposed CIDAD, a CoAP-IoT anomaly detection dataset, achieving notable results but limited by dataset reliance and a focus on shallow ML models, suggesting potential for future research expansion.

Mukherjee et al. [20] utilized supervised ML models on a 350 K dataset to predict anomalies in IoT systems, achieving an average accuracy for the entire dataset, showcasing the effectiveness of LR, DT, RF, and ANN in identifying threats and anomalies in smart devices, with a novel application on the Ds2OS dataset, emphasizing the need for further research to understand the correlations between IoT networks and attacks for enhanced prevention strategies. A limitation of this study is the lack of exploration into the varying operation of micro-services in the IoT network at different times, potentially leading to anomalies due to fluctuations in typical behavior within IoT services. Odeh et al. [21] laid out an innovative ensemble-based DL approach, incorporating LSTM, GRU, and CNN methods for anomaly detection, achieving outstanding accuracies for CNN-LSTM and CNN-GRU, respectively, using the NSL-KDD dataset and showcasing the effectiveness of this combined framework in promptly and precisely detecting anomalies.

Yazdinejad et al. [22] launched a powerful ensemble DL model for cyber threat hunting in imbalanced IIoT datasets, achieving exceptional accuracies. The study's limitations include the inability to identify anomalies and their specific locations in the system for enhanced protection of data processing and analysis, potentially leading to serious failures or shutdowns in the Industrial IoT. Additionally, the work does not explore multi-view approaches to optimize accuracy, presenting a potential avenue for future investigation. Hazman et al. [23] introduced the IDS-SIoEL framework for IoT-based smart environments, employing Ensemble Learning with an optimal anomaly detection model

using AdaBoost, Boruta, mutual information, and correlation feature selection techniques, achieving impressive performance in accuracy, 33.68 seconds for learning, and 0.02156 seconds for detection on IoT-23. A limitation of this study is the exclusion of multi-class classification and an IDS model utilizing DL algorithms.

Alwasel et al. [24] proposed integrating graph theory with ML for aberration detection in industrial IoT, showcasing improved classification accuracy for LR, support vector machines, and K-means clustering. However, a limitation lies in the need for further investigation to optimize strategies and interactions between graph theory and network data analysis. This includes exploring diverse data preparation methods, feature extraction techniques, and ML approaches to achieve accurate and efficient classification of normal and attack data. Elsayed et al. [25] suggested that the ToN-IoT dataset, a Guaranteed Automated Two-level IDS utilizing an improved Long Short-Term Memory system, performed better in terms of accuracy, detection rate, and precision, thereby addressing security threats in IoT and SDN with remarkable effectiveness. The limitation of this study lies in its current state, lacking refinement through feature selection, inability to detect zero-day attacks on IoT systems, and not yet being tested in real-world IoT networks composed of mobile devices.

Saba et al. [26] proposed a CNN-centric method for anomaly-based Intrusion Detection Systems (IDS) in IoT, applying it to the NID and BoT-IoT datasets and achieving notable accuracy levels, addressing critical security challenges and demonstrating the effectiveness of DL in enhancing IoT security. The limitation of this study lies in the need for further research and improvements in IoT to improve threat detection rates, highlighting the need to construct and create security protocols both inside and outside of IoT equipment, with the goal of creating techniques utilizing a variety of algorithms, including numerous DL algorithms. K.-H et al. [27] suggested IMIDS, CNN-based IDS for IoT devices, achieving an F-measure of 97%, outperforming competitors, and addressing the training data shortage issue by employing a novel attack data generator. Yet, the limitation lies in the evaluation of two popular IDS datasets without exploring real-world IoT scenarios.

Shafiq et al. [28] explored the transferability of a pretrained autoencoder model across IoT devices, achieving an average improvement in detection accuracy for Mirai, there was a percentage of 9.52%, while for Bashlite, it stood at 44.59%, showcasing the most significant enhancement observed at 26.68% and 73.00%, respectively in transfer learning, using collections comprising seven devices affected by Mirai and nine devices affected by Bashlite, yet limitations include the lack of deep relationship among static features of IoT devices and their typical traffic patterns, suggesting the need for more representative feature datasets. Salman et al. [29] proposed a framework utilizing ML for the identification of IoT devices, classification of traffic, and detection of malicious network activity, attaining as much as 94.5% accuracy in identifying device types, 93.5% accuracy in classifying traffic types, and 97% accuracy in detecting anomalous traffic using a diverse dataset, predominantly highlighting the effectiveness of RF, yet acknowledging the challenge of detecting unknown traffic types and the potential impact of tunneling and anonymization on classification accuracy. Ullah et al. [30] presented a comprehensive exploration of DL models, including LSTM, BiLSTM, GRU, and hybrid architectures, achieving high performance parameters on seven datasets, demonstrating promising results while acknowledging the need for further investigation into optimization techniques for small datasets and the exploration of ensemble methods.

Research in IoT security and anomaly detection often lacks robust anomaly classification, hindering practical applicability. Additionally, studies commonly overlook comprehensive evaluations across diverse IoT scenarios and attacks, limiting the generalizability and effectiveness of proposed detection methods. Bridging these gaps is crucial for enhancing real-world applicability and developing robust IDS for IoT security.

3. Materials and Methods

3.1. Dataset Description

The IoT-23 dataset is a recent compilation of network traffic data originating from IoT devices. It was created in collaboration with Avast Software in Prague and captured at the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. The features meticulously optimized and utilized for training CNN are derived from the comprehensive IoT23 dataset, specifically formatted in CSV for easy accessibility and analysis. The dataset is categorized into twenty captures, representing network traffic from infected IoT devices. Each capture is labeled with the malware sample's name executed during the scenario. Additionally, three captures depict benign network traffic from real IoT devices, labeled with the respective devices' names. It is noteworthy that every malevolent situation involves executing a particular virus example on a Raspberry Pi, utilizing different procedures as well as performing diverse activities. Figure 3 visually presents a sample dataset, offering a graphical representation of the data under consideration.

The dataset is marked with labels that describe the type of network flow categorization through manual examination of the network. Labels include "Attack," signifying an attempt from the compromised device to infiltrate another host; "Benign," indicating no suspicious or malicious activities; "C&C" (Command and Control), denoting a onnection to a command-and-control server; "Okiru,"

denoting characteristics of Okiru botnet; and "PartOfAHorizontalPortScan," signifying connections used for gathering information through a horizontal port scan for potential subsequent attacks. These labels are assigned based on payload and behavioral analysis, enriching the dataset with valuable insights for the development and evaluation of ML algorithms targeting IoT malware detection.

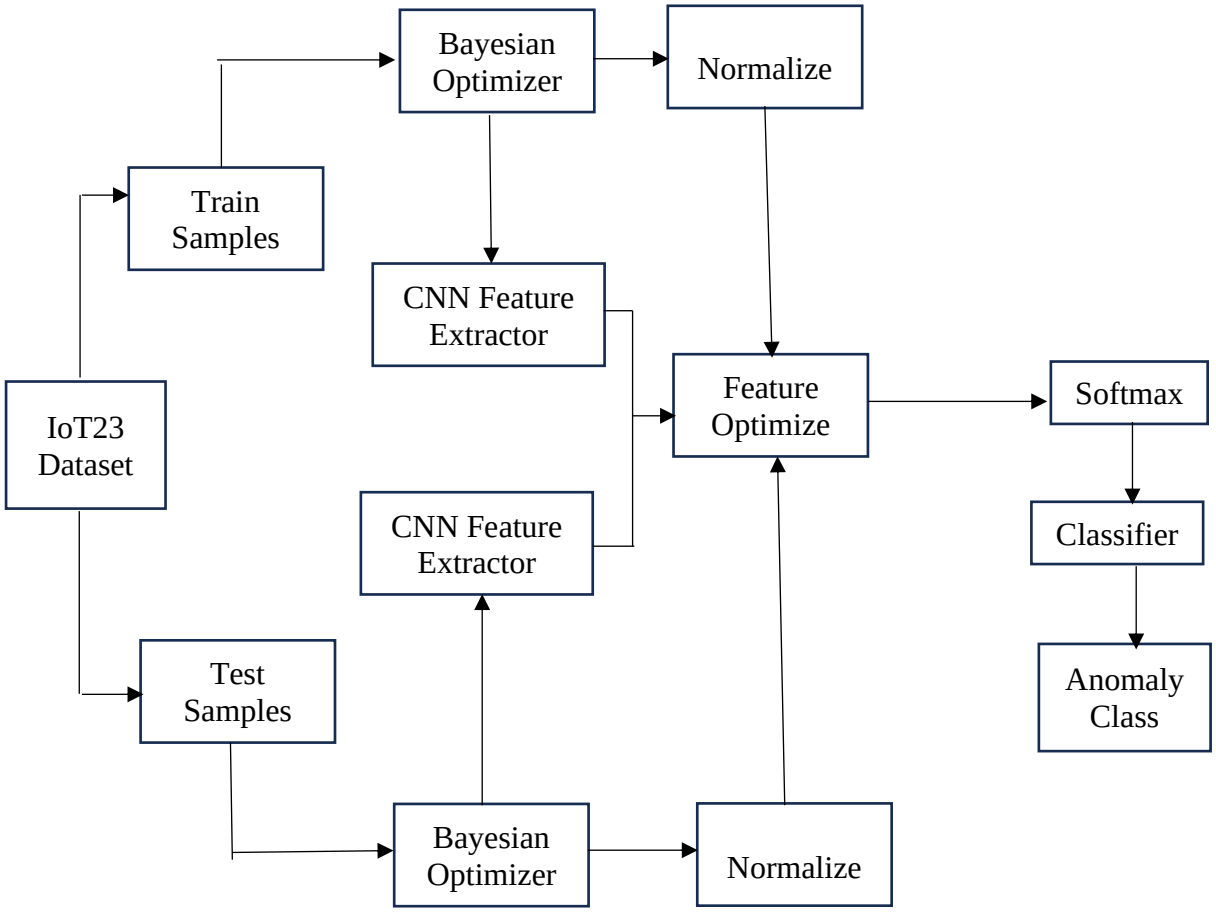

Fig. 2 Block diagram of proposed methodology

```
dataset.head(10)
```

	duration	orig_bytes	resp_bytes	missed_bytes	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	proto_icmp	proto_tcp	...	conn_state_RSTR	conn_state_RSTR
0	3.69	93	97	95	222.0	189.0	481	3.74	166.0	460	...	321.5	46
1	7.19	185	193	185	264.0	203.0	518	6.28	221.0	485	...	191.0	38
2	7.96	174	185	190	300.0	254.0	550	7.93	260.0	524	...	282.5	43
3	8.48	188	194	184	314.0	246.0	571	8.28	289.0	536	...	253.5	40
4	8.88	94	92	83	288.0	247.0	559	8.17	275.5	524	...	238.5	41
5	9.04	95	95	91	291.0	258.0	562	8.52	296.0	557	...	277.0	41
6	8.73	195	197	198	305.0	233.0	591	8.65	273.0	554	...	277.0	41
7	7.98	94	94	91	248.0	228.0	543	7.85	238.5	528	...	259.0	40
8	9.13	77	90	95	276.0	271.0	568	8.96	310.5	575	...	267.0	40
9	7.67	191	193	193	243.0	233.0	544	7.78	228.0	514	...	298.0	43

10 rows × 25 columns

Fig. 3 Visualization of dataset

```
dataset.describe()
```

	duration	orig_bytes	resp_bytes	missed_bytes	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	proto_icmp	proto_tcp	...
count	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	19086.000000	...
mean	7.393091	129.457299	130.580635	124.700618	180.568768	163.052054	466.559887	6.645782	197.055402	490.673373	...
std	18.692441	47.099761	49.579959	51.896650	80.161880	74.326883	85.078811	2.094584	58.345334	97.103065	...
min	0.000000	47.000000	45.000000	33.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	6.380000	83.000000	82.000000	77.000000	111.500000	101.125000	429.000000	5.770000	161.500000	451.000000	...
50%	7.500000	140.000000	141.000000	125.000000	169.500000	153.000000	479.000000	7.330000	200.500000	505.000000	...
75%	8.220000	176.000000	181.000000	178.000000	252.000000	225.000000	520.000000	8.070000	238.000000	551.000000	...
max	748.000000	200.000000	200.000000	200.000000	376.000000	355.500000	609.000000	10.000000	331.000000	652.000000	...

8 rows × 24 columns

Fig. 4 Summary statistics

3.2. Dataset Preprocessing and Exploratory Data Analysis

Data preprocessing is a crucial step in preparing datasets for analysis and modeling. Null value checking involves identifying and handling missing data within the dataset. This process typically includes assessing the presence of null or NaN values in each column and deciding on appropriate strategies for imputation or removal. Duplicate value checking aims to identify and eliminate redundant entries within the dataset, ensuring data integrity. Duplicates can skew analysis results and lead to biased models, so their detection and removal are essential. By systematically addressing missing values and duplicates, data preprocessing enhances the quality of the dataset, laying a solid foundation for accurate and reliable analysis or model training.

Exploratory Data Analysis (EDA) [31] encompasses various methods and techniques to gain insights into the underlying patterns and structure of a dataset. Some common EDA methods, including summary statistics, data visualization, feature distribution, and heat maps, are used here.

Summary statistics are essential measures that provide a concise overview of the central tendency and dispersion of numerical data within a dataset, which is demonstrated in Figure 4. Figure 5 showcases the visualization of the data, presenting a graphical representation that aids in comprehending the patterns, relationships, or trends within the dataset. Table 1 shows the class labels and their corresponding counts.

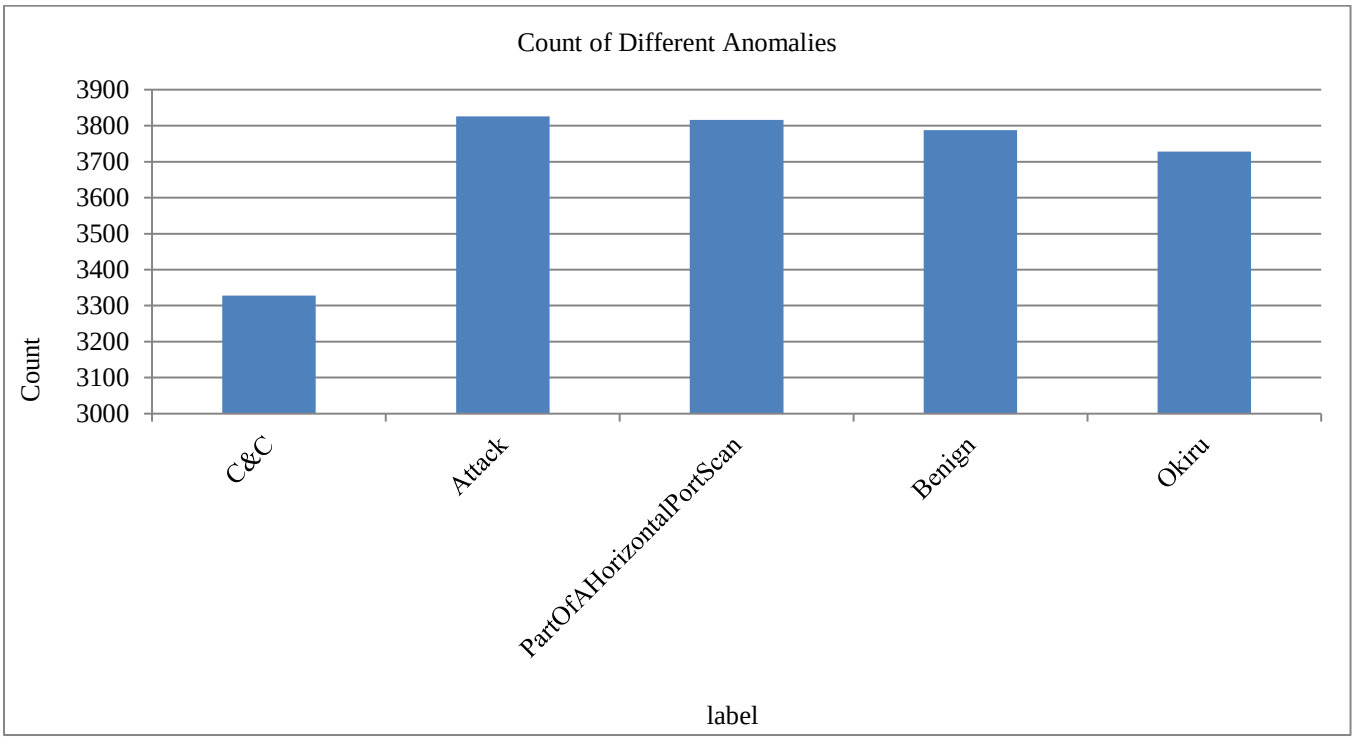

Fig. 5 Dataset visualization

Table 1. Count of class labels

Class Label	Count
C&C	3328
Attack	3826
Part of A Horizontal Port Scan	3816
Benign	3788
Okiru	3728

Histograms are a fundamental and effective data visualization technique used in Exploratory Data Analysis (EDA) to illustrate the distribution of numerical data. This graphical representation divides the data into bins or intervals and displays the frequency or count of observations within each bin through vertical bars. Each bar's height represents the count of data points falling within the designated range. Figure 6 illustrates the distribution of features within the dataset, providing a visual representation of how the various elements are spread or concentrated.

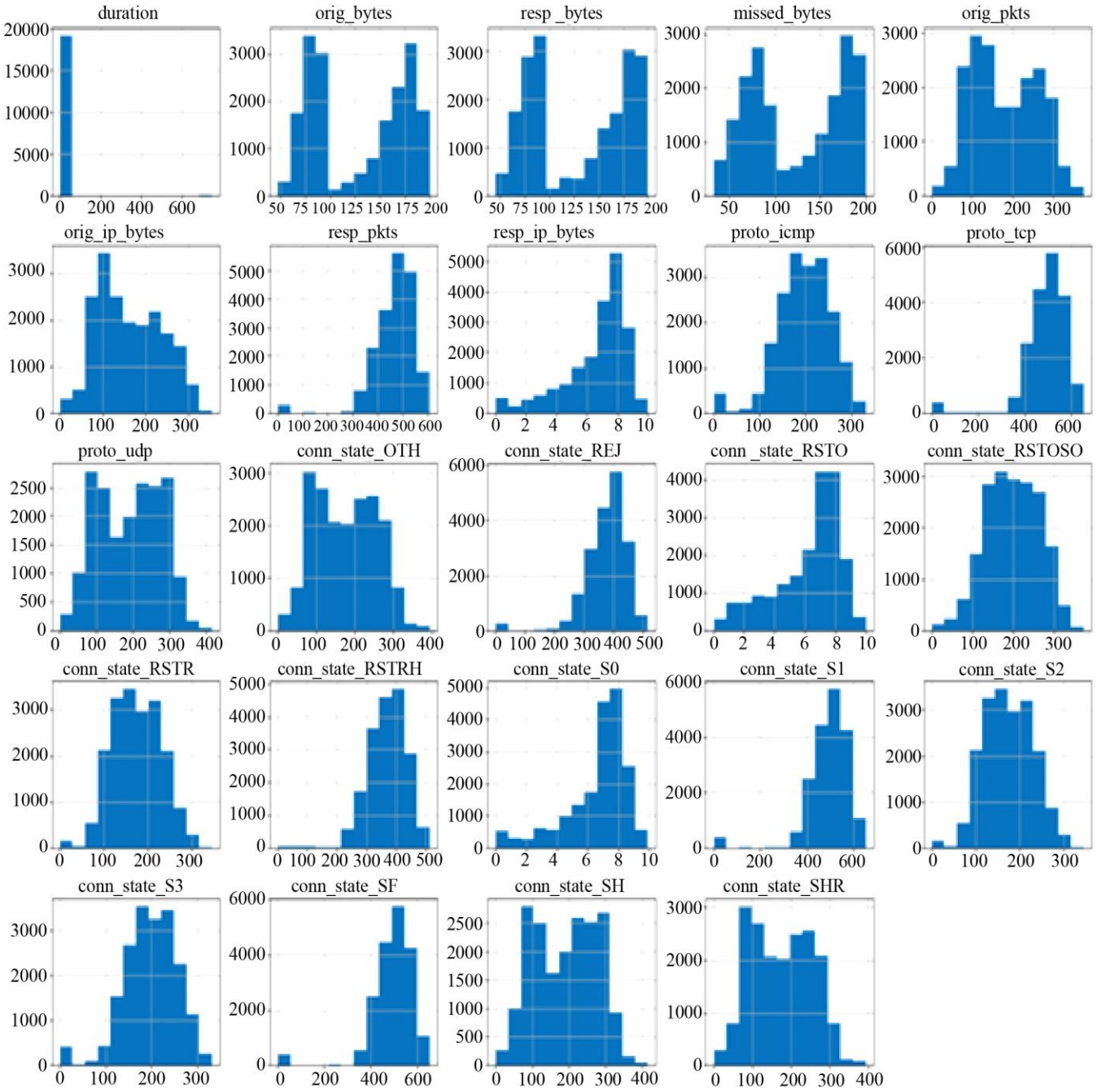

Fig. 6 Feature distribution

3.3. Bayesian Optimization

Bayesian optimization plays a pivotal role in refining the features used in CNN for anomaly classification in IoT security. Specifically, the study employs the NBOA to optimize the features fed into the CNN. Bayesian optimization is a powerful iterative technique that leverages probabilistic models to systematically explore and exploit the parameter space, seeking optimal configurations.

In this study, NBOA ensures that the CNN is trained on features that are finely tuned and adapted to the unique characteristics of the IoT23 dataset. This strategic optimization process enhances the model's overall performance, contributing to the accurate and efficient classification of anomalies in IoT security.

Figure 7 presents a flowchart illustrating the process of Bayesian optimization within the study. This visual representation delineates the sequential steps involved in leveraging Bayesian optimization, showcasing how the algorithm iteratively refines and explores the parameter space. Bayesian optimization leverages Gaussian processes [32], which are probabilistic or Bayesian models, to iteratively find the minimum of a function (f(x)) within a bounded set (X). Unlike traditional optimization methods that provide point estimates, Bayesian optimization considers the entire a-posteriori predictive distribution, allowing for the natural incorporation of uncertainty statistics. This method is particularly effective in scenarios where sampling the function is resource-intensive. It builds a posterior distribution of functions, continually refining as more observations are made.

Figure 8 exhibits the Gaussian process and utility function after nine steps in the study, offering a visual depiction of how these components evolve during the Bayesian optimization process. Matching a Gaussian process to known samples, enhancing the posterior distribution, and choosing the next exploration point using an exploration strategy are all steps in the optimization process. By minimizing the number of steps required to uncover nearly ideal parameter combinations, Bayesian optimization strikes a balance between exploration and exploitation. Variational Bayes approach is employed to normalize the probability ($\boldsymbol{P_r}$ (data)) in Bayes' Theorem. Optimization aims to find the best parameters and optimize the quantity $\boldsymbol{P_r(params|data)}$ with a focus on increasing this quantity during the exploration process. This iterative approach enhances algorithm certainty about regions in the parameter space worth exploring.

$$P_r(params|data) = \frac{P_r(data|params)P_r(params)}{P_r(data)} \quad (1)$$

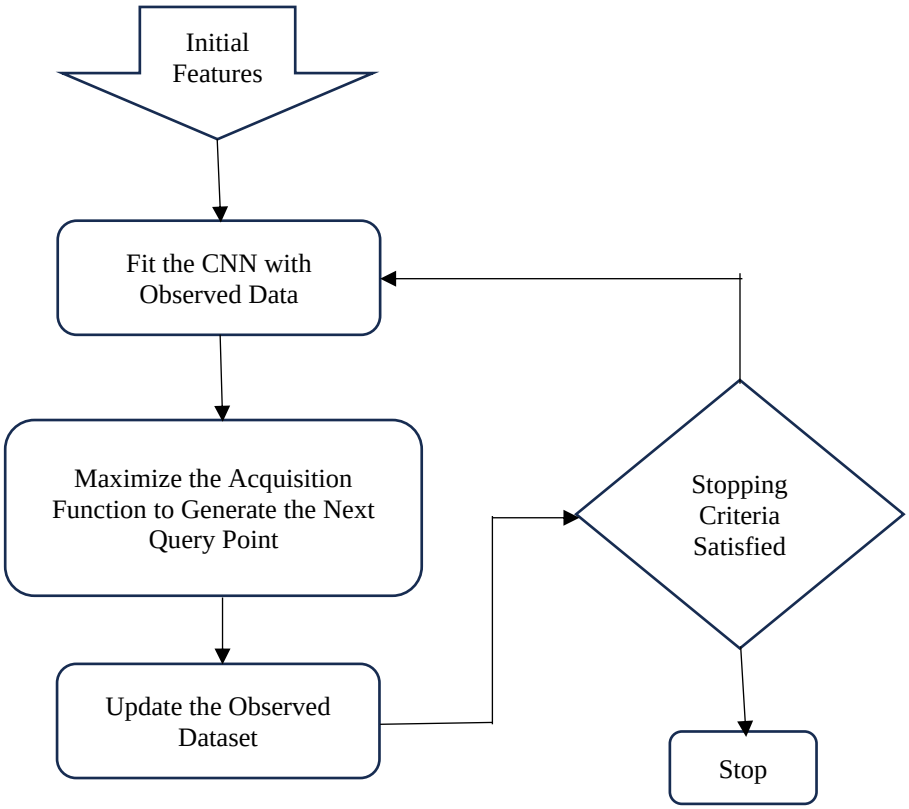

Fig. 7 Bayesian optimization

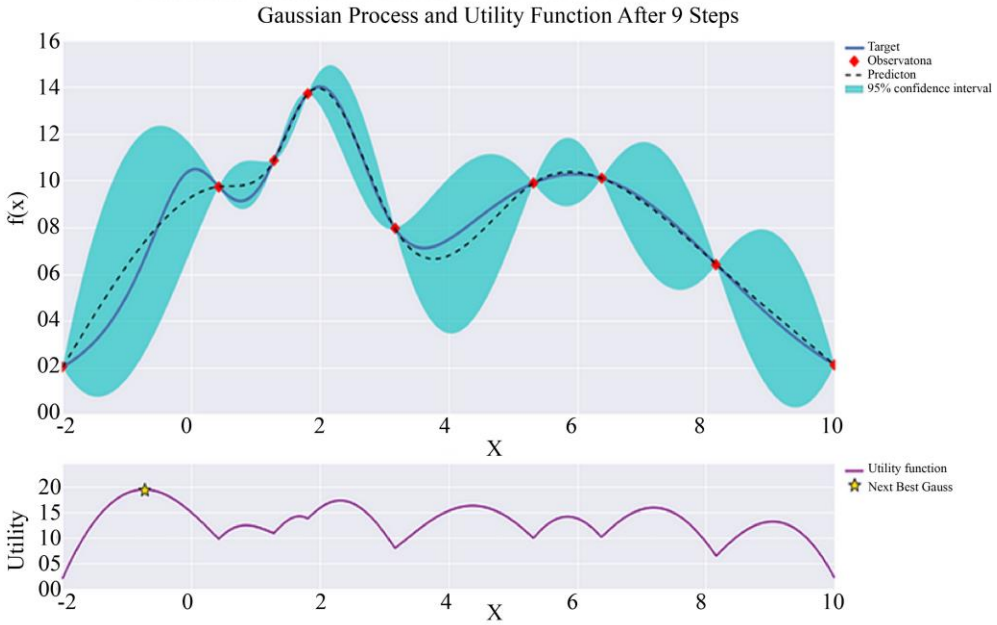

Fig. 8 Gaussian process and utility function after 9 steps

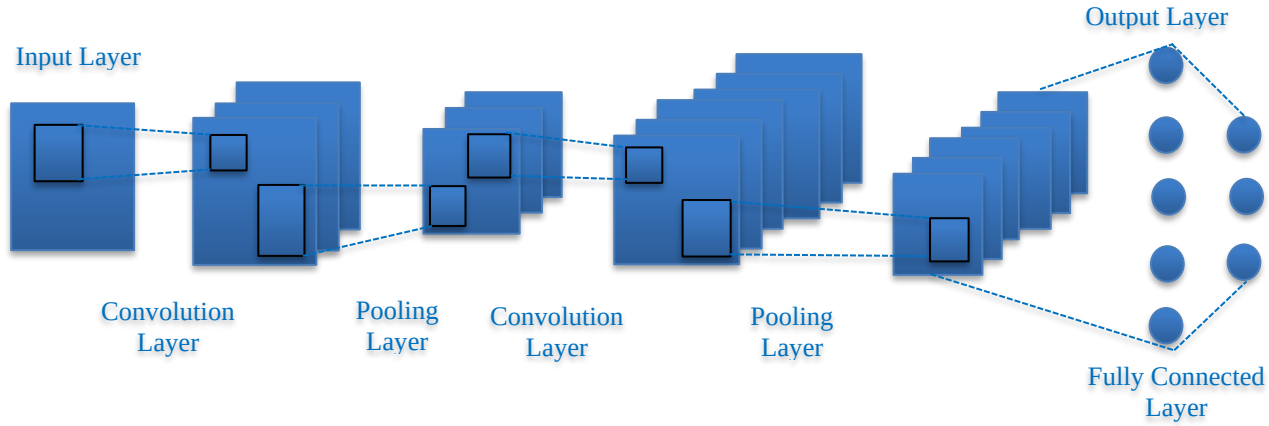

Fig. 9 Basic CNN architecture

3.4. Convolutional Neural Network

In tasks related to classification, groundbreaking performance has been demonstrated by Convolutional Neural Networks (CNNs). Their superiority over other ML approaches stems from their ability to handle transformations such as scaling, rotation, and Field of View (FOV), providing robust results. A standard design for CNN includes layers like input, convolutional, pooling, fully connected, and output layers.

The basic structure of a CNN model is illustrated in Figure 9. To effectively map an image's class labels, the input layer must initially receive the images. The Convolution Layer, serving as the core of a CNN-based DL model, employs a set of trainable convolution filters. The resulting two-dimensional maps are generated by taking the dot product of the input vector and the convolution filter.

The convolutional layer's operation on an input image X, where X has dimensions ijc and c represents the number of channels, is defined by the following Equation (2):

$$f(X) = \sum_{k=0}^{c} w_k * X_k + b \quad (2)$$

Variables in this context include weight (w), bias (b), nodes (k), and the convolution operator (*). A feature map is generated by applying convolution to an image using a weight vector within a convolution layer. In the training process, backpropagation is employed to adjust and optimize the weight vector. This layer is commonly integrated into CNNs because of the non-linearity introduced by the Rectified Linear Unit (RELU), ensuring a consistent gradient for all positive input values and addressing non-differentiability. Equation (3) represents the ReLU function when these values are substituted.

$$ReLU(X) = \begin{cases} 0, & if\ X < 0 \\ X, & otherwise \end{cases} \quad (3)$$

To enhance training efficiency beyond what can be achieved with standard sigmoid or hyperbolic tangent functions, the ReLU function is employed in CNNs. Unlike traditional neural networks facing the vanishing gradient problem, integrating the ReLU activation function in CNN architectures can mitigate this issue.

The pooling layer, situated between convolution layers, serves to down-sample the output from convolution layers. Utilizing two types of pooling functions, namely average pooling and max pooling, CNN reduces dimensionality without sacrificing meaningful information. Average pooling computes the average value for non-overlapping sub-partitions of the convolved image, while max pooling achieves the same objective by returning the maximum value for each partition. The pooling layer contributes to intermediate dimensionality reduction between two convolution layers.

The final layer in a typical CNN design is the fully connected layer, positioned just before the output layer. Each node in this layer connects to every other node, resembling a classic neural network with adjustable parameters. To reduce computational complexity, the dropout approach is applied in this layer, removing certain nodes and connections. As CNN-based classification methods extract more intricate information, system complexity increases.

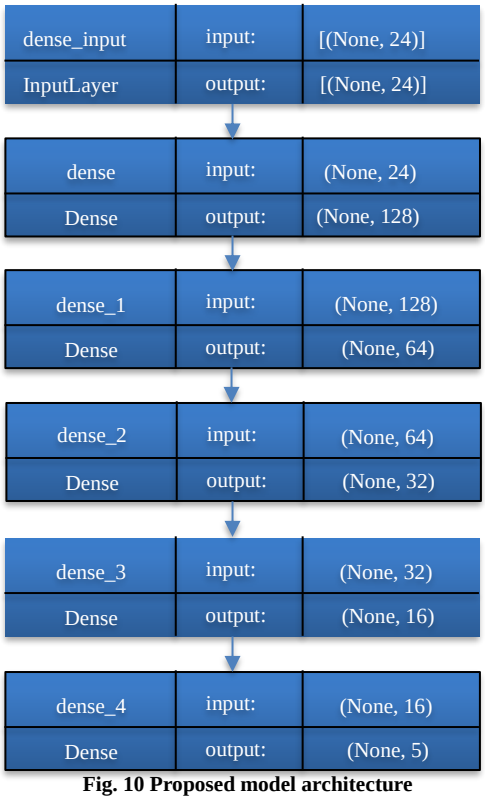

Fig. 10 Proposed model architecture

The proposed neural network architecture is a sequential model designed for multi-class classification tasks using the Keras library built on top of TensorFlow. Figure 10 represents the model architecture of the proposed model in the study, providing a visual depiction of the architecture or conceptual framework designed to address the objectives outlined in the research. The model follows a feedforward structure consisting of a linear stack of five Dense layers. The initial layer, with 128 neurons, accepts an input shape of 24 features and applies the Rectified Linear Unit (ReLU) activation function, introducing non-linearity to capture complex relationships in the data. Subsequent layers, containing 64, 32, and 16 neurons correspondingly, also utilize the ReLU activation function. These hidden layers serve as feature extractors, progressively reducing the dimensionality of the input.

The final layer, comprising 5 neurons and employing the softmax activation function, is tailored for multi-class classification. It outputs probability distributions across the five classes, facilitating the assignment of class labels to input data. The model is compiled using categorical crossentropy as the loss function, a standard choice for multiclass classification problems, and the Adam optimizer for efficient gradient-based optimization. The evaluation metric chosen is accuracy, providing insights into the model's performance during training. The training process involves iteratively adjusting the model's parameters over 500 epochs using a batch size of 500 instances. Finally, the model, after being trained, is employed to predict outcomes on the test data, generating class probabilities for each input instance.

3.5. Performance Evaluation

Table 2 offers a thorough summary of the performance metrics and their corresponding equations in the study.

Table 2. Performance metrics with equation

Performance Metrics	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-Score	$2 \times \frac{(Precision \times Recall)}{(Precision+Recall)}$
TP- True Positive, TN- True Negative, FP- False Positive, FN- False Negative	

Table 3. Hyperparameters

Parameters	values
Number of Epochs	4000
Batch Size	500
Iterations	500
Loss function	Categorical_crossentropy
Optimizer	Adam

4. Results and Discussion

4.1. Hardware and Software Setup

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 3 furnishes a detailed account of the hyperparameters used in the study, presenting a comprehensive list and their respective values.

In the specified model configuration, hyperparameters have a vital role in establishing learning dynamics and optimization strategies during the training process. "Categorical_crossentropy" loss function is chosen, indicating that the model is likely performing a multi-class classification task, while the "adam" optimizer is selected for its efficiency in adapting learning rates during training. The hyperparameter choices include training for a substantial number of epochs. These hyperparameters collectively influence the model's ability to converge to an optimal solution and can significantly impact the training efficiency and final performance of the DL model.

4.2. Experimental Results

Figure 11 showcases the accuracy of training and validation, offering insights into the model's performance in terms of training data, and Figure 12 visualizes the training and validation loss, offering a perspective on the model's convergence or divergence during the training period. The x-axis corresponds to the number of epochs, providing a temporal view of the model's learning process.

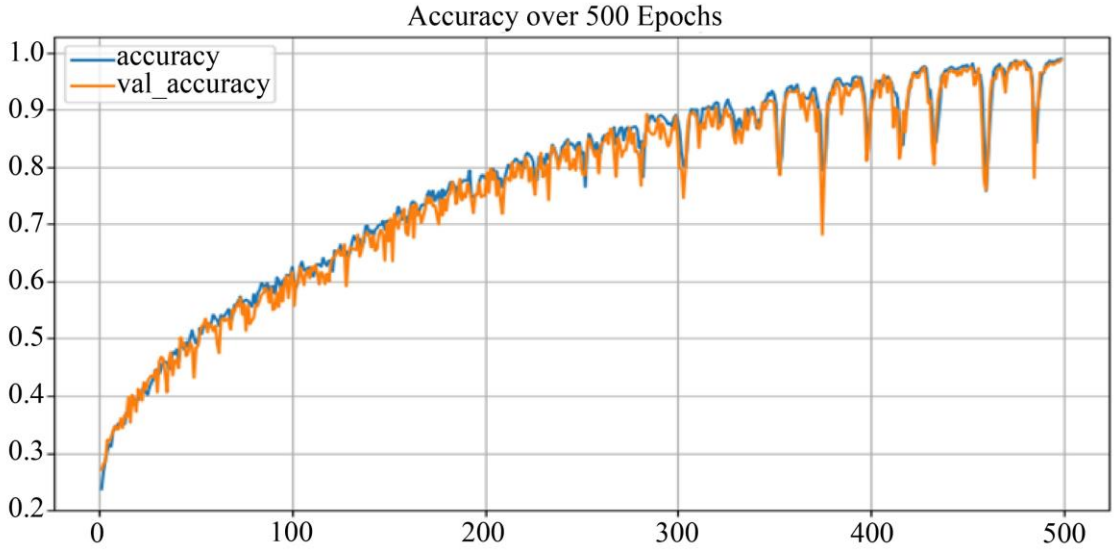

Fig. 11 Accuracy plot

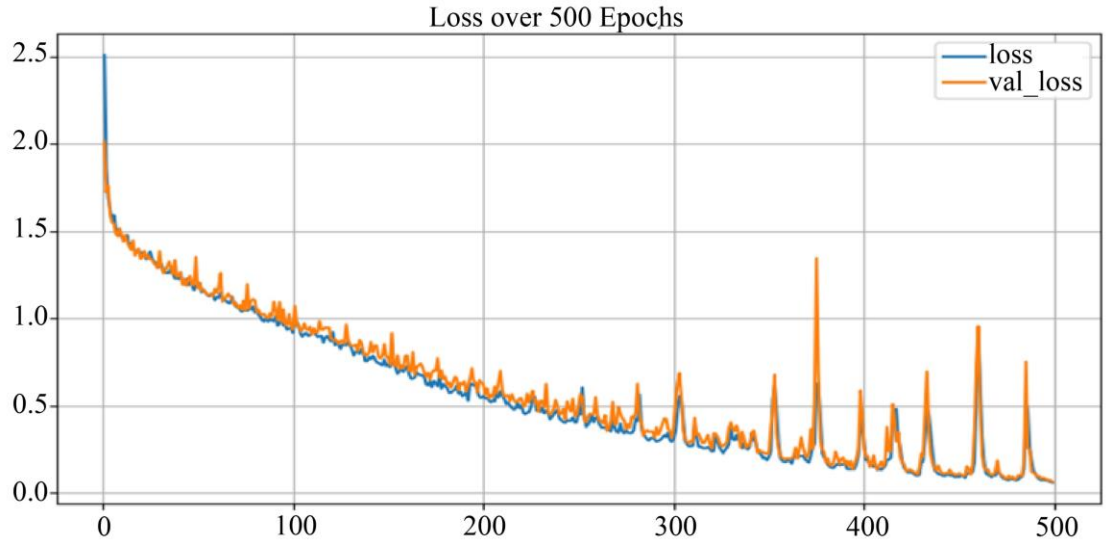

Fig. 12 Loss plot

The accuracy plot demonstrates the model's ability to correctly classify instances, reaching a peak accuracy of approximately 98.69% by the end of the training period. This upward trend indicates that the model continually refines its predictive capabilities, capturing intricate patterns in the training data. On the other hand, the loss plot portrays the model's convergence towards minimizing the training loss. As epochs progress, the model systematically reduces its training loss, indicative of improved optimization and the successful adjustment of internal parameters. The convergence of the loss curve suggests that the model is effectively learning from the training data, achieving an equilibrium between preventing overfitting and underfitting. Together, these plots demonstrate the model's outstanding performance, demonstrating high accuracy and minimal loss, demonstrating its capacity to accurately categorize abnormalities in the studied data.

Table 4. Classification report of model

Class	Precision	Recall	F1-Score
C&C	0.98	0.97	0.98
Attacks	0.97	0.98	0.98
PartOfAHorizontalPortScan	1.00	1.00	1.00
Benign	0.99	0.99	0.99
Okiru	0.99	0.99	0.99

The classification report offers a thorough assessment of a model's effectiveness in several classes. A thorough analysis of the model's classification report, including precision, recall, and F1-score for every class, is provided in Table 4.

The F1-score is the harmonic mean of precision and recall. Precision indicates the accuracy of positive predictions, while memory gauges the capacity to record all positive examples.

Interestingly, the class "PartofaHorizontalPortScan" demonstrated immaculate model performance with perfect precision, recall, and F1-score. Other classes with precision and recall scores between 0.98 and 0.99, like "C&C," "Attack," "Benign," and "Okiru," also show good performance. These findings imply that the model has a good capacity for correctly classifying cases in a range of classes.

Each class's total number of instances is displayed in the "support" column, which aids in understanding the distribution of data among the categorization groups. In summary, this classification report provides a thorough analysis of the model's efficacy, emphasizing its strong ability to differentiate between several classes with high recall, precision, and F1-score values.

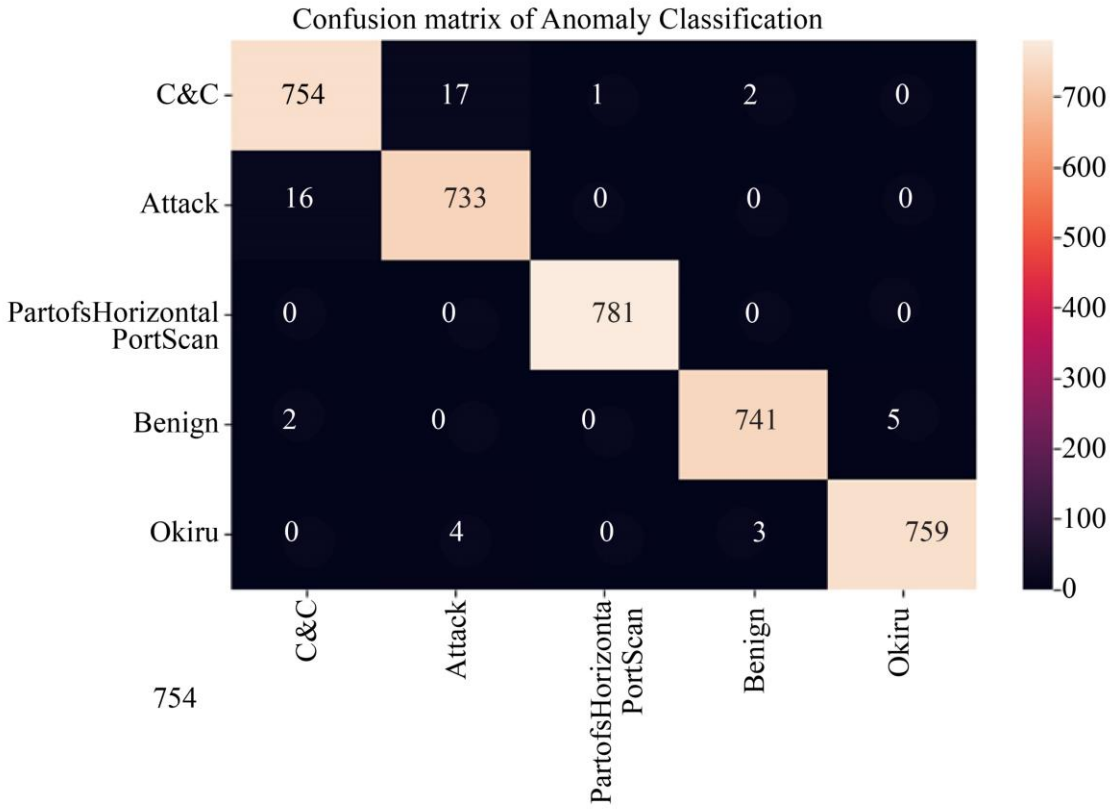

Fig. 13 Confusion matrix

Utilizing a confusion matrix, the experimental findings from the given data show a thorough assessment of a classification model's performance. Figure 13 displays a confusion matrix, which is essential for assessing the performance of a classification model's efficacy in anomaly classification. A tabular representation of the model's predictions compared to the actual outcomes for each class is provided by the confusion matrix. Assessing the model's overall effectiveness, memory, precision, and accuracy in categorization is especially beneficial. The matrix provides information on true positives, true negatives, false positives, and false negatives by assigning a unique combination of expected and actual class labels to each cell. Table 5 provides a comprehensive display of the performance metrics associated with the proposed approach in the study.

Table 5. Performance evaluation of the proposed model

Performance matrix	Values (%)
Accuracy Score	98.69
Misclassification Score	1.30
Precision Score	98
Recall Score	98
F1 Score	98

4.3. Performance Comparison

The performance of the proposed anomaly detection system is assessed against existing models, evaluating computational efficiency. Table 6 provides a comparison of the proposed method with the existing models.

Table 6. Performance comparison with existing methods

Author Details	Title	Model	Accuracy	Precision	Recall	F1-Score
Vishwakarma et al. [14]	A new two-phase IDS with Naïve Bayes ML for data classification and elliptic envelop method for anomaly detection	Elliptic envelop method for anomaly detection	98.95 %	95.40%	97.51 %	96.44 %
Awajan et al. [17]	A Novel DL-Based IDS for IoT Networks	DL- Based IDS System	93.74 %	93.71%	93.82 %	93.47 %
Elsayed et al. [25]	Securing IoT and SDN Systems Using Deep Learning Based Automatic IDS	Deep Learning	96. 56 %	97.3 %	97.35 %	97.4 %
Khan et al. [33]	Voting Classifier Based IDS for IoT Networks	DT- RFKNN- NB	87 %	90 %	88 %	87 %
Proposed Methodology	Deep Learning Model with Normalized Bayesian Optimizer for Anomaly Classification in IoT Security	Deep Learning and Normalized Bayesian Optimization	98.69 %	98 %	98 %	98 %

Comparing the performance metrics of various models, the proposed methodology in this study achieves the highest accuracy at 98.69%, with recall, F1-score and precision at 98%. It outperforms other models such as the two-phase IDS with Naïve Bayes, elliptic envelop method for anomaly detection (Accuracy: 98.95%, Precision: 95.40%, Recall: 97.51%, F1-score: 96.44%), the DL-based IDS for IoT networks (Accuracy: 93.74%, Precision: 93.71%, Recall: 93.82%, F1-score: 93.47%), the deep-learning based automatic IDS (Accuracy: 96.56%, Precision: 97.3%, Recall: 97.35%, F1-score: 97.4%), and the voting classifier-based IDS for IoT networks with DT-RFkNN-NB (Accuracy: 87%, Precision: 90%, Recall: 88%, F1-score: 87%). The proposed model demonstrates superior overall performance in anomaly classification for IoT security.

5. Conclusion

The study delves into the critical domain of IoT security, recognizing the heightened vulnerabilities posed by the widespread adoption of IoT devices. The proposed approach, integrating a CNN with feature optimization through the

NBOA, addresses the challenges of anomaly classification in IoT security. Leveraging the IoT23 dataset, the study showcases the efficacy of the CNN model trained on optimized features, achieving a remarkable accuracy of 98.69%. This surpasses the performance of traditional CNN models and underscores the significance of feature optimization in enhancing anomaly detection accuracy. The methodology's success in fine-tuning feature selection through NBOA establishes a promising avenue for bolstering IoT security against threats and potential breaches. As the IoT landscape continues to expand, the proposed approach provides a valuable contribution to the ongoing efforts to fortify the security infrastructure of IoT systems, ensuring the reliability and integrity of connected devices and networks. Further research can explore additional optimization techniques and scalability considerations to advance the resilience of anomaly detection in the ever-changing landscape of IoT security.

Acknowledgments

I would like to express my sincere gratitude to all those who contributed to the completion of this research paper. I extend my heartfelt thanks to my supervisor, my family, my colleagues and fellow researchers for their encouragement and understanding during the demanding phases of this work.

References

[1] Antar Shaddad Abdul-Qawyet al., "The Internet of Things (IoT): An Overview," *International Journal of Engineering Research and Applications*, vol. 5, no. 12, pp. 71-82, 2015. [Google Scholar] [Publisher Link]

[2] Joseph Louis, and Phillip S. Dunston, "Integrating IoT Into Operational Workflows for Real-Time and Automated Decision-Making in Repetitive Construction Operations," *Automation in Construction*, vol. 94, pp. 317-327, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Vignesh Govindraj, Mithileysh Sathiyanarayanan, and Babangida Abubakar, "Customary Homes to Smart Homes Using Internet of Things (IoT) and Mobile Application," *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Bengaluru, India, pp. 1059-1063, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sharath Anand, and Sudhir K. Routray, "Issues and Challenges in Healthcare Narrowband IoT," *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, pp. 486-489, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ashwini Deshpande, Prajakta Pitale, and Sangita Sanap, "Industrial Automation Using Internet of Things (IOT)," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 5, no. 2, pp. 266-269, 2016. [Google Scholar]

[6] Viraaji Mothukuri et al., "Federated-Learning-based Anomaly Detection for IoT Security Attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545-2554, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sundar Krishnan, Ashar Neyaz, and Qingzhong Liu, "IoT Network Attack Detection using Supervised Machine Learning," *International Journal of Artificial Intelligence and Expert Systems*, vol. 10, no. 2, pp. 18-32, 2021. [Google Scholar] [Publisher Link]

[8] Giampaolo Casolla et al., "Exploring Unsupervised Learning Techniques for the Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2621-2628, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Shailendra Rathore, and Jong Hyuk Park, "Semi-Supervised Learning Based Distributed Attack Detection Framework for IoT," *Applied Soft Computing*, vol. 72, pp. 79-89, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Francesco Cauteruccio et al., "A Framework for Anomaly Detection and Classification in Multiple IoT Scenarios," *Future Generation Computer Systems*, vol. 114, pp. 322-335, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Tom Lawrence, and Li Zhang, "IoTNet: An Efficient and Accurate Convolutional Neural Network for IoT Devices," *Sensors*, vol. 19, no. 24, pp. 1-27, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Krishna Chaitanya Atmakuri, and Y. Venkata Raghava Rao, "An IOT Based Novel Approach to Predict Air Quality Index (AQI) Using Optimized Bayesian Networks," *Journal of Mechanics of Continua and Mathematical Sciences*, vol. 14, no. 2, pp. 482-497, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sanjay Chakraborty et al., "Detection and Classification of Novel Attacks and Anomaly in IoT Network Using Rule Based Deep Learning Model," *arXiv*, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Monika Vishwakarma, and Nishtha Kesswani, "A New Two-Phase Intrusion Detection System with Naïve Bayes Machine Learning for Data Classification and Elliptic Envelop Method for Anomaly Detection," *Decision Analytics Journal*, vol. 7, pp. 1-8, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Harshit Shah et al., "Deep Learning-Based Malicious Smart Contract and Intrusion Detection System for IoT Environment," *Mathematics*, vol. 11, no. 2, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Nouman Shamim et al., "Efficient Approach for Anomaly Detection in IoT Using System Calls," *Sensors*, vol. 23, no. 2, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Albara Awajan, "A Novel Deep Learning-Based Intrusion Detection System for IoT Networks," *Computers*, vol. 12, no. 2, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mansi Bhavsar et al., "Anomaly-Based Intrusion Detection System for IoT Application," *Discover Internet of Things*, vol. 3, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Laura Vigoya et al., "Application of Machine Learning Algorithms for the Validation of a New CoAP-IoT Anomaly Detection Dataset," *Applied Sciences*, vol. 13, no. 7, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Indrajit Mukherjee, Nilesh Kumar Sahu, and Sudip Kumar Sahana, "Simulation and Modeling for Anomaly Detection in IoT Network Using Machine Learning," *International Journal of Wireless Information Networks*, vol. 30, no. 2, pp. 173-189, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ammar Odeh, and Anas Abu Taleb, "Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection," *Applied Sciences*, vol. 13, no. 21, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Abbas Yazdinejad et al., "An Ensemble Deep Learning Model for Cyber Threat Hunting in Industrial Internet of Things," *Digital Communications and Networks*, vol. 9, no. 1, pp. 101-110, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Chaimae Hazman et al., "lIDS-SIoEL: Intrusion Detection Framework for IoT-Based Smart Environments Security Using Ensemble Learning," *Cluster Computing*, vol. 26, no. 6, pp. 4069-4083, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Badar Alwasel et al., "Leveraging Graph-Based Representations to Enhance Machine Learning Performance in IIoT Network Security and Attack Detection," *Applied Sciences*, vol. 13, no. 13, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Rania A. Elsayed et al., "Securing IoT and SDN Systems Using Deep-Learning Based Automatic Intrusion Detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Tanzila Saba et al., "Anomaly-Based Intrusion Detection System for IoT Networks through Deep Learning Model," *Computers and Electrical Engineering*, vol. 99, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Kim-Hung Le et al., "IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT," *Electronics*, vol. 11, no. 4, pp. 1-16, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Unsub Shafiq et al., "Transfer Learning Auto-Encoder Neural Networks for Anomaly Detection of DDoS Generating IoT Devices," *Security and Communication Networks*, vol. 2022, no. 1, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ola Salman et al., "A Machine Learning Based Framework for IoT Device Identification and Abnormal Traffic Detection," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Imtiaz Ullah, and Qusay H. Mahmoud, "Design and Development of RNN Anomaly Detection Model for IoT Networks," *IEEE Access*, vol. 10, pp. 62722-62750, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Kabita Sahoo et al., "Exploratory Data Analysis Using Python," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 4727-4735, 2019. [Google Scholar] [Publisher Link]

[32] Andreas Damianou, and Neil D. Lawrence, "Deep Gaussian Processes," *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pp. 207-215, 2013. [Google Scholar] [Publisher Link]

[33] Muhammad Almas Khan et al., "Voting Classifier-Based Intrusion Detection for IoT Networks," *Advances on Smart and Soft Computing*, pp. 313-328, 2022. [CrossRef] [Google Scholar] [Publisher Link]