electronics

MDPI

Article

Cost-Efficient Hybrid Filter-Based Parameter Selection Scheme for Intrusion Detection System in IoT

Gabriel Chukwunonso Amaizu [1,2], Akshita Maradapu Vera Venkata Sai [1,*], Madhuri Siddula [3] and Dong-Seong Kim [2]

[1] Digital Twin Research Group, Towson University, Towson, MD 21252, USA; gamaizu@towson.edu
[2] IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea; dskim@kumoh.ac.kr
[3] Department of Computer Science, North Carolina Agricultural and Technical University, Greensboro, NC 27411, USA; msiddula@ncat.edu
* Correspondence: amaradapuveravenkatasai@towson.edu

Abstract: The rapid growth of Internet of Things (IoT) devices has brought about significant advancements in automation, data collection, and connectivity across various domains. However, this increased interconnectedness also poses substantial security challenges, making IoT networks attractive targets for malicious actors. Intrusion detection systems (IDSs) play a vital role in protecting IoT environments from cyber threats, necessitating the development of sophisticated and effective NIDS solutions. This paper proposes an IDS that addresses the curse of dimensionality by eliminating redundant and highly correlated features, followed by a wrapper-based feature ranking to determine their importance. Additionally, the IDS incorporates cutting-edge image processing techniques to reconstruct data into images, which are further enhanced through a filtering process. Finally, a meta classifier, consisting of three base models, is employed for efficient and accurate intrusion detection. Simulation results using industry-standard datasets demonstrate that the hybrid parameter selection approach significantly reduces computational costs while maintaining reliability. Furthermore, the combination of image transformation and ensemble learning techniques achieves higher detection accuracy, further enhancing the effectiveness of the proposed IDS.

Keywords: computational efficiency; IIoT; image processing; machine learning; network intrusion detection; parameter selection

check for updates

Academic Editor: Djuradj Budimir

Received: 31 December 2024
Revised: 31 January 2025
Accepted: 10 February 2025
Published: 13 February 2025

Citation: Amaizu, G.C.; Maradapu Vera Venkata Sai, A.; Siddula, M.; Kim, D.-S. Cost-Efficient Hybrid Filter-Based Parameter Selection Scheme for Intrusion Detection System in IoT. *Electronics* **2025**, *14*, 726. https://doi.org/10.3390/electronics14040726

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) is continuously evolving, playing a vital role in our daily life, while its application spreads to areas such as smart vehicles [1], smart homes [2], smart health [3], smart factories [4], and logistics [5]. Cisco predicted that 66% of the world population will be using the Internet by 2023, a 15% increase from 2018. Also, there will be 3.6 connected devices per person, whereas it was just 2.4 connected devices per person in 2018 [6].

The pervasiveness and subsequent acceptability of IoT in various industries have made IoT devices and infrastructures a conspicuous cyberattack target by malicious actors [7,8]. These actors continually exploit vulnerabilities in network infrastructures to gain unauthorized access, exfiltrate sensitive information, disrupt services, and cause significant financial and reputational damage. Consequently, the need for robust and reliable security

mechanisms to counter these threats has never been more critical. To combat the ever-present cyberattacks, the concept of intrusion detection systems (IDSs) was coined. IDSs serve as an integral defence mechanism against such attacks. Over the years, IDSs have been broadly classified into three groups: misuse- or signature-based, anomaly-based, and hybrid IDSs. Misuse-based approaches tend to check network traffic for a known pattern of intrusion. As the system checks incoming traffic signatures, it can only detect known attacks. Furthermore, it is incredibly challenging to write signatures for all possible variations of attack. On the other hand, anomaly-based IDSs are capable of detecting both known and unknown attacks. These algorithms learn the patterns of non-intrusive traffic. Hence, they are capable of identifying an intrusive or unusual traffic pattern when they occur [9]. Hence, the research community has adopted anomaly-based IDSs [10] by virtue of them being able to detect both known and unknown attacks. Machine learning and deep learning models have been used to varying degrees of success for anomaly detection [11]. Meanwhile, ML models have suffered from low detection accuracy, and DL models have been prone to overfitting [12]. These models are trained to learn the patterns of normal traffic and of anomalous traffic; then, they are deployed to classify traffic in real time.

Computer vision and image processing have since been gaining traction and have widely been used to great effects in the healthcare sector, especially for the detection of lung cancer. However, transferring that success to the detection of network anomalies is still at an infant stage. One of the drawbacks to this is the apparent need to transform network data (which, by their nature, are non-image data) into image data. Converting non-image data to images has emerged as a promising technique in data analysis. It offers several advantages, including the utilization of powerful deep learning models for image recognition and pattern detection. Deep learning algorithms, such as convolutional neural networks (CNNs), have demonstrated remarkable performance in image-related tasks [13]. By transforming non-image data into image representations, researchers can leverage the vast array of CNN architectures and image processing techniques, enabling them to extract meaningful patterns and insights from complex datasets [14]. Furthermore, transforming network data into images allows the proposed IDSs to utilize advanced image processing and deep learning techniques, such as CNNs. This transformation not only enhances feature representation by capturing intricate spatial and frequency patterns but also enables more effective classification of complex attack types. By leveraging these techniques, the IDS achieves improved detection accuracy and robustness compared to traditional methods. Moreover, accurate IDSs require careful parameter selection to retain relevant and discriminative variables while discarding redundant ones [15]. Existing approaches, such as filter-based and wrapper-based methods [16], have limitations: wrapper methods are computationally expensive, while filter methods can be unstable [17]. Hybrid filter-based selection methods offer a reliable alternative by combining multiple statistical approaches to assess input variable characteristics and inter-relationships. This ensures discriminative parameter selection, minimizing computational costs and addressing the curse of dimensionality.

This work presents a four-phase architecture for detecting and classifying cyberattacks in IoTs. The first phase consists of three steps—basic data preprocessing, outlier removal using the interquartile range (IQR), and data normalization using three algorithms—with the best-performing one automatically selected. To eliminate indiscriminate and redundant parameters, a hybrid filter-based parameter selection algorithm is proposed and implemented in the second phase. Parameter selection is a process of identifying optimal parameters using mathematical formulas or tools [15,18]. In the third phase, non-image network data are transformed into image data using the algorithm proposed in [19], in order to leverage the power of image processing and classification. The Gabor filter is

applied to capture relevant shape information from the resulting images. Finally, in the fourth phase, an ensemble CNN model is proposed for classification, consisting of a meta-classifier and three base CNN models with distinct architectures, hyperparameters, and feature representations. The ensemble model combines predictions from multiple base models to enhance the accuracy of IDSs.

The subsequent sections of this paper are organized as follows: Section 2 discusses related techniques used in IDSs. Section 3 provides detailed explanations of each of the four phases of the proposed scheme. Section 4 describes the experimental setup, presents the observed results, and provides discussions. Finally, this paper is concluded in Section 5.

2. Related Works and Background Study

The SHDA [20] scheme introduces a novel approach to mitigate slow HTTP DDoS attacks by leveraging SDN controllers instead of deploying applications on web servers. By performing DDoS checks on incoming incomplete HTTP requests and forwarding only the complete ones to the server, the scheme effectively filters out malicious traffic. Similarly, in [21], authors propose a cluster-based solution for DDoS mitigation in wireless sensor networks. Through random number assignment and periodic validation, sensor nodes are protected from unauthorized access, ensuring the integrity of the network. Additionally, the REATO framework [22] addresses various DDoS attacks with a focus on rapid detection, countermeasures, and recovery, demonstrating favourable response time and latency. Furthermore, a multi-level mitigation framework [23] aligns with the layers of the Industrial Internet of Things (IIoT) architecture, protecting IIoT nodes through SDN-based gateways, IIoT control units at the fog level, and big data analysis at the cloud computing level for enhanced defence against cyberattacks. In the fog-to-things intrusion detection architecture [24], a two-level classification approach is proposed. Ensemble learning is used for anomaly detection in the first level, while the second level focuses on attack classification. In another study [25], an IDS is implemented using ensemble learning and decision tree recursive feature elimination (RFE) on the NSL-KDD dataset. Evaluation of ensemble techniques in [26] shows that stacking outperforms AdaBoost and Bagging. Additionally, a cluster-based ensemble classifier in [27] combines clustering and boosting for improved IDS prediction. SEHIDS, a self-evolving host-based intrusion detection system designed for IoT networks was introduced in [28]. SEHIDS enhances security by utilizing adaptive deep learning models that evolve dynamically to counter new threats. It employs a hybrid detection approach, combining signature-based and anomaly-based methods to improve the detection accuracy while minimizing false positives. The system integrates federated learning, enabling distributed training across IoT nodes while preserving privacy.

An ML-based network intrusion detection optimization for cloud computing environments was proposed in [29]. This system combines support vector machine and extreme gradient boosting (XGBoost) with the crow search algorithm (CSA) for hyperparameter optimization. The model enhances classification accuracy using XGBoost-based feature selection and is evaluated on NSL-KDD and UNR-IDD datasets. Results demonstrate improved detection performance and efficiency compared to traditional approaches. A hybrid CNN-BiLSTM-based network intrusion detection system tailored for SDN was proposed in [30]. The model integrates a CNN for spatial feature extraction and BiLSTM for temporal feature learning, enhancing intrusion detection performance. This study utilizes various datasets, incorporating hybrid feature selection to improve classification accuracy. Their approach outperforms baseline models, such as CNN, AlexNet, and LeNet-5, achieving high accuracy and efficiency in detecting diverse attack types. However, a common limitation of these approaches is their dependency on traditional tabular data representations,

which may not fully capture the complex patterns in network traffic. Additionally, while ensemble techniques enhance classification, they often suffer from high computational costs, making them impractical for real-time IDS deployment in IoT environments.

Numerous studies have delved into the application of CNNs in IDS. However, only a limited subset of these endeavours have involved a comprehensive transformation of network data into image data. In a recent study in [31], an innovative IDS grounded in representation learning via fast Fourier transformation (FFT) was introduced. This unique approach entails the conversion of each traffic instance into an image format, effectively recasting the intrusion detection task as a classification problem. In a similar vein, the work in [32] proposed the utilization of network spectrogram images, generated through short-time FFT, for enhancing intrusion detection performance and mitigating false alarm rates. Building upon these concepts, a more recent contribution by authors [33] presented an IDS system that initially translates network traffic into image data. To bolster discriminative features, the system employs Gabor filters before harnessing CNNs for the classification task. Authors [34] proposes Lightweight-Fed-NIDS, a federated learning-based IDS that integrates structured model pruning for efficiency. Using ResNet-50, ResNet-101, and VGG-19, it extracts features from network flow images while reducing computational overhead. Evaluations on UNSW-NB15, USTC-TFC2016, and CIC-IDS-2017 show a 99% accuracy. Despite the advantages of image-based IDS models, they introduce computational complexity that may hinder deployment in low-resource environments. FFT-based transformations, while effective in feature enhancement, are computationally expensive and may not scale well for large datasets. Additionally, most existing image-based IDS methods focus primarily on detection accuracy without optimizing computational efficiency, making them impractical for real-time applications.

In the realm of IDSs, approaches that convert network traffic into image data typically fall into one of two categories. The initial method entails the conversion of one-dimensional traffic data into a multidimensional matrix, while the alternative method employs FTT to effectuate the image transformation process [35]. Despite the advantages offered by these techniques, they still exhibit notable limitations. The first approach can compromise the inherent correlations among different features [36], and the utilization of FFT has been observed to introduce complexity concerns, particularly when applied to large-scale datasets [37]. Furthermore, none of the aforementioned studies explicitly address the trade-offs between detection performance and computational cost, particularly in resource-constrained IoT environments. Optimizing feature selection remains an open challenge, as many methods either prioritize accuracy at the expense of efficiency or vice versa. This research introduces a novel hybrid approach that integrates feature selection and computational cost reduction, ensuring that network traffic transformation into image data are both effective and computationally viable. Notably, this study employs a cutting-edge technique for translating network traffic into image data. Following this initial transformation, the data undergo further refinement before constructing a classifier model using an ensemble CNN.

3. System Methodology

The proposed system framework consists of four phases: preprocessing, parameter selection, data transformation, and classification. Figure 1 provides a graphical representation of a complete IDS. In the preprocessing phase, the dataset undergoes cleaning to remove zeroes (columns that consist of only zero values) and null values, followed by outlier elimination using the IQR method. Data normalization is then performed using three different methods: min–max, standard scaler, and robust scaler. The method with the highest average percentile rank is automatically selected. This marks the transition to phase

two, which involves eliminating highly correlated features and ranking the remaining features based on their importance. This parameter selection phase yields the optimal parameters that contribute the most to the model. Moving on to the data transformation phase, the selected features are transformed from their tabular form into images to employ the power of image processing algorithms. The Gabor filter is applied to further enhance the transformed images. Finally, in the classification phase, an ensemble CNN, consisting of three base models, is utilized to accurately detect and classify network traffic into various attack types and benign traffic. The subsequent subsections delve into the details of each phase, their components, and objectives.

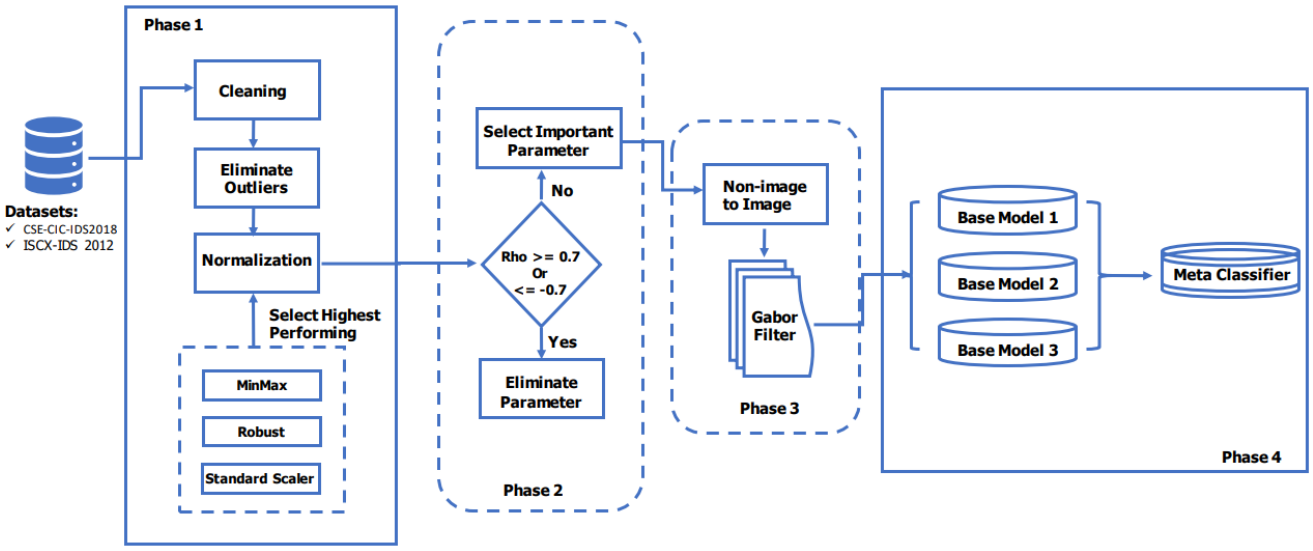


Figure 1. The proposed framework for detecting and classifying cyberattacks in IoT environments follows a coherent flow consisting of four interconnected phases: preprocessing, parameter selection, data transformation, and classification.

3.1. Preprocessing

This study utilizes three state-of-the-art datasets: CSE-CIC-IDS2018 [38] (Data A), ISCX-IDS-2012 [39] (Data B), and IoT-23 [40] (Data C). These datasets were collected over multiple days and accurately represent real network traffic. Due to their differences, the processing techniques applied to each dataset vary slightly. The datasets consist of multiple CSV files, which are merged together for both Data A and B. Column names are standardized by removing extra trailing spaces in Data B. Figure 2 shows the class distributions for Data A. Eleven classes were merged to form a "Web Attack" class for both Data A and Data B, resulting in the number of classes narrowing down to six for both datasets. The datasets contain various attack classes and a benign class, but some classes have limited samples. To ensure equal representation of all classes, data balancing is performed using the synthetic minority oversampling technique (SMOTE) [41]. SMOTE generates new synthetic samples from the minority class instead of duplicating existing samples. This ensures that the resulting model trained on the balanced data are not biased toward the majority class. The balanced Data A class distribution is shown in Figure 3.

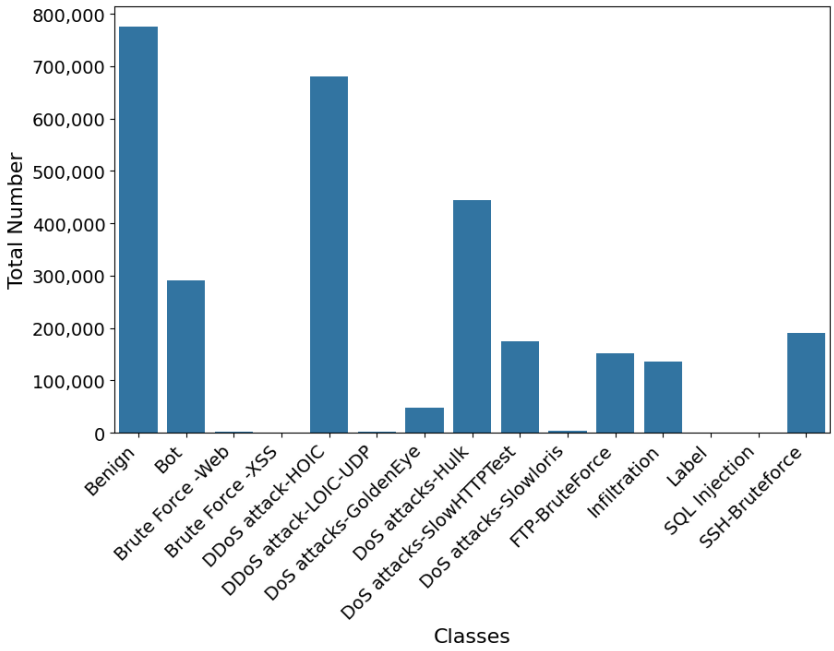


Figure 2. Class distribution for Data A showing an imbalanced distribution of number of samples per class.

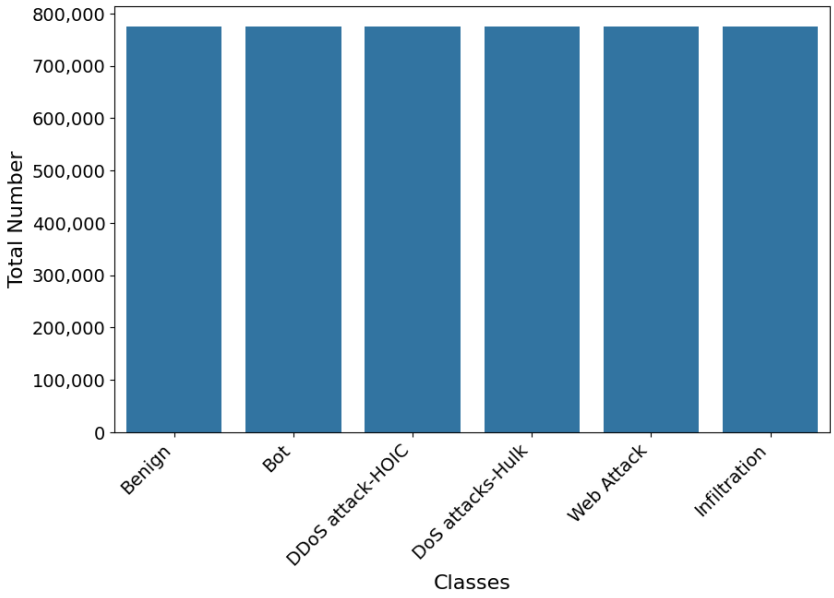
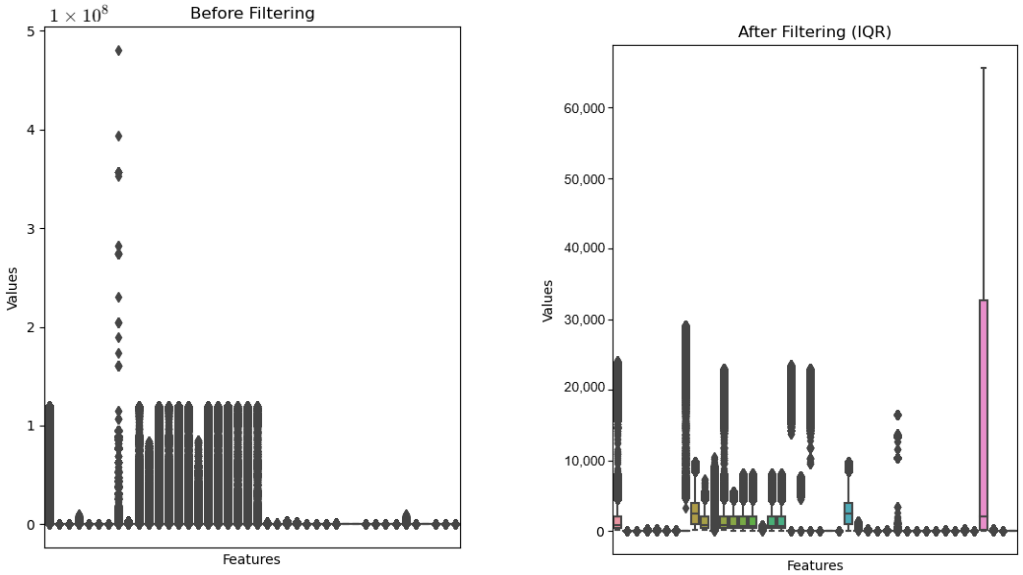


Figure 3. Class distribution for Data A where samples are evenly distributed.

3.1.1. Outlier Elimination

Outlier elimination is crucial in network security research to ensure data analysis reliability and accuracy. This study, we utilize the interquartile range (IQR) method to identify and remove outliers from the dataset. Outliers are extreme observations deviating significantly from the data distribution, which can distort statistical analyses. The IQR method calculates the range between the first quartile (Q1) and the third quartile (Q3), with data points falling below Q1 $-$ 1.5 $\times$ IQR or above Q3 + 1.5 $\times$ IQR considered outliers and excluded. By systematically comparing data points against the thresholds, outliers are eliminated iteratively. This results in a refined dataset representing the underlying data distribution, reducing bias and facilitating meaningful pattern extraction. Applying the IQR method improved analysis accuracy and reliability by focusing on the typical data range. Results of the before and after IQR implementation on Data A are shown in

Figure 4a, and 4b, respectively, revealing the presence of outliers in the initial images and their absence in the filtered images after IQR outlier removal. Figure 4a shows the dataset with extreme variations in feature values, some reaching 10^8. These high-magnitude outliers indicate severe skewness, which can distort model learning and lead to overfitting. Without filtering, certain features may dominate the training process, reducing the model's ability to distinguish normal and attack traffic patterns effectively, while Figure 4b demonstrates the dataset after IQR-based outlier removal, where extreme values have been mitigated. The feature distribution is now more balanced, reducing noise while preserving important patterns. This step prevents outliers from skewing the model, ensuring better generalization and improved classification accuracy when transforming network data into images for CNN-based analysis.



(**a**) (**b**)

Figure 4. Before and after filtering of Data A. (**a**) Data A before outlier elimination. (**b**) After outlier elimination using IQR on Data A.

3.1.2. Normalization

Normalization plays a crucial role in our network security research to standardize the data for fair comparisons. We utilize three common normalization methods: min–max, standard scaler, and robust scaler. The standard scaler is mathematically represented as in Equation (1):

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}. \tag{1}$$

where X_{scaled} is the normalized dataset, X is the original dataset, μ represents the mean, and σ is the standard deviation.

The mathematical representation of min–max is presented in Equation (2):

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2}$$

where X_{min} and X_{max} represents the minimum and maximum values, respectively.

Lastly, the robust scaler is represented mathematically in Equation (3):

$$X_{\text{scaled}} = \frac{X - \text{median}}{\text{IQR}} \tag{3}$$

where median represents the median value.

After applying each method, the mean, std, and IQR of the normalized data for all datasets are recorded. These measures characterized the central tendency, spread, and robustness of the data distribution. To compare the methods' performance, we calculated percentile ranks for the mean, std, and IQR of each method, providing a relative measure of how each method's statistics compared within the dataset. The highest-performing method is determined by computing an average percentile rank, obtained by averaging the percentile ranks of the mean, std, and IQR for each method. The method with the highest average percentile rank consistently exhibited desirable statistical characteristics across multiple measures, indicating its superior performance in standardizing the data.

By selecting the highest-performing normalization method, we ensured reliable and standardized inputs for data analysis and subsequent modelling processes. This choice minimized bias and enabled fair comparisons, enhancing the robustness and integrity of the network security research outcomes. The selected method's ability to consistently demonstrate desirable statistical properties contributed to the reliability and meaningfulness of our research findings, reinforcing the importance of employing appropriate normalization techniques in network security data analysis. Tables 1 and 2 show values obtained by each normalization method used for Data A and B, respectively. In Data A, the highest-performing method is the min–max scaler, while the robust scaler performed better in Data B. Hence, min–max is used in Data A and robust scaler for Data B for normalization.

Table 1. Mean, standard deviation, IQR, and percentile rankings of various normalization algorithms on Data A.

	Mean	Std.	IQR	Mean Percentile Rank	Std. Percentile Rank	IQR Percentile Rank	Average Percentile Rank
Standard Scaler	6.3172×10^{-16}	1.0000	0.0431	33.33	66.67	66.67	55.56
Min–Max Scaler	0.2364	0.3192	0.33	100.0	33.33	100.0	77.78
Robust Scaler	0.1917	9.3435	0.0	66.67	100.0	33.33	66.67

Table 2. Mean, standard deviation, IQR, and percentile rankings of various normalization algorithms on Data B.

	Mean	Std.	IQR	Mean Percentile Rank	Std. Percentile Rank	IQR Percentile Rank	Average Percentile Rank
Standard Scaler	-2.7035×10^{-17}	1.0000	0.2442	33.33	66.667	66.667	55.55
Min–Max Scaler	0.1086	0.1776	0.1667	66.67	33.3333	33.33	44.44
Robust Scaler	59.4385	1250.71	0.5264	100.0	100.0	100.0	100.0

3.2. Proposed Hybrid Filter-Based Parameter Selection

Feature selection is a crucial step in data analysis for identifying relevant parameters for accurate analysis. However, existing methods often struggle to capture complex relationships among features and optimize parameter selection simultaneously. To address this challenge, we propose a hybrid filter-based feature selection method that leverages correlation analysis and a chi-squared evaluation. This subsection introduces the motivation and objectives of the hybrid method. The main aim is to obtain a list of optimal features from the given dataset which will then be used to train and evaluate the model in the next phase.

The hybrid feature selection method combines correlation analysis and a chi-squared evaluation to identify uncorrelated parameters and select the most discriminative features. It begins by calculating correlation coefficients between features and then determines uncorrelated parameters based on significance level and correlation thresholds using Pearson's correlation coefficient (PCC) given in Equation (4).

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}, \tag{4}$$

where σ_X and σ_Y are the standard deviations of X and Y, respectively, and $\text{cov}(X,Y)$ is the covariance.

Figure 5 shows the correlation heatmap between features in Data A. The dark boxes represent highly correlated features that should be eliminated, while Figure 6 shows the same Data A after highly correlated features were eliminated. On their own, correlation tests for parameter/feature selection are limited by instability issues, and their results are unreliable for industrial applications. To mitigate this factor, we employ a chi-squared evaluation to measure the discriminative power of the selected parameters.

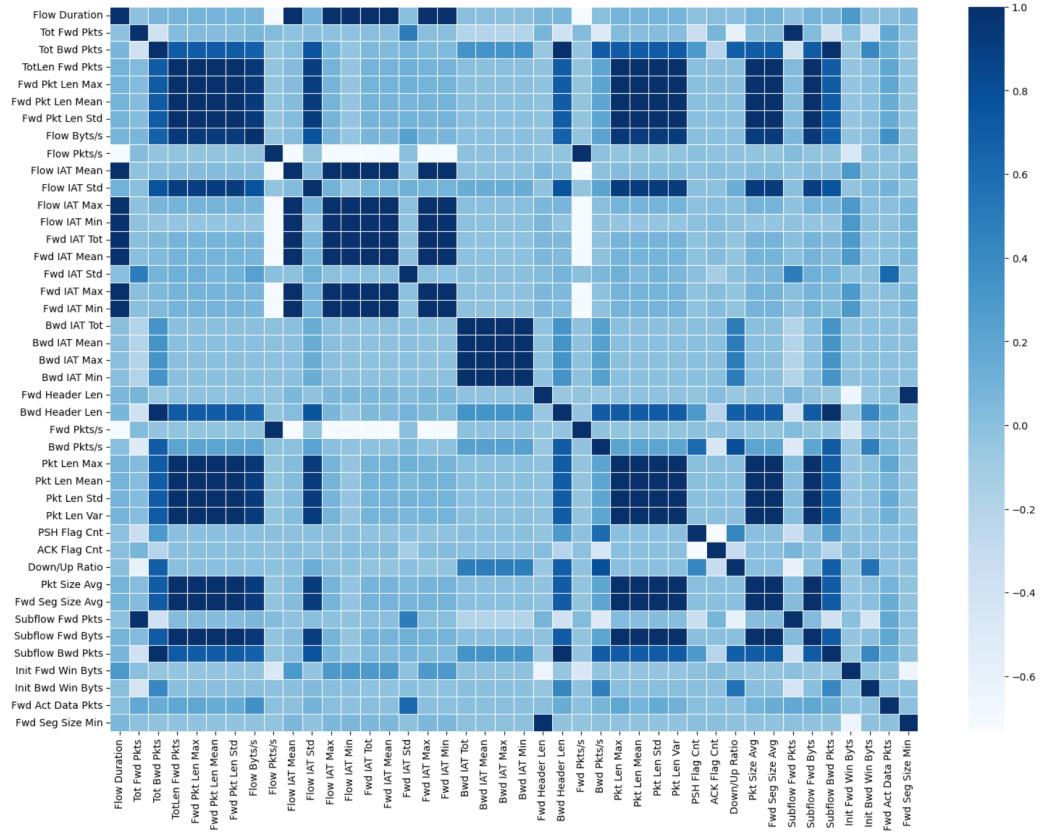


Figure 5. Correlation heatmap of Data A. It shows that highly correlated and redundant parameters exist in the dataset.

Chi-squared is denoted mathematically as

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}. \tag{5}$$

where c is the degree of freedom, O is observed values, and E_i is the expected values. A high chi-squared value indicates that the feature is dependent on the class and is retained, while a lower chi-squared value indicates independence between the parameter and the class. By using chi-square, we aim to detect which features are highly dependent on the classes and which are not. Features that are class-independent contribute little or nothing to the model and are hence not needed for classification. In Figure 7, the result of applying this ranked-based parameter selection to Data A is shown. Some features contributed very little, while some have a higher level of contribution. The five highest-ranked features are

selected, and these become the optimal features that would be used in training the model. This process is further explained in Algorithm 1.

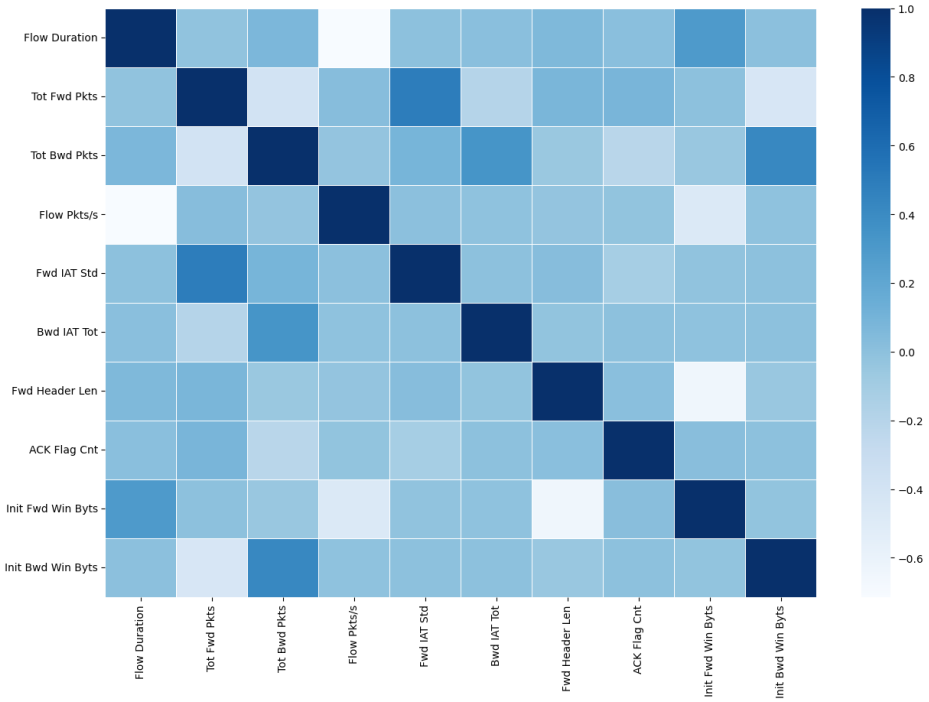


Figure 6. A plot of the resulting parameters after performing a correlation test and eliminating features whose correlation score is greater than the correlation threshold. NB: The darker the red, the higher the correlation. A feature will always have a high correlation with itself, hence the diagonal.

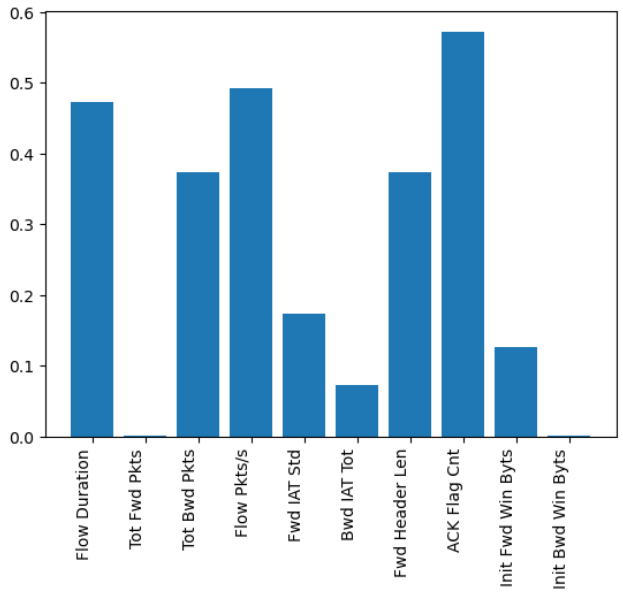


Figure 7. Feature ranking depicting features of Data A and their ranks. The five highest-ranked features are automatically selected from this stage.

The algorithm begins by calculating the correlation coefficients between each pair of features in the feature set F using Equation (4) and storing them in the correlation matrix C. Additionally, it calculates the p-value p_i for each feature. Next, the algorithm checks if a feature has a p-value less than or equal to the significance level P and is either negatively correlated (less than or equal to R_{neg}) or positively correlated (greater than or equal to R_{pos})

with another feature. If these conditions are met, the feature is considered uncorrelated and is added to the set of uncorrelated parameters F_s.

After identifying the uncorrelated parameters, the algorithm selects the most discriminative features from the set F_s. It calculates the chi-squared value between each pair of features in F_s using Equation (5) and stores them in a sorted array called chi_{array}. The top features with the highest chi-squared values are then selected and added to the set of optimal parameters F_c. Finally, the algorithm returns the sets F_s and F_c as the uncorrelated parameters and optimal parameters, respectively.

Algorithm 1: Pseudo-code of the proposed hybrid parameter selection method.

Input : $F = f_1, f_2, f_3, \ldots, f_k$: Original Feature Set;
D: Dimension of Parameters;
$P = 0.05$: Significance Level;
$R_{neg} = -0.7$: Negative Correlation Threshold;
$R_{pos} = 0.7$: Positive Correlation Threshold;
Output: F_s: Uncorrelated Parameters;
F_c: Optimal Parameters;

Step 1: Calculate the matrix C_{ij} of correlation coefficients between each f_i and f_j in F using Equation (4);

Step 2: Calculate the p-value p_i for each feature;

if $p_i \leq P$ *and* $(C_{ij} \leq R_{pos}$ *or* $C_{ij} \geq R_{neg})$ **then**
    return F_s with dimension d $(d < D)$;
else
    Ignore (F - F_s);
end

chi_array $\leftarrow \varnothing$;
max_no_of_features $\leftarrow$ 8;

for *each feature in the set* F_s $(i = 1, \ldots, d)$ **do**
    Calculate the chi-squared value (chi_val) between features in F_s using Equation (5);
    chi_val $\leftarrow$ chi.squared(F_{s_i}, F_{s_j});
    Append $(i,$ chi_val) to chi_array;
end

Sort chi_array in descending order;

$F_c \leftarrow$ chi_array[: max_no_of_features];

return F_c with dimension c $(c < d)$;

3.3. Network Non-Image Data Transformation to Image Data

Converting non-image data to images has gained significant attention in various domains due to its potential benefits and advantages. One compelling reason to consider this conversion is the ability to leverage the power of deep learning and image analysis techniques. Deep learning models, such as CNNs, have shown exceptional capabilities in image recognition and pattern detection. By transforming non-image data into images, researchers and practitioners can tap into the vast array of well-established image processing tools and methodologies, thereby unlocking the potential to extract meaningful patterns and insights from complex datasets [13]. Another key advantage of converting non-image

data to images is enhanced visualization and interpretability. Traditional tabular representations often struggle to convey the intricate relationships and patterns within the data. By mapping the tabular information onto images, the human visual system can exploit its innate ability to perceive and understand visual patterns more effectively [14]. Images offer a rich visual context that can provide intuitive insights into complex relationships, spatial distributions, and anomalies within the data. This visual representation facilitates a more comprehensive understanding of the underlying data structure, enabling researchers to make informed decisions, identify trends, and communicate findings more effectively.

Moreover, the transformation of network traffic data into images allows for the effective application of advanced image processing algorithms, such as the Gabor filter, which enhances discriminative features by capturing spatial and frequency patterns. This approach enables a more nuanced and robust analysis of attack patterns, addressing the limitations of traditional tabular and numerical data representations.

Phase 3 of the proposed IDS involves the transformation of the datasets into images suitable for a CNN model. To achieve this, DeepInsight [19] is implemented and applied to both datasets. DeepInsight was built by designing a feature-to-image transformation pipeline that arranges non-image data into a structured 2D space based on feature similarity. It first projects features onto a 2D plane using dimensionality reduction techniques (e.g., t-SNE or PCA) to preserve relationships. Next, it maps feature values to pixel intensities, forming an image representation. Finally, the transformed data are fed into CNNs, leveraging their ability to extract spatial patterns, making DeepInsight effective for non-image data classification. Figure 8 shows six samples (one from each class label) of the transformed Data A.

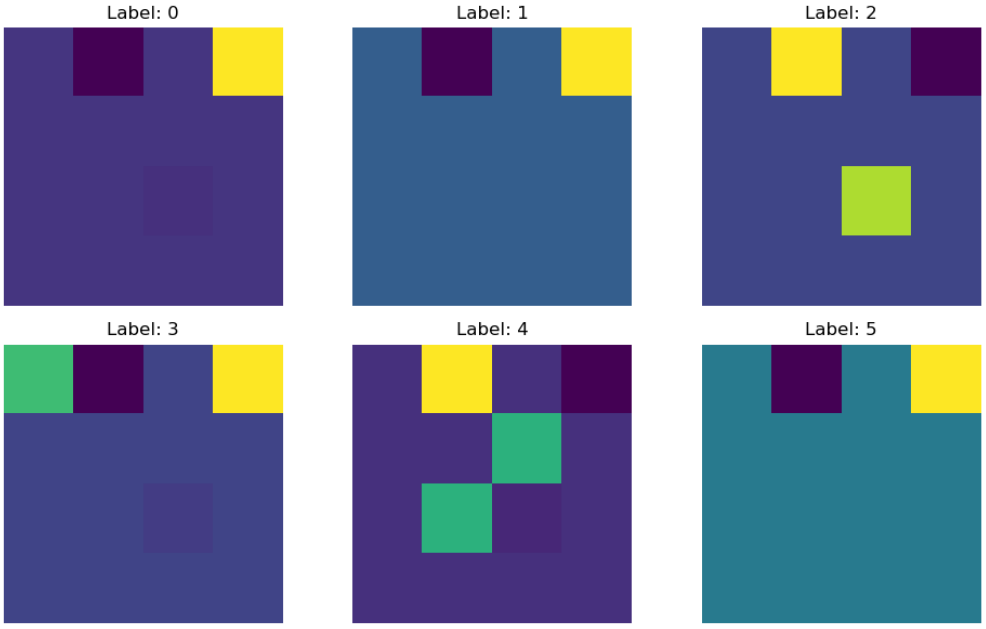


Figure 8. ample images of each label in Data A after non-image to image transformation.

Gabor Filter

After the data have been transformed into image data from non-image data, the newly generated image data are further transformed using the Gabor filter. A Gabor filter is a linear filter commonly used in image processing tasks. The Gabor filter convolves with an input image to produce a filtered image where a certain frequency and orientation information is enhanced, while other information is suppressed. The parameters for the Gabor filter include frequency, orientation, spatial aspect ratio, and bandwidth. It can be represented mathematically as Equation (6).

$$g(x,y) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \cdot \left(\cos\left(2\pi\frac{x'}{\lambda} + \phi\right)\right) \quad (6)$$

where $g(x,y)$ is the filter response at position (x,y), x' and y' are coordinates of (x,y), and σ is the standard deviation of the Gaussian envelope, controlling the spatial extent of the filter. λ determines the frequency response of the filter. ϕ is the phase offset of the filter. In Table 3, the parameters and their respective values used in this work are shown. The original image for one sample of each class label is displayed next to its filtered version for Data A in Figure 9.

The Gabor filter enhances the interpretability of image-transformed network data by isolating critical patterns that correspond to intrusion characteristics. This step not only refines the feature set but also significantly reduces noise, ensuring that the CNN model focuses on the most relevant information during the training process.

Table 3. Gabor filter parameters.

Parameter	Frequency	Sigma σ	Psi ψ	Theta θ	Gamma γ
Value	0.6	2.0	0.0	0.8	1.0

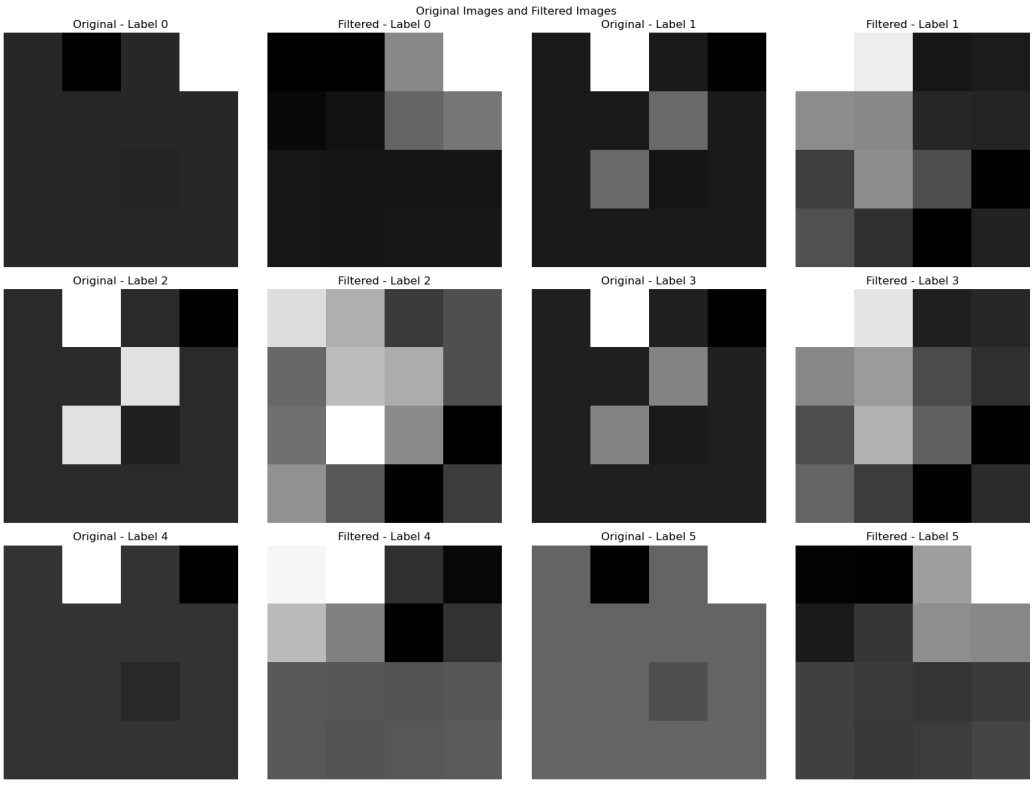


Figure 9. One sample from each label original and Gabor-filtered image of Data A.

3.4. Proposed Meta Classifier

The fourth and final phase of the proposed IDS is the classification phase, where a model is trained using the transformed data from phase 3. Ensemble learning is an all-encompassing term used for describing models that combines a number of classifiers or regressors to make a decision. It is the ML or DL way of implementing the wisdom of the crowd. The prevalent reason behind ensemble learning is the need to eliminate errors associated with stand-alone ML and DL models by combining multiple such models which will invariably lead to better prediction performance better than any single model [42]. Ensemble learning has been shown to have a range of positive impacts on intrusion

detection systems. They have improved classification accuracy and are more resilient to noisy data [43].

This work proposes the use of an ensemble CNN model. The ensemble CNN model combines the predictions of multiple base CNN models to improve overall performance and enhance the accuracy of the IDS. Each base CNN model has its own unique architecture, hyperparameters, and feature representations, contributing to diverse perspectives and capturing different aspects of the data. The model consists of three base models and a meta-classifier. The first base CNN model (Model 1) is constructed with a 2D convolutional layer with 32 filters and a ReLU activation function. It is followed by a 2D max-pooling layer with a pool size of (2, 2) a dropout layer with a rate of 0.25, and an Adam optimizer. The output is then flattened to be compatible with subsequent layers. The second base CNN model (Model 2) has a similar architecture to Model 1 but utilizes a 2D convolutional layer with 64 filters. The third base CNN model (Model 3) employs a 2D convolutional layer with 128 filters and a ReLU activation function. It is followed by a 2D max-pooling layer with a pool size of (3, 3) a dropout layer with a rate of 0.2, and an Adamax optimizer. All models are compiled using the categorical_crossentropy loss function. Table 4 provides a detailed overview of the parameters used in the models.

Table 4. Base models' configuration details.

Model	Filter	Kernel Size	Pool Size	Dropout	Optimizer
Model 1	32	3×3	2×2	0.25	Adam
Model 2	64	3×3	2×2	0.25	Adam
Model 3	128	3×3	3×3	0.20	Adamax

To leverage the predictive capabilities of these base models, they are combined using a meta-classifier that incorporates voting to aggregate predictions from individual CNN models. The voting aggregator is configured with the three base models (Model 1, Model 2, and Model 3) as its estimators. By using a hard-voting strategy, the final prediction is determined by the majority vote of the base models. By integrating multiple base CNN models within a meta-classifier, our approach aims to improve classification accuracy and robustness. This ensemble design allows us to leverage the complementary strengths of the individual models, enhancing the overall performance of the classification task.

4. Experimentation and Results Analysis

4.1. Setup

The proposed scheme was implemented on an Ubuntu server equipped with a Jupyter notebook environment. The system configuration included three NVIDIA GeForce RTX 3090 GPUs with CUDA version 12.1. For the transformation of non-image data into image data, DeepInsight [19] was utilized. The Gabor filter, employed for image transformation, was implemented using the OpenCV cv2 library. To assess the effectiveness of the proposed framework, several evaluation metrics were employed, including accuracy, precision, F1 score, and recall. Additionally, the impact of the Gabor filter on the framework and the testing time was evaluated.

4.2. Dataset

This study utilizes three state-of-the-art datasets: CSE-CIC-IDS2018 [38] (Data A), ISCX-IDS-2012 [39] (Data B), and IoT-23 [40] (Data C). These datasets were collected over multiple days and accurately represent real network traffic. Due to their differences, the processing techniques applied to each dataset vary slightly as detailed in Section 3.1.

4.3. Baselines

The first base CNN model (Model 1) is constructed with a 2D convolutional layer with 32 filters and a ReLU activation function. It is followed by a 2D max-pooling layer with a pool size of (2, 2) a dropout layer with a rate of 0.25, and an Adam optimizer. The output is then flattened to be compatible with subsequent layers. The second base CNN model (Model 2) has a similar architecture to Model 1 but utilizes a 2D convolutional layer with 64 filters. The third base CNN model (Model 3) employs a 2D convolutional layer with 128 filters and a ReLU activation function. It is followed by a 2D max-pooling layer with a pool size of (3, 3) a dropout layer with a rate of 0.2, and an Adamax optimizer. All models are compiled using the categorical_crossentropy loss function. This paper makes use of KerasTuner to specify different hyperparameters, with the best-performing hyperparameters being retained. Table 4 provides a detailed overview of the parameters used in the models.

To assess the effectiveness of the proposed hybrid filter-based parameter selection algorithm, we conducted experiments considering three different scenarios. In Scenario 1, the proposed algorithm was utilized for parameter selection. In Scenario 2, parameter selection was performed based solely on PCC without incorporating the chi-squared feature ranking. Lastly, Scenario 3 involved no parameter selection algorithm.

4.4. Evaluation Metrics

To evaluate the effectiveness of the proposed IDS, the following metrics were used:

Accuracy is the fraction of correctly classified traffic (including both normal and malicious) out of all traffic.
Recall measures the proportion of actual intrusions that the IDS successfully detects.
Precision measures the proportion of correctly identified intrusions out of all instances classified as intrusions.
F1 score is the harmonic mean of precision and recall, providing a single metric to balance between the two.

To evaluate the computational efficiency of the proposed IDS, the following metrics were used:

Testing/inference time is the average time taken to classify a single instance during testing, indicating the system's suitability for real-time detection.

4.5. Results

The performance metrics of the base models and the proposed ensemble CNN are summarized in Tables 5–7 for Data A, B, and C, respectively. Table 5 shows the accuracy, F1 score, recall, and precision of the three base models (Base Model 1, Base Model 2, and Base Model 3) and the proposed scheme, while Table 6 presents the corresponding metrics all models on Data A.

Table 5. Performance metrics of base models and proposed scheme on Data A.

Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
Base Model 1	92.34	91.57	93.12	90.98
Base Model 2	93.27	92.11	94.06	91.86
Base Model 3	94.81	93.74	95.29	92.93
Proposed	98.12	98.05	98.19	98.11

Among the base models, Base Model 3 achieved the highest accuracy of 94.81%, with an F1 score of 93.74%, recall of 95.29%, and precision of 92.93%. Base Model 2 and

Base Model 1 also exhibited strong performance, with accuracies of 93.27% and 92.34%, respectively. The proposed ensemble CNN outperformed all base models, achieving an impressive accuracy of 98.12%, an F1 score of 98.05%, a recall of 98.19%, and a precision of 98.11% on Data A. These results demonstrate the effectiveness of the proposed classifier in improving classification performance compared to individual base models. The higher accuracy and balanced F1 score, recall, and precision of the meta-classifier validate its suitability for the task at hand.

Table 6. Performance metrics of base models and proposed scheme on Data B.

Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
Base Model 1	91.76	90.45	92.34	89.76
Base Model 2	93.12	91.98	94.27	90.84
Base Model 3	92.87	91.73	93.94	90.21
Proposed	96.45	95.62	96.78	94.87

Table 7. Performance metrics of base models and proposed scheme on Data C.

Model	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)
Base Model 1	92.10	90.80	92.70	90.10
Base Model 2	93.50	92.30	94.60	91.20
Base Model 3	93.20	92.00	94.30	90.80
Proposed	97.02	96.10	97.25	95.34

The test times (TT) for different scenarios of Data A and B are provided in Table 8. The test time is reported in seconds and represents the time taken by the models for testing. For Data A, the test time for Scenario 1 was 1.9 s, 4.2 s for Scenario, and 12 s for Scenario 3. On the other hand, for Data B, the test time for Scenario 1 was 2 s, 5 s for Scenario 2, and 10.1 s for Scenario 3. These results indicate that the test time varied across different scenarios. The longer test times observed in both Scenario 2 and 3 for both datasets can be attributed to the increased complexity and occurrence of redundant and highly correlated features of the data, resulting in more computation and processing time. It is important to consider these test times in practical applications where real-time or time-sensitive processing is required as is the case of IDSs. These insights into the test times provide valuable information for system design and optimization in terms of computational resources and efficiency.

Table 8. Test time of various parameter selection algorithms for Data A and B.

Dataset	Scenario 1 TT (s)	Scenario 2 TT (s)	Scenario 3 TT (s)
Data A	1.9	4.2	12
Data B	2	5	10.1

The impact of the Gabor filter on the framework is studied and results are recorded in Table 9. For Data A, the accuracy with the Gabor filter was 98.12%. Without the filter, it achieved an accuracy of 95.86%. Similarly, for Data B, the accuracy with the Gabor filter was 96.45%. Without the filter, it achieved an accuracy of 96.02%. These results demonstrate the impact of the Gabor filter on the classification performance of the model. The higher accuracy observed with the Gabor filter indicates its effectiveness in enhancing the discriminative features and improving the classification accuracy. The improvement in accuracy can be attributed to the filter's ability to extract relevant spatial frequency information from the input data, which aids in better representation and discrimination of the classes. Although it is important to note that the use of the Gabor filter may introduce

additional computational complexity due to the additional processing steps involved. However, the significant improvement in accuracy justifies its inclusion in the classification pipeline, particularly in scenarios where achieving higher accuracy is a priority. Moreover, this potential additional computational complexity is mitigated by the proposed parameter selection algorithm. The inclusion of image processing techniques, particularly the Gabor filter, significantly improves the IDS's classification performance by enhancing feature representation, as evidenced by the accuracy improvements in Table 9.

Table 9. Impact of Gabor filter on models' accuracy with both datasets.

Dataset	Accuracy with Gabor Filter (%)	Accuracy without Filter (%)
Data A	98.12	95.86
Data B	96.45	96.02

Finally, in Table 10, the performance of our proposed scheme was compared to models from the literature [28,30,32,34]. Although [28] slightly outperforms our model in all metrics, it comes at a significantly higher computational complexity, as reflected in the loner testing time of 7.1 s. In contrast, our proposed model achieves a strong balance between accuracy (98.12%) and computational efficiency (1.9 s testing time), demonstrating its practicality for real-time intrusion detection.

Table 10. Proposed scheme comparison with other frameworks.

Ref	Accuracy (%)	F1 Score (%)	Recall (%)	Precision (%)	Testing Time (s)
[28]	99.90	99.90	99.90	99.90	7.1
[30]	98.42	94.5	92.81	96.44	-
[32]	98.75	98.67	98.69	98.65	
[34]	97.07	94.46	-	-	2.41
Proposed	98.12	98.05	98.19	98.11	1.9

Despite employing image transformation techniques, which often introduce computational overhead, our approach successfully maintains a high detection performance while significantly reducing testing times compared to traditional deep learning-based IDS models. This result validates the efficiency of our proposed scheme, ensuring that IDSs remain both effective and scalable for IoT.

4.6. Limitations and Future Works

While the proposed IDS framework demonstrates significant improvements in intrusion detection for IoT environments, several limitations remain that warrant further investigation.

A key limitation of this study is the reliance on public datasets rather than real-world IoT network validations. Although benchmark datasets offer consistency, they may not fully represent the complexity and evolving threats in real IoT environments. Real-world testing is needed to evaluate adaptability and performance under dynamic conditions.

Additionally, this study evaluates computational efficiency using inference time. Metrics such as energy consumption, memory footprint, and latency are crucial for assessing feasibility in resource-constrained IoT devices. These metrics can be measured when deployed in an IoT environment, as mentioned above. Future work will incorporate these aspects to better understand deployment trade-offs and scalability.

5. Conclusions

This work presents a comprehensive four-phase architecture for detecting and classifying cyberattacks in IoTs. The four phases are data preprocessing, feature selection, image transformation, and ensemble classification. In the first phase, effective data preprocessing techniques are employed, including outlier removal and data normalization using the best-performing algorithm. This ensures that the data are prepared appropriately for subsequent analysis. The second phase focuses on mitigating the curse of dimensionality through a hybrid filter-based parameter selection approach. Redundant and highly correlated features were eliminated, and the remaining features were ranked based on their importance to the classifier. This resulted in improved detection performance with a reduced computational cost. In the third phase, the data were transformed into image representations, and the Gabor filter was applied to capture relevant shape information. This enabled the utilization of image processing techniques and enhanced the accuracy of the detection system. The final phase involved the implementation of an ensemble CNN model, comprising three distinct base models and a meta-classifier. The ensemble model combines the predictive powers of the base models, resulting in improved accuracy and robustness in attack detection and classification. Experimental results demonstrate the effectiveness of the proposed framework. The filter-based parameter selection approach significantly reduces computational costs while maintaining high accuracy. The meta-classifier outperforms the individual base models, highlighting the benefits of combining diverse perspectives in model training. Additionally, the application of the Gabor filter further improves the accuracy of the system. By leveraging salient data, incorporating a meta-classifier, and utilizing the Gabor filter, the proposed framework achieves a balance between computational efficiency and accurate attack detection. It provides a robust solution for intrusion detection in IoT environments, with potential applications in enhancing network security and protecting IoT devices from cyber threats.

Author Contributions: Conceptualization, G.C.A.; Methodology, G.C.A.; Formal analysis, G.C.A.; Investigation, G.C.A.; Writing—original draft, G.C.A.; Writing—review & editing, A.M.V.V.S. and M.S.; Supervision, A.M.V.V.S. and D.-S.K.; Project administration, D.-S.K.; Funding acquisition, A.M.V.V.S. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2024-2020-0-01612, 50%) and Priority Research Centers Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology. This research was also supported in part by National Science Foundation (NSF) of USA under Grant No. 2200673 and Towson University's

Data Availability Statement: The data presented in this study are available in [UNB] at [https://www.unb.ca/cic/datasets/ids-2018.html], [UNB] at [https://www.unb.ca/cic/datasets/ids.html], and [Zenodo] at [https://zenodo.org/records/4743746], all accessed on 30 December 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hu, Y.; Chen, C.; He, J.; Yang, B.; Guan, X. IoT-Based Proactive Energy Supply Control for Connected Electric Vehicles. *IEEE Internet Things J.* **2019**, *6*, 7395–7405. [CrossRef]
2. Bianchi, V.; Bassoli, M.; Lombardo, G.; Fornacciari, P.; Mordonini, M.; De Munari, I. IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment. *IEEE Internet Things J.* **2019**, *6*, 8553–8562. [CrossRef]

3. Verma, P.; Sood, S.K. Fog Assisted-IoT Enabled Patient Health Monitoring in Smart Homes. *IEEE Internet Things J.* **2018**, *5*, 1789–1796. [CrossRef]
4. Nwakanma, C.I.; Islam, F.B.; Maharani, M.P.; Lee, J.M.; Kim, D.S. Detection and Classification of Human Activity for Emergency Response in Smart Factory Shop Floor. *Appl. Sci.* **2021**, *11*, 3662. [CrossRef]
5. Tran-Dang, H.; Krommenacker, N.; Charpentier, P.; Kim, D.S. The Internet of Things for Logistics: Perspectives, Application Review, and Challenges. *IETE Tech. Rev.* **2020**, *39*, 93–121. [CrossRef]
6. Large Scale DDoS Attack. Available online: https://github.com (accessed on 10 March 2022).
7. Tran-Dang, H.; Krommenacker, N.; Charpentier, P.; Kim, D. Toward the Internet of Things for Physical Internet: Perspectives and Challenges. *IEEE Internet Things J.* **2020**, *7*, 4711–4736. [CrossRef]
8. Bertino, E.; Islam, N. Botnets and Internet of Things Security. *Computer* **2017**, *50*, 76–79. [CrossRef]
9. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 303–336. [CrossRef]
10. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [CrossRef]
11. Alsoufi, M.A.; Razak, S.; Siraj, M.M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 8383. [CrossRef]
12. Abdelmoumin, G.; Rawat, D.B.; Rahman, A. On the Performance of Machine Learning Models for Anomaly-Based Intelligent Intrusion Detection Systems for the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 4280–4290. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
14. Li, L.; Gan, Z.; Cheng, Y.; Liu, J. Relation-Aware Graph Attention Network for Visual Question Answering. *arXiv* **2019**, arXiv:1903.12314. Available online: http://arxiv.org/abs/1903.12314 (accessed on 30 December 2024).
15. Mohamad, M.; Selamat, A.; Krejcar, O.; Fujita, H.; Wu, T. An analysis on new hybrid parameter selection model performance over big data set. *Knowl.-Based Syst.* **2020**, *192*, 105441. [CrossRef]
16. Visalakshi, S.; Radha, V. A literature review of feature selection techniques and applications: Review of feature selection in data mining. In Proceedings of the 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2014; pp. 1–6. [CrossRef]
17. Khaire, U.M.; Dhanalakshmi, R. Stability of feature selection algorithm: A review. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *34*, 1060–1073. [CrossRef]
18. Kumar, D.A.; Rengasamy, R. Parameterization reduction using soft set theory for better decision making. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, Salem, India, 21–22 February 2013; pp. 365–367. [CrossRef]
19. Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.A.; Tsunoda, T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **2019**, *9*, 11399. [CrossRef] [PubMed]
20. Hong, K.; Kim, Y.; Choi, H.; Park, J. SDN-Assisted Slow HTTP DDoS Attack Defense Method. *IEEE Commun. Lett.* **2018**, *22*, 688–691. [CrossRef]
21. Abidoye, A.P.; Obagbuwa, I.C. DDoS attacks in WSNs: Detection and countermeasures. *IET Wirel. Sens. Syst.* **2018**, *8*, 52–59. [CrossRef]
22. Sicari, S.; Rizzardi, A.; Miorandi, D.; Coen-Porisini, A. REATO: REActing TO Denial of Service Attacks in the Internet of Things. *Comput. Netw.* **2018**, *137*, 37–48. [CrossRef]
23. Yan, Q.; Huang, W.; Luo, X.; Gong, Q.; Yu, F.R. A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 30–36. [CrossRef]
24. Illy, P.; Kaddoum, G.; Miranda Moreira, C.; Kaur, K.; Garg, S. Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019. [CrossRef]
25. Lian, W.; Nie, G.; Jia, B.; Shi, D.; Fan, Q.; Liang, Y. An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning. *Math. Probl. Eng.* **2020**, *2020*, 2835023. [CrossRef]
26. Thanh, H.N.; Lang, T.V. Use the ensemble methods when detecting DoS attacks in Network Intrusion Detection Systems. *EAI Endorsed Trans. Context Aware Syst. Appl.* **2019**, *6*, e5. [CrossRef]
27. Jabbar, M.A.; Aluvalu, R.; Reddy, S.S.S. Cluster Based Ensemble Classification for Intrusion Detection System. In Proceedings of the 9th International Conference on Machine Learning and Computing, ICMLC 2017, Singapore, 24–26 February 2017; pp. 253–257. [CrossRef]
28. Baz, M. SEHIDS: Self Evolving Host-Based Intrusion Detection System for IoT Networks. *Sensors* **2022**, *22*, 6505. [CrossRef]
29. Kumar Samriya, J.; Kumar, S.; Kumar, M.; Wu, H.; Singh Gill, S. Machine Learning-Based Network Intrusion Detection Optimization for Cloud Computing Environments. *IEEE Trans. Consum. Electron.* **2024**, *70*, 7449–7460. [CrossRef]

30. Ben Said, R.; Sabir, Z.; Askerzade, I. CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection. *IEEE Access* **2023**, *11*, 138732–138747. [CrossRef]
31. Liu, W.; Liu, X.; Di, X.; Qi, H. A novel network intrusion detection algorithm based on Fast Fourier Transformation. In Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; pp. 1–6. [CrossRef]
32. Khan, A.S.; Ahmad, Z.; Abdullah, J.; Ahmad, F. A Spectrogram Image-Based Network Anomaly Detection System Using Deep Convolutional Neural Network. *IEEE Access* **2021**, *9*, 87079–87093. [CrossRef]
33. Murtaza Ahmed Siddiqi, W.P. An Optimized and Hybrid Framework for Image Processing Based Network Intrusion Detection System. *Comput. Mater. Contin.* **2022**, *73*, 3921–3949. [CrossRef]
34. Bouayad, A.; Alami, H.; Janati Idrissi, M.; Berrada, I. Lightweight Federated Learning for Efficient Network Intrusion Detection. *IEEE Access* **2024**, *12*, 172027–172045. [CrossRef]
35. Siddiqi, M.A.; Pak, W. Tier-Based Optimization for Synthesized Network Intrusion Detection System. *IEEE Access* **2022**, *10*, 108530–108544. [CrossRef]
36. Badr, W. Why Feature Correlation Matters.... A Lot. *Towards Data Sci.* **2019**, *22*. Available online: https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4/ (accessed on 30 December 2024).
37. Rajaby, E.; Sayedi, S.M. A structured review of sparse fast Fourier transform algorithms. *Digit. Signal Process.* **2022**, *123*, 103403. [CrossRef]
38. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the International Conference on Information Systems Security and Privacy, Funchal-Madeira, Portugal, 22–24 January 2018.
39. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
40. Garcia, S.; Parmisano, A.; Erquiaga, M.J. IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic. Zenodo, 2020. Available online: https://zenodo.org/records/4743746 (accessed on 30 December 2024).
41. Chawla, N.V.; Bowyer, K.; Hall, L.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
42. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
43. Gao, X.; Shan, C.; Hu, C.; Niu, Z.; Liu, Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access* **2019**, *7*, 82512–82521. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.