

RESEARCH ARTICLE

Tier-Based Optimization for Synthesized Network Intrusion Detection System

MURTAZA AHMED SIDDIQI AND WOOGUIL PAK 

Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding author: Wooguil Pak (wooguilpak@yu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) under Grant NRF-2022R1A2C1011774 and in part by 2022 Yeungnam University Research Grant.


ABSTRACT The innovation and evolution of hacking methodologies have led to a sharp rise in cyber attacks, highlighting the need for enhanced network security approaches. Network intrusion detection systems based on machine learning are playing a significant role in the domain of network security. However, designing an optimal framework for a network intrusion detection system is an ongoing concern. In this study, an optimal framework for a network intrusion detection system based on image processing is proposed. The framework is a fusion of augmented feature selection flow with an image transformation and enhancement methodology. Initially, the proposed framework reduces the number of features to achieve overall efficiency. Later, the non-image data is transformed into images. The transformed images are then enhanced for achieving effective anomaly detection based on a deep-learning classifier. The proposed method is implemented on three diverse benchmark datasets of intrusion detection. To illustrate the efficiency of the proposed framework it is compared with some of the most recent publications on image-processing-based network intrusion detection systems.

INDEX TERMS CNN, CSE-CIC-IDS 2018, CIC-IDS 2017, ISCX-IDS 2012, intrusion detection, network intrusion detection system.

I. INTRODUCTION

The pervasive use of interconnected computer systems has become an irreplaceable aspect of organizational and daily life activities. Concurrently, it had led to concerns about the online privacy and security of the users [1], [2]. As per recent surveys, the reported cyberattacks in 2021 were approximately 5.1 billion [3], [4]. The reports also indicate a surge in sophisticated and high-impact cyberattacks on critical infrastructure globally [4], [5]. Understandably, such a high number of cyberattacks indicate the need for enhancement in network security approaches. Machine Learning (ML) based Network Intrusion Detection Systems (NIDS) are considered to be among the most effective approach to counter network attacks. However, sustaining the efficiency and effectiveness of ML-based NIDS against ever-mutating network attacks is a highly challenging task. Designing an optimal framework for

ML-based NIDS is an ongoing struggle [6], [7], [8]. There is a constant compromise between achieving high efficiency and effectiveness. The ML-based NIDS with high efficiency may not be highly effective, while the one with high effectiveness may not be highly efficient [9], [10]. In efforts to optimize ML-based NIDS, researchers have worked on multidimensional approaches i.e. feature selection, data augmentation, classification algorithms, and hybrid algorithms to optimize the NIDS framework [11], [12]. Even with all the efforts, the degree of successful malicious attacks is increasing rapidly. Hence, a refined and scalable intrusion detection method is essential to counter the cybersecurity concern. With the advancements in Deep Learning (DL) and image processing, security experts are exploring the possibilities of using DL and image processing for NIDS [13], [14], [15], [16], [17]. DL is an improved form of the neural network (NN) as it overcomes three significant training phase issues of NN i.e. over-fitting, vanishing gradient, and computational load [18]. The convolutional neural network (CNN) is among the DL

The associate editor coordinating the review of this manuscript and approving it for publication was Marina Gavrilova .

models that are designed predominantly for image data [19]. CNN is among the recent and highly accurate classification approaches in image processing [20]. In the last few decades, the use of image processing in the healthcare industry has obligated researchers for achieving extreme precision in image analysis, detection, and classification [21]. The exploration of image processing for NIDS is intriguing due to the high precision results achieved by CNN and image processing methods. The fusion of image processing in NIDS is relatively new and requires innovation. One of the major concerns for image processing-based NIDS is the conversion of non-image network traffic into images for visual processing. A few of the prominent methods for converting the non-image data into images are by converting a one-dimensional vector to a multi-dimensional matrix, using the Fourier domain, and using spectrogram-based image transformation [14], [15], [16]. The mentioned approaches do have concerns regarding general application and image transformation results, which are discussed in the related work section. Such issues have opened doors for further exploration of methods that can improve the conversion of non-image data into images. In our earlier study [17], we implemented the image processing-based NIDS with CNN based classifier. In that study, we used all the features of the implemented datasets. The inclusion of all features was based on the notion that higher pixel images are conducive to detecting anomalies [22]. In the prior study [17] the accuracy of anomaly detection was above 90% on all the datasets. This work is an enhancement to the earlier work by augmenting the proposed framework through feature selection. This study attempts to contribute to the two key areas of image-based NIDS. First, the framework uses a reformed filter-based feature selection flow to achieve overall optimization of the NIDS. The augmented feature selection approach increases the overall efficiency of the NIDS. The second is an innovative framework to transform non-image data into image format. The method of transforming non-image data into images can further be divided into two steps. Initially, the framework transforms non-image data into images. Later, the converted images are enhanced to attain improved anomaly detection using a CNN-based classifier. Even with the fewer pixels of image representation, the proposed framework achieved a detection rate of over 92% on CSE-CIC-IDS 2018 [23], CIC-IDS 2017 [24], and ISCX-IDS 2012 [25] datasets.

The remaining of the paper is structured as follows: Section II discusses the related studies on ML, DL, and image processing-based NIDS. Section III presents the proposed methodology and elaborates on each step of the framework. Section IV gives details on the implementation of the proposed NIDS. Section V highlights the results and comparison of the proposed and recent prominent image-based NIDS approaches. Section VI discusses the outcomes of the proposed methodology in contrast to the results of the implemented comparative approaches. Section VII concludes the study with a future direction of research.

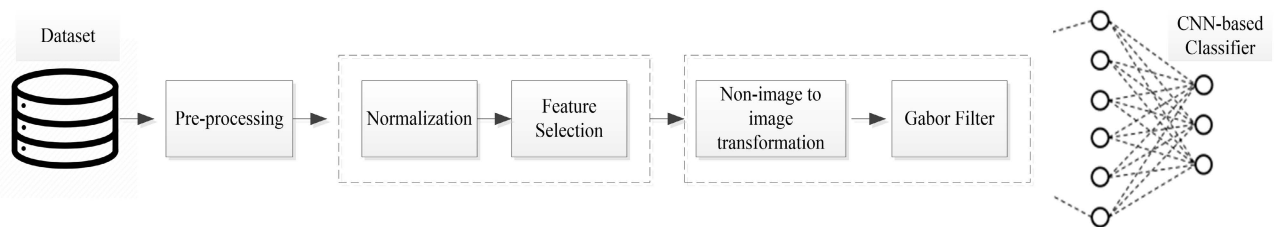
II. RELATED WORK

The researchers have worked extensively to incorporate ML in NIDS. Despite the extended research, the struggle to achieve an optimal framework for ML-based NIDS is a challenging and ongoing task. To optimize ML-based NIDS researchers have explored hybrid and innovative approaches for data pre-processing, feature selection, and prediction algorithms. In recent developments, researchers are exploring DL for NIDS solutions [26]. Among these DL models, CNN is considered a highly effective and efficient model. Generally due to its ability to reconstruct features and learn in-depth patterns from images [27]. Table 1 represents a summary of recent and prominent publications in the domain of image processing-based NIDS.

As seen in Table 1, most of the papers use two main approaches to convert non-image data into image format. One is to simply transform one-dimensional data into a multi-dimensional matrix. The second is to use the Fourier domain for the transformation of non-image data into image format. Both of the mentioned methods have some advantages and disadvantages. For instance, the first approach is highly efficient but can compromise the correlation between features [32]. Such a compromise can influence the NIDS's ability to detect sophisticated attacks. On the other hand, image transformation by using Fourier domain-based may have complexity issues when it comes to big data [33]. The mentioned issues highlight the room for improvement in the domain of converting non-image network traffic into image format. The application of CNN in the domain of NIDS brings high precision for classification. However, this high precision relies heavily on the transformation of non-image network data into images for a CNN-based classifier. For example, Xiao et al. [28] use a fusion of principle component analysis (PCA) and auto-encoder (AE) for feature engineering, and then two-dimensional images were created for a CNN-based classifier. The proposed hybrid approach was not so successful in detecting minor attack labels in the dataset. Similarly, Zhang et al. [34] proposed a highly complex approach for converting non-image data into images. The proposed approach used the P-Zigzag algorithm for creating two-dimensional greyscale images for CNN (gcForest) classifier. Despite the high computational cost, the proposed model was very effective in detecting anomalies. Further, Jiang et al [35] proposed an effective but highly complex IDS approach. The proposed framework initially balances the dataset by using the one-side selection (OSS) to decrease a large number of samples in the main category and Synthetic Minority Over-sampling Technique (SMOTE) to upsurge the samples of minority samples. After data balancing, the spatial features are extracted using CNN, and temporal features are extracted using a Bi-directional long short-term memory (BiLSTM). The fusion of CNN and BiLSTM also creates a deep layered network for classification. The discussed research highlights two main concerns of image processing-based NIDS. One is the continuous challenge of achieving

TABLE 1. Summary of recent and prominent publications in the domain of image processing-based NIDS.

Reference	Classifier	Datasets	Published	Proposed approach
[13]	ResNet-50 (CNN)	UNSW-NB15, BOUN Ddos	2021	The network traffic features are reshaped into 4 channel images (Alpha, Green, Red, and Blue). The images are then used to train and test the classifier.
[14]	CNN	NSL-KDD	2019	The network traffic is converted to images by Fast Fourier Transformation (FFT) for classification.
[15]	CNN	CIC-IDS2017	2021	The network traffic is transformed to spectrogram images with the help of a short-time Fourier transform (STFT) for anomaly detection.
[16]	CNN	KDD-CUP99, CSE-CIC-IDS2018	2020	The network traffic is changed into two types of images. One 2D grayscale image. Second, 2D RGB (red, blue, green) images for classification.
[28]	CNN	KDD-CUP99	2019	The 1D network traffic is simply converted to a 2D image representation for CNN-based classification.
[29]	CNN	KDD-CUP99, NSL-KDD	2020	After preprocessing the datasets, the one-dimensional dataset of network intrusion detection are transformed into two-dimensional data for CNN-based feature selection and classification.
[30]	CNN	UNSW-NB15	2021	After data preprocessing the dataset is reshaped into 14*14 matrices. The matrices are then converted into gray-scale images for CNN based classifier.
[31]	CNN	CIC-IDS2017, Car-Hacking dataset	2022	After normalizing the dataset, the data samples were chunked based on feature size and timestamps of the network traffic. Then the datasets were converted into three color images (i.e. CICIDS2017 with 20 features was converted to 20*20*3 matrix for RGB images) for a CNN-based classifier.

**FIGURE 1.** Overall flow of the proposed framework.

an optimized NIDS framework. Second is room for improvement in the approach of converting non-image network data into image format.

III. PROPOSED METHOD

The proposed framework is a fusion of two phases. In this section, both fragments will be discussed separately to give a clear idea of the proposed framework. First, the augmented flow of feature selection and data transformation is discussed. Second, the process of converting non-image data to image format is elaborated. Figure 1 represents the overall flow of the proposed NIDS framework.

A. DATASET PRE-PROCESSING

The datasets used for the implementation of the proposed NIDS framework are CSE-CIC-IDS 2018 [23], CIC IDS 2017 [24], and ISCX IDS 2012 [25]. The mentioned datasets are among the benchmarked and well-known datasets for testing NIDS [36]. The datasets are generated by modeling real-world traffic and attack patterns. To create the datasets the attacks were generated for several days based on existing tools and profiles. Datasets also contain a large volume of both normal and attack traffic generated by various operating systems. Due to the stated reasons, the datasets present diverse and sophisticated attack approaches that are

highly suitable for testing the proposed NIDS framework. The pre-processing steps applied to the datasets are the same for all the conducted experiments. Primarily simple data cleaning is used for the datasets. The basic cleaning resolved the issues of, missing values, samples, duplicates, and infinite symbol records from the datasets. Then negative time samples in the datasets are also removed. In the datasets, CIC IDS 2017, and ISCX IDS 2012 the sample of labels “BENIGN” and “NORMAL” respectively are very high in quantity. To evade bias, samples of the “BENIGN” and “NORMAL” classes are reduced. In the CIC IDS 2017 dataset, two classes “Infiltration” and “Heartbleed” are removed as they had insufficient samples. The three classes representing web attacks in dataset CIC-IDS 2017 are merged into one class of “Web attack”. Further, SMOTE is applied with Edited Nearest Neighbors (ENN) to clean the training sets of each dataset. The SMOTEEN balances all the labels in the datasets. Table 2 represents the details of the datasets after pre-processing.

B. DATA TRANSFORMATION

Securing a network of diverse interconnected devices is a challenging task for ML-based NIDS. To optimize and facilitate ML-based NIDS, data normalization or transformation plays an integral part. The benchmarked datasets

TABLE 2. Details of datasets ISCX IDS 2012, CIC IDS 2017, and CSE CIC IDS 2018.

Dataset	CSE CIC IDS 2018	CIC IDS 2017	ISCX IDS 2012
Number of features	79	79	82
Number of classes	10	11	5
Number of samples	899,441	2,519,738	1,703,784
Number of each class	Benign:89,811 Bot:89,991 Brute Force-Web:89,942 DDoS HOIC attack:89,999 DDoS LOIC-UDP attack:90,000 DoS GoldenEye attack:89,851 DoS Slowloris attack:89,887 FTP-BruteForce: 90,000 SQL Injection: 89,963 SSH-Bruteforce:89,997	Benign:225,412 Bot:229,731 DDoS:229,186 DoS GoldenEye: 229,713 DoS Hulk: 228,473 DoS Slowhttptest: 229,136 DoS slowloris: 228,970 FTP-Patator: 229,823 PortScan: 229,939 SSH-Patator: 229,562 Web Attack: 229,793	BruteForceSSH: 392,492 DDoS: 337,141 HTTPDoS: 341,806 Infiltration: 319,241 Normal: 313,104

Algorithm 1 Statistical Model for Identifying the Suitable Transformation/Normalization/Scaling Method

- 1: **Output:** Suitable transformation/normalization/scaling approach for the dataset.
- 2: **Input:** Dataset, " d "
- 3: Where;
- 4: $d \triangleq (d_1, d_2, \dots, d_k), k(\in \mathbb{N})$: datasets.
- 5: i th data: $d_i \triangleq (f_1^i, f_2^i, \dots, f_n^i)$,
- 6: n = Total features.
- 7: N_m = m -th normalization.
- 8: Step 1: Pre-Process the dataset.
- 9: $d'' \leftarrow Pre - Processing(d)$
- 10: Step 2: Apply transformation/normalization/scaling on dataset, (i.e. N_m).
- 11: $d^{(m)} \leftarrow N_m(d'')$, where $d_i^{(m)} \triangleq (f_1^{(m),i}, \dots, f_n^{(m),i})$
- 12: Step 3: Compute Median, Mean and, Skewness of each feature.
- 13: $mean_j^{(m)} = mean(f_j^{(m),1}, \dots, f_j^{(m),k})$
- 14: $median_j^{(m)} = median(f_j^{(m),1}, \dots, f_j^{(m),k})$
- 15: $skewness_j^{(m)} = abs(skewness(f_j^{(m),1}, \dots, f_j^{(m),k}))$
- 16: Step 4: Calculate average Median, Mean and Skewness of the dataset.
- 17: $\overline{mean}^{(m)} = mean(mean_1^{(m)}, \dots, mean_n^{(m)})$
- 18: $\overline{median}^{(m)} = median(median_1^{(m)}, \dots, median_n^{(m)})$
- 19: $\overline{skewness}^{(m)} = skewness(skewness_1^{(m)}, \dots, skewness_n^{(m)})$
- 20: Step 5: Apply Rank(R) and percentile on $\overline{median}^{(m)}$, $\overline{mean}^{(m)}$, and $\overline{skewness}^{(m)}$ of each $N_m, \forall m$.
- 21: Step 6: Sum the R of $\overline{median}^{(m)}$, $\overline{mean}^{(m)}$, and, $\overline{skewness}^{(m)}$ to know the appropriate normalization (N_{m^*}).
- 22: $m^* = argmax_m \{R(\overline{mean}^{(m)}) + R(\overline{median}^{(m)}) + R(\overline{skewness}^{(m)})\}$

or real network traffic are not normally distributed and are skewed [37]. ML algorithms have a tendency to perform better when the data is normalized, as it tends to increase the general structure and relation among features [38]. However, identifying the most appropriate normalization or data transformation for the data or dataset is a dubious task. In this study, the statistical approach proposed in the paper [8] is implemented to recognize the most appropriate normalization for the three datasets. As the proposed statistical method is simple and efficient in terms of implementation and computational requirements. The only difference between Algorithm 1 in this study and the algorithm suggested in the paper [8] is the flow of feature selection. In this study, the algorithm is implemented before feature selection, while

in the research work [8] it was implemented after feature selection. Subsequently, normalizing the dataset presents a more prominent and suitable correlation between the features for a feature selection based on correlation. Algorithm 1 represents the flow in which the algorithm is implemented in this study.

For this study, five prominent normalization methods were implemented on all three datasets. The methods implemented are MinMax, Robust scaler, Standard Scaler, L2 standardization, and Yeo-Johnson. The MinMax [39] approach can mathematically be represented as (1).

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

TABLE 3. Average median, mean, and skewness of each dataset based on different normalization methods.

Normalization	CSE CIC IDS 2018			CIC IDS 2017			ISCX IDS 2012		
	Avg. Mean	Avg. Median	Avg. Skewness	Avg. Mean	Avg. Median	Avg. Skewness	Avg. Mean	Avg. Median	Avg. Skewness
Yeo-Johnson	-0.1948	1.0622	8.3986	0	-0.279	5.017	0	-0.338	2.031
L2 Normalization	0.0767	0.1008	9.0973	0.040	0.015	3.058	0.043	0.020	8.231
Robust Scaler	0.2128	0	26.6221	26581.63	0	15.235	54874	0	12.412
Standard Scaler	-1.0853	1.121	26.6221	0	-0.277	15.235	0	0.064	12.412
Min-Max	0.1071	0.1072	26.6221	0.193	0.155	9.986	-0.806	-0.939	12.412

Equation (2) represents the Robust scaler [40] method. Where, ‘x’ denote the values while $Q_1 = 25^{th}$ quantile and $Q_3 = 75^{th}$ quantile.

$$x = \frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \tag{2}$$

Mathematically the Standard scaler [39] can be represented as (3), where ‘s’ signifies the standard deviation and ‘μ’ indicate the mean.

$$x_{scaler} = \frac{x - \mu_{mean}}{s_{stddiv}} \tag{3}$$

Equation 4 represents the L2-standardization [41], where ‘x’ represents the attributes of dataset features.

$$\|x\|_2 = (|x_1|^2 + \dots + |x_n|^2)^{1/2} \tag{4}$$

Yeo-Johnson [38] can be denoted as (5). Where ‘j’ represents feature attributes, ‘λ’ can be a \mathbb{R} , and $\lambda = 1$ gives the identity conversion.

$$j_i^{(\lambda)} = \begin{cases} \left(\frac{(j_1 + 1)^\lambda - 1}{\lambda} \right) & \text{if } \lambda \neq 0, j \geq 0 \\ \log(j_i + 1) & \text{if } \lambda = 0, j \geq 0 \\ \left(\frac{-[(j_1 + 1)^2 - (\lambda) - 1]}{2 - \lambda} \right) & \text{if } \lambda \neq 2, j < 0 \\ -\log(-j_i + 1) & \text{if } \lambda = 2, j < 0 \end{cases} \tag{5}$$

Based on Algorithm 1, the median, mean, and skewness of each dataset are computed using (6), (7) and (8).

$$Mean = \frac{\sum_{i=1}^n x_i}{n} \tag{6}$$

$$Median = \begin{cases} x \left[\frac{n}{2} \right] & \text{if } n \text{ is even} \\ \frac{\left(x \left[\frac{n-1}{2} \right] + x \left[\frac{n+1}{2} \right] \right)}{2} & \text{if } n \text{ is odd} \end{cases} \tag{7}$$

$$Skewness = \frac{n \sum_{i=1}^n (x_i - \bar{x})^3}{(n - 1)(n - 2)s^3} \tag{8}$$

where ‘n’ in (6), (7), and (8) represent the number of attributes or values in the dataset, ‘x’ denotes the attribute or value in a dataset. Further in (8), ‘x̄’ and ‘s’ are the mean and standard deviation respectively. For the suggested statistical process, skewness is considered an absolute or positive value. The median, mean, and skewness of the datasets can be seen in Table 3.

After computing the matrices shown in Table 3, percentile ranking is applied to find the most appropriate normalization

method. The formula for ranking and percentile can be represented as (9) and (10).

$$Percentile = \frac{x}{N} \times 100 \tag{9}$$

$$Rank = \frac{Percentile}{100(n + 1)} \tag{10}$$

where ‘x’ is the number of values beneath the particular value. The ‘N’ signifies the total number of values, and ‘n’ highlights the number of values. Ranks are allotted based on descending order. Table 4 represents the ranking of each normalization approach based on the Rank and Percentile method.

Based on Table 4, it can be seen that the Yeo-Johnson transformation was able to attain the highest rank among all the normalization methods. Except for dataset CSE-CIC-IDS 2018, where L2 Normalization achieved the same rank as Yeo-Johnson. Later, based on our classification results it is highlighted that the Yeo-Johnson was able to achieve precision higher than L2 Normalization.

C. FEATURE SELECTION

In this age of big data, an immense amount of data is transferred every second. Such a high transaction rate makes real-time incursion detection a problematic task. ML which is the most suitable methodology for NIDS does tend to suffer from a low anomaly detection rate with high-dimensional data. Traditionally, features are selected after performing basic pre-processing. In this study, we experimented by using power transformation before filter-based FS, as normalizing data before applying a statistical-based FS can improve the probability of selecting relevant features. The feature selection flow is adopted as per the study [7] as represented in Figure 2.

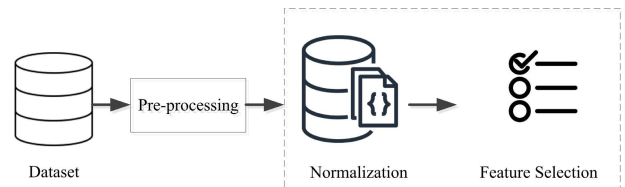


FIGURE 2. Feature selection flow for the proposed NIDS framework.

Based on the study in papers [7] and [11], Pearson correlation (PC) is implemented to select the features from the datasets. Equation (11) represents the mathematical

TABLE 4. After applying the Rank and Percentile approach to Table 3 and computing the Ranks for each normalization.

	CSE CIC IDS 2018				CIC IDS 2017			
	Avg. Mean	Avg. Median	Avg. Skewness		Avg. Mean	Avg. Median	Avg. Skewness	
Normalization	Rank	Rank	Rank	\sum Rank	Rank	Rank	Rank	\sum Rank
Standard Scaler	5	1	1	7	4	4	1	9
Robust Scaler	1	5	1	7	1	3	1	5
Yeo-Johnson	4	2	5	11	4	5	5	13
Min-Max	2	3	1	6	2	1	3	6
L2 Normalization	3	4	4	11	3	2	5	10

ISCX IDS 2012				
	Avg. Mean	Avg. Median	Avg. Skewness	
Normalization	Rank	Rank	Rank	\sum Rank
Yeo-Johnson	3	4	5	12
L2 Normalization	2	2	4	8
Robust Scaler	1	3	3	7
Standard Scaler	3	1	1	5
Min-Max	5	5	1	11

representation of PC.

$$P_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^m (y_i - \bar{y})^2}} \quad (11)$$

where:

P_{xy} = PC coefficient value,

x_i = Discrete sample values of each dependent attributes,

y_i = Discrete sample values of the individual attribute,

\bar{x} = Average of all sample values of each dependent attribute,

\bar{y} = Average of all sample values of the individual attribute.

m = Total attributes.

Table 5 represents the total number of features selected by the PC approach.

TABLE 5. Number of original features in the dataset features and features after PC feature selection.

	CSE CIC IDS 2018	CIC IDS 2017	ISCX IDS 2012
Total Features	79	79	82
Features Selected	72	61	41

D. CONVERTING DATASETS TO IMAGES

After feature selection, the datasets ISCX IDS 2012, CIC IDS 2017, and CSE CIC IDS 2018 are ready to be transformed into images. As figure 1 highlights, the transformation of non-image data into image format is based on two phases. Initially, the DeepInsight-based [42] approach is implemented. The Kernel Principal Component Analysis (KPCA) [43] is used to map the dataset features from a 1D space to a 2D space. Due to the mapping by KPCA, the dataset features are expected to be linearly discrete. The 2D space mapping represents features as points in the Cartesian plane. The plotted points only represent the position of features in 2D space and not the attributes of those features. To facilitate the CNN-based classifier, the convex hull algorithm is used to create a small rectangular shape. This rectangular shape

contains all the mapped features of the dataset. The next step is to transform the Cartesian coordinates into pixels. During the transformation of Cartesian coordinates to pixels, some of the features are averaged due to the limitation of pixels. The limitation of pixels is due to the size of the image. As with feature selection, the quantity of features is reduced resulting in a limited pixel representation of images. The newly generated frame of pixels represents the positions of the dataset features. The feature attributes are then mapped based on the frame of the pixel representing features. The overlapping pixels of features are averaged and assigned the same pixel location. After this process, each sample of the label in a dataset is converted into an image representing that sampled label. Once all the datasets are converted from non-image data to image format the Gabor filter [44], [45] is used to further improve the generated images.

1) GABOR FILTER

Gabor filter plays a significant role in modifying, extracting, improving, or representing digital graphical data. These filters have also shown remarkable localization properties in both frequency and spatial domains. The Gabor filters can be considered as special kinds of band-pass filters. Based on the configuration, they allow a particular band of frequencies to pass while stopping the others. The parameter settings for the Gabor filter depend on the task at hand. To implement the Gabor filter, two types of parameters are configured. First, the parameters that define how the Gabor filter will be. Second, which features will the Gabor filter react to. The parameters used for the Gabor filter in the proposed framework can be seen in Figure 3.

A two-dimensional Gabor filter can be considered as a sinusoidal signal of a particular frequency and direction, regulated by a Gaussian wave. To represent the orthogonal direction, the Gabor filter has both imaginary and real components. The complex, real and imaginary equations of the Gabor filter can be represented as Equations (12), (13), and (14) respectively. Both the real and imaginary components can be used separately or can be shaped into a complex

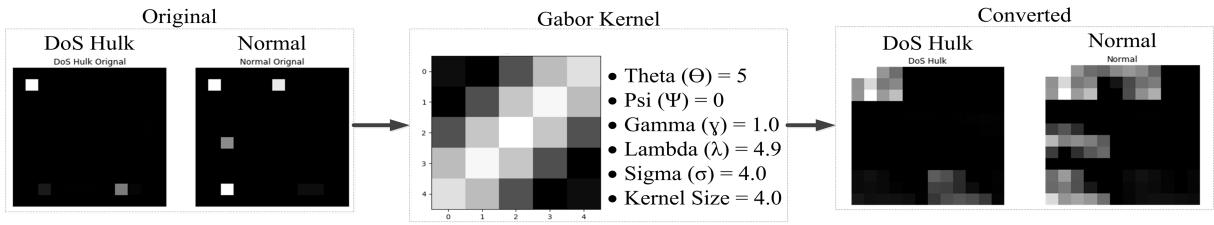


FIGURE 3. Gabor filter parameters with sample images of DoS Hulk and Normal label generated by defined Gabor filter.

number component.

$$G(x, y; \lambda, \theta, \phi, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \times \exp\left(i\left(2\pi\frac{x'}{\lambda} + \phi\right)\right) \quad (12)$$

$$\Re\{G(x, y; \lambda, \theta, \phi, \gamma)\} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \times \cos\left(2\pi\frac{x'}{\lambda} + \phi\right) \quad (13)$$

$$\Im\{G(x, y; \lambda, \theta, \phi, \gamma)\} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \times \sin\left(2\pi\frac{x'}{\lambda} + \phi\right) \quad (14)$$

where:

$$x' = x \cos \theta + y \sin \theta,$$

$$y' = -x \sin \theta + y \cos \theta.$$

λ = Wavelength of the sinusoidal part,

θ = Controls the positioning of the Gabor function,

γ = Spatial point ratio,

σ = The standard deviation(σ) of the Gaussian covering,

ϕ = The phase offset/error of the sinusoidal function.

The parameters $\lambda, \theta, \gamma, \sigma,$ and ϕ define the form of the Gabor function.

After the transformation of images with the Gabor filter, the process of converting non-image data into image format is completed. Figure 4 illustrates an overview of the proposed procedure for converting non-image data into image format.

E. CNN-MODEL

The final block of the proposed framework is the CNN-based classifier. The CNN-based classifier is implemented due to its potential to achieve high accuracy and computational efficiency. It is also among the most prominent classifier in recent research publications. Implementing a CNN-based classifier also provides ground for comparing the proposed framework with recent prominent methods. The sequential CNN model implemented for the experiments consists of 12 layers. The layers consist of an input layer, three conv2D layers, four dropout layers, flatten layer, and three dense layers including an output layer. The kernel size for each convolutional layer is three. The convolutional layers and the dense layers used Relu as the activation function. Whereas, the output layer

TABLE 6. Summary of the CNN-model parameters.

Parameter	Details
Model	Sequential
Layers	Input,Conv2D,Dropout,Flatten,Dense,Output
Kernel size	3 × 3
Activation functions	Relu, Softmax
Dropout	0.2
Optimizer	Adam
Learning rate	0.001
Loss function	sparse categorical cross-entropy

TABLE 7. Implemented image processing-based NIDS for comparison.

Reference	Summary of Image Transformation Method
[14]	Used the Fast Fourier Transformation (FFT) to convert non-image datasets to image format for a CNN-based NIDS.
[15]	Implemented the short-time Fourier transform (STFT) with spectrogram to generate spectrogram-based images from non-image dataset for a CNN-based NIDS.
[16]	Converting the non-image dataset to a 2D-Gray scale image for a CNN-based NIDS
[17]	Transforming non-image dataset into images based on Deepnsight and Gabor filter for CNN-based NIDS
[42]	The DeepInsight methodology was used to turn non-image data into images for a CNN-based NIDS

used the softmax as an activation function. A dropout of 0.2 is used for the dropout layers. For training, an Adam optimizer with a 0.001 learning rate is implemented. The sparse categorical cross-entropy is used as a loss function. The CNN model is implemented with the help of Keras (python library). Table 6 represents the summary of the parameter settings for the CNN model.

IV. IMPLEMENTATION

The implementation of the proposed framework is on python (v 3.6) programming language with GPU-enabled TensorFlow (v 2.3.1) on the Keras framework is used. The DeepInsight tool based on python is publicly accessible [42]. The tool was downloaded and fused with the proposed framework. The Gabor kernel is created using the cv2 library.

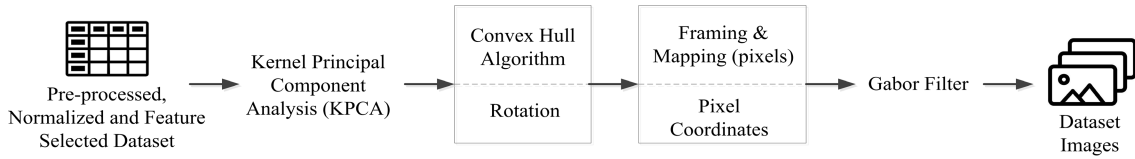


FIGURE 4. Illustration of converting non-image dataset into image format based on the proposed framework.

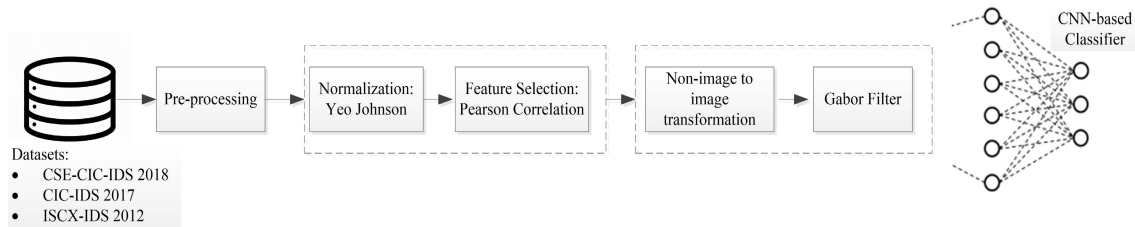


FIGURE 5. Implementation flow and components of the proposed framework.

TABLE 8. Assessment of the proposed method and comparative approaches of NIDS-based on image processing.

Matrix	CSE-CIC-IDS 2018						CIC-IDS 2017 Dataset					
	Proposed	[11]	[12]	[13]	[14]	[39]	Proposed	[11]	[12]	[13]	[14]	[39]
Accuracy	97.75%	97.12%	97.80%	97.09%	98.18%	96.79%	98.79%	97.65%	98.18%	98.49%	98.70%	89.59%
F1-Score	97.74%	97.12%	97.74%	97.07%	98.18%	96.78%	98.77%	97.62%	98.14%	98.48%	98.69%	89.66%
Precision	97.97%	97.19%	97.88%	97.08%	98.22%	96.73%	98.80%	97.67%	98.15%	98.49%	98.70%	90.93%
Recall	97.76%	97.13%	97.72%	97.08%	98.18%	96.78%	98.77%	97.63%	98.16%	98.48%	98.69%	89.57%
Kappa	97.50%	96.80%	97.56%	96.77%	97.98%	96.43%	98.67%	97.41%	98.01%	98.25%	98.57%	88.55%
Num. of features	72	79	79	79	79	79	61	79	79	79	79	79

Matrix	ISCX-IDS 2012 Dataset					
	Proposed	[11]	[12]	[13]	[14]	[39]
Accuracy	92.92%	88.85%	91.50%	92.09%	92.50%	78.78%
F1-Score	92.61%	88.30%	91.30%	91.79%	92.20%	75.56%
Precision	92.89%	88.54%	91.47%	91.95%	92.22%	82.99%
Recall	92.65%	88.36%	90.91%	91.91%	92.23%	77.91%
Kappa	91.54%	86.03%	90.46%	90.10%	91.61%	73.40%
Num. of features	41	82	82	82	82	82

Then filter2D method is convolved with the Gabor filter to extract the specific patterns from the images. To highlight the general application of the proposed framework three different NIDS benchmark datasets are implemented. After converting the NIDS datasets to images, each image dataset is classified using the CNN classifier. To estimate the efficiency of the CNN classification precision, accuracy, F1-score, recall, Cohen’s kappa coefficient, and receiver operating characteristics (ROC) are measured as performance assessment metrics. The classification precision, accuracy, F1-score, recall, and kappa coefficient are computed using Equations (15) to (19).

$$Accuracy = \frac{TP + TN}{Total} \tag{15}$$

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{18}$$

$$Kappa(\kappa) = \frac{p_0 - p_e}{1 - p_e} \tag{19}$$

The accuracy represents the correlation of correctly predicted events to the total number of events. Precision can be defined as the percentage of properly classified attacks on all the samples classified as attacks. The recall represents the ratio of all the appropriately predicted attack samples to all the actual attack samples. The F1 score is kind of an average between precision and recall. An F1 score is used to examine the correctness of a classification model. The TN(True Negative) and TP(True Positive) are the appropriately classified attack and normal events respectively. Whereas, FP(False Positive) and FN(False Negative) are incorrectly classified events as normal and attack, respectively. The ROC curve is a visual depiction of the classification model at all prediction edges. In equation (19), ‘ p_0 ’ is the general precision of the ML model, and ‘ p_e ’ signifies the balance between the ML model estimates and the true class or label values as if occurring by coincidence. The CNN- model is trained for 100 epochs with an 80/20 ratio of train and test datasets respectively. Figure 5, highlights the flow including the components of

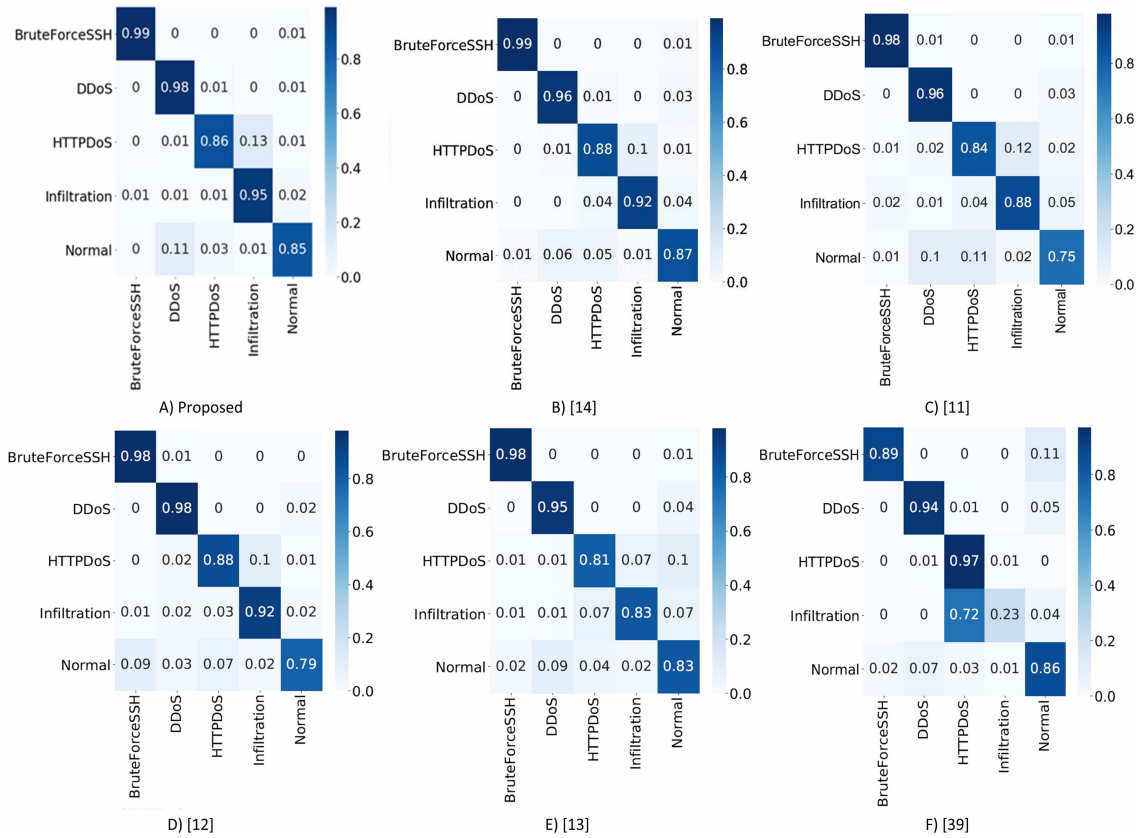


FIGURE 6. The confusion matrices of the proposed and comparative methods based on ISCX-IDS 2012 dataset.

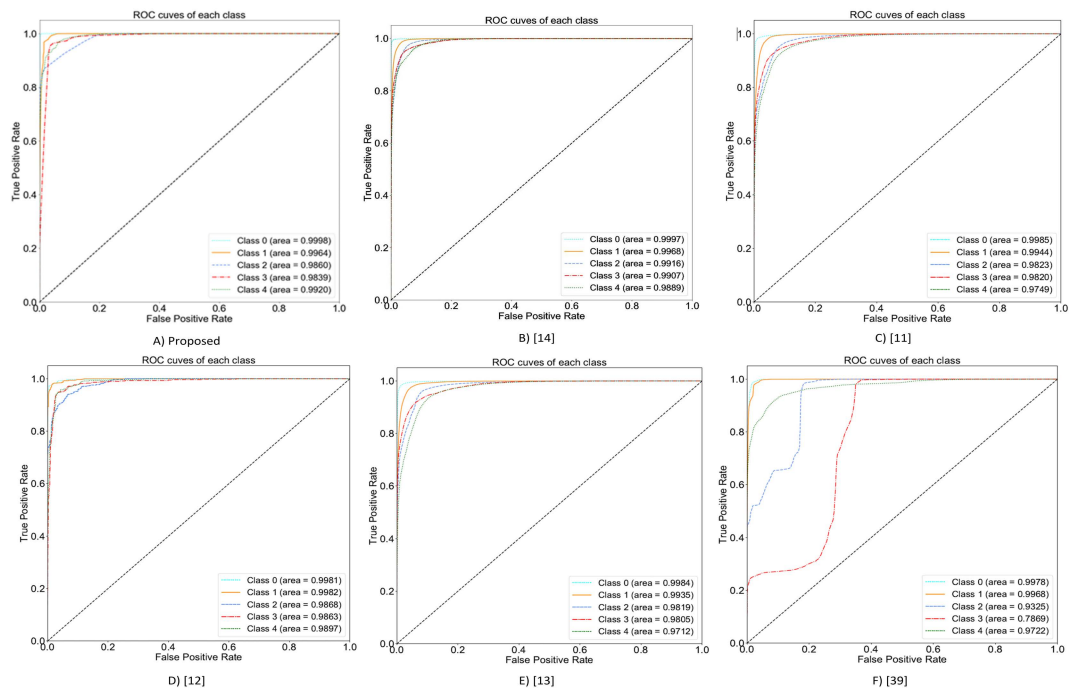


FIGURE 7. The receiver operating characteristic curves of the suggested and comparative approaches on the ISCX-IDS 2012 dataset.

the proposed NIDS framework. The components highlighted are the datasets used for the experimentation, normalization

approach, feature selection method, non-image to image conversion, and image enhancement procedure.

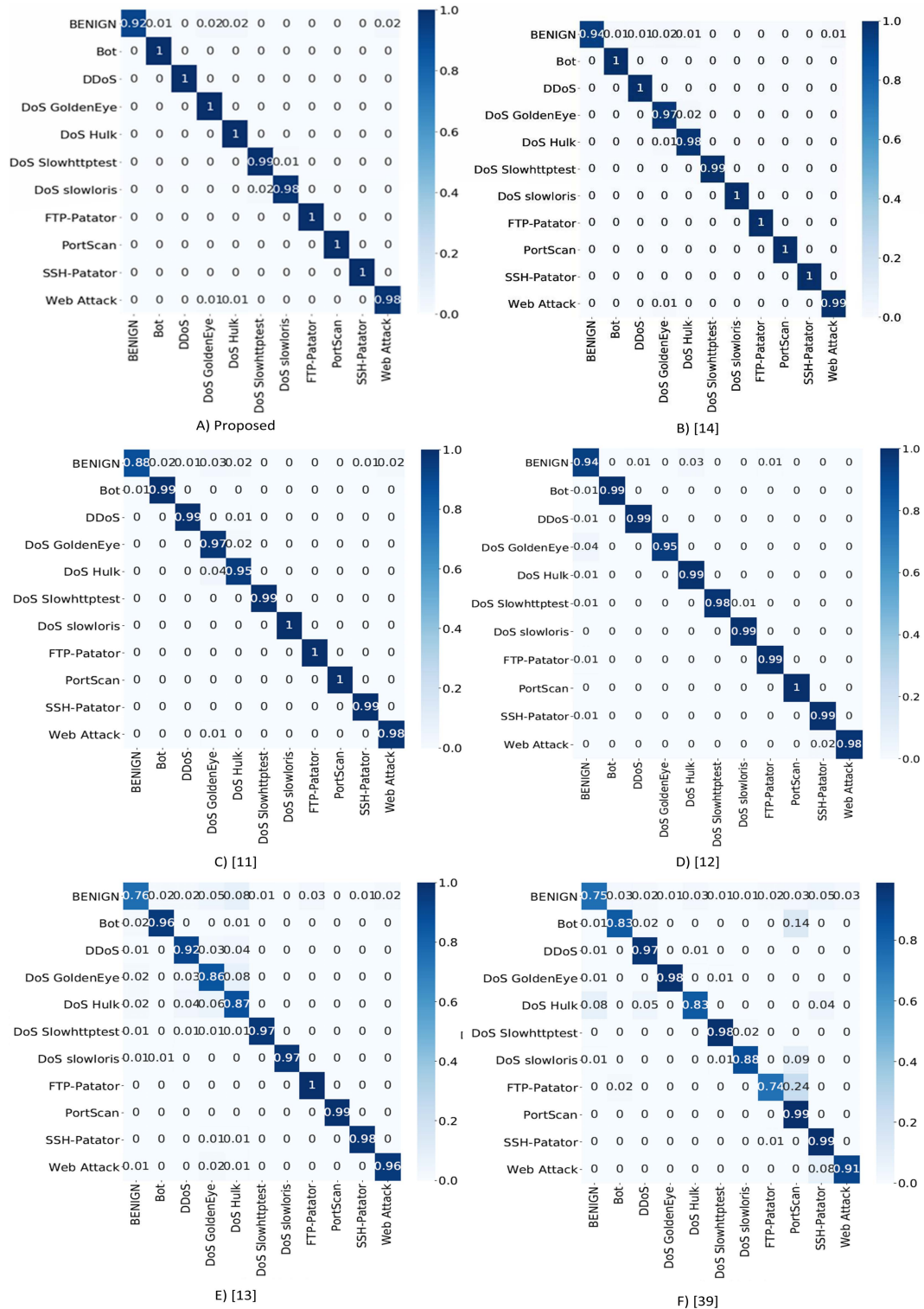


FIGURE 8. The confusion matrices of the proposed and comparative approaches on the CIC IDS 2017 dataset.

V. RESULTS AND COMPARISON

To highlight the capability of the suggested framework it is compared with some of the recent notable approaches. The five comparative NIDS approaches implemented for comparison are shown in Table 7. Table 7 also presents a summary of

the method adopted by the study to transform the non-image data into an image format.

To provide comparable grounds for evaluation, the proposed framework and comparison methods are implemented with the same parameter settings. Such as the datasets after

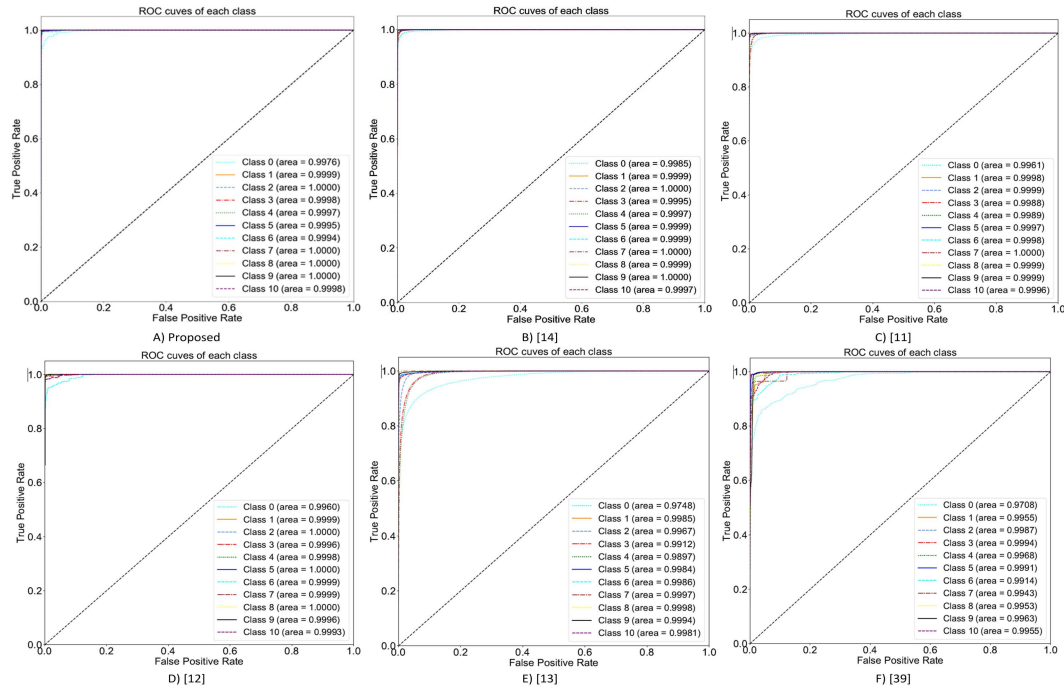


FIGURE 9. The receiver operating characteristic curves of the comparative and proposed approach on CIC IDS 2017 dataset.

pre-processing, and the CNN model for classification. While the approach for converting the non-image datasets to image format was based on the method described in the published work. The first comparative approach used FFT to create images from non-image data. The FFT is an optimized and fast algorithm of the discrete Fourier transformation (DFT). The DFT can be represented as the equation(20). To generate the FFT-based images, 5184 sampling points were taken as per the process explained in the research paper.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} \quad k \in \{0, 1, \dots, N - 1\} \quad (20)$$

where $X(N)$ is the signal sampling in the time domain. To implement the STFT-based spectrogram images, the STFT of a discrete-time signal $x[n]$ can be represented as the equation (21).

$$x[n] = X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w_{hm}[n - m]e^{-j\omega n} \quad (21)$$

The $x[n] = (f_1, \dots, f_{n-1})$ represents the input data vector with ‘ f ’ as features of the dataset. While ‘ m ’ represents the time and ‘ ω ’ represents the angular frequency. The mathematical representation of the Hanging window function ($w_{hm}[n]$) can be seen as the equation (22).

$$w_{hm}[n] = \frac{1}{2} \left(1 - \cos\left(2\pi \frac{n}{N}\right) \right) \quad 0 \leq n \leq N - 1 \quad (22)$$

where ‘ N ’ presents the length of observation time. The final step of generating the spectrogram images is based on the

TABLE 9. Time (sec) consumed by proposed framework and comparative algorithms for converting non-image data to image format.

Dataset	Proposed	[11]	[12]	[13]	[14]	[39]
CSE CIC IDS 2018	9.2	110.3	34.9	5.3	9.5	9.4
CIC IDS 2017	25.5	324.9	36.2	12.5	25.9	25.1
ISCX IDS 2012	16	1727.2	209.4	8.4	16.8	16.3

equation (23).

$$Spectrogram(m, \omega) = |STFT x[n]|^2 = |X(m, \omega)|^2 \quad (23)$$

With the help of equations (21) to (23), the datasets were converted into spectrogram-based image datasets. The paper that implemented 2D-gray scale images, presented two different methods of generating images from non-image datasets. Method one presented an approach to generate a 3-channel RGB (Red, Green, Blue) image. While method two presented a 1 channel 2D gray-scale image. Both methods follow the same process to generate the initial image for RGB and grey-scale conversion. After the initial pre-processing, the features of the dataset are re-scaled between the values of 0 to 255. Then 2D images of 13*9 and 13*6 pixels are generated for the CSE-CIC-IDS 2018 and NSL-KDD datasets respectively. For comparison purposes, the 2D gray-scale images of datasets were generated based on the process defined by the paper. The fourth competitor is our earlier work, which followed the same approach as in this paper. Except for the augmented

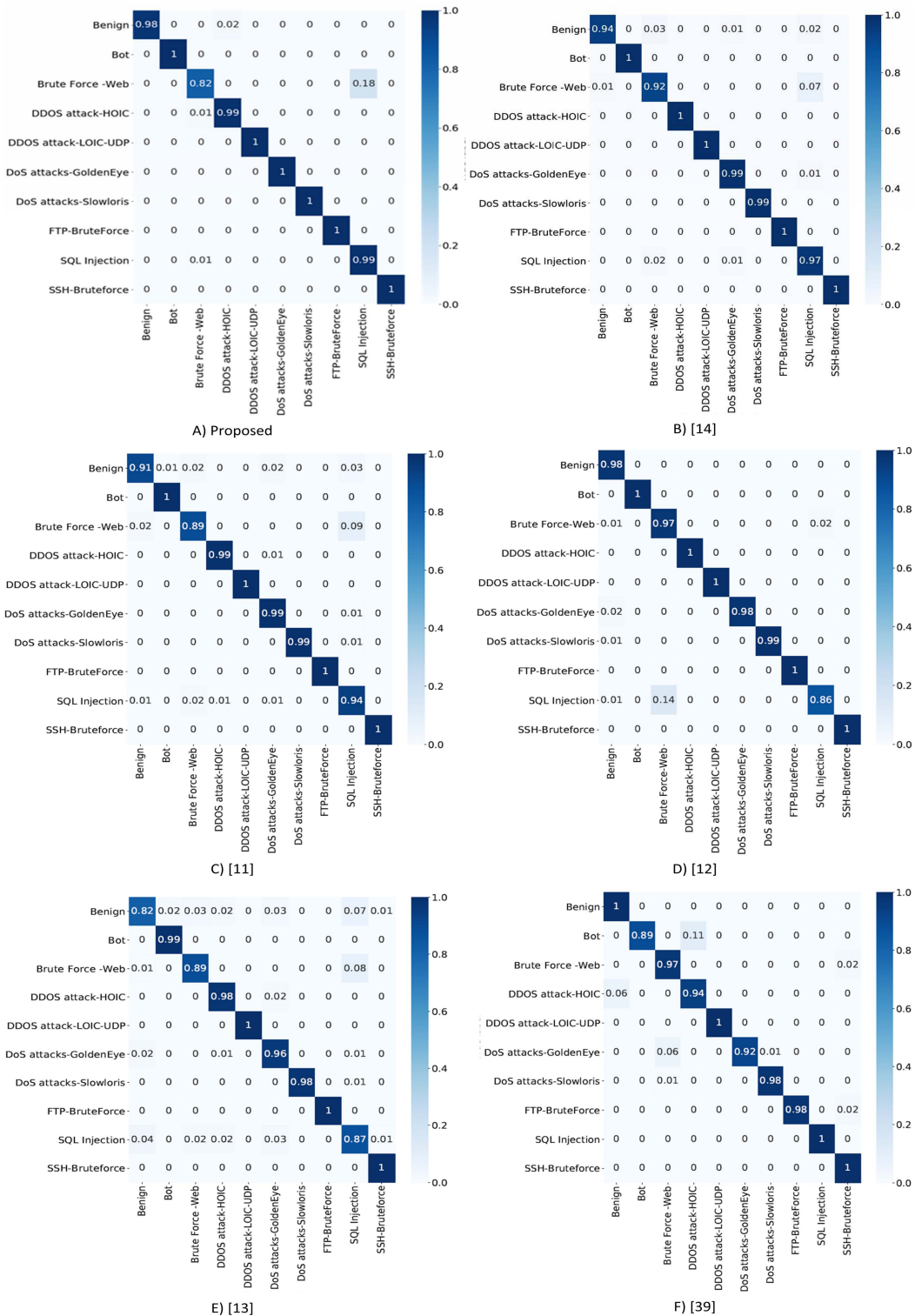


FIGURE 10. Confusion matrix of the proposed and comparative approaches based on CSE-CIC-IDS 2018 dataset.

feature selection adopted in this study. The fifth approach implemented is based on the DeepInsight methodology to create images from non-image data. This implementation highlights the image classification results without the fusion

of the Gabor filter. As compared to the relative approaches, the proposed framework in this paper is implemented on four different datasets. While the comparative work is implemented on one of two datasets. This highlights the fact that the

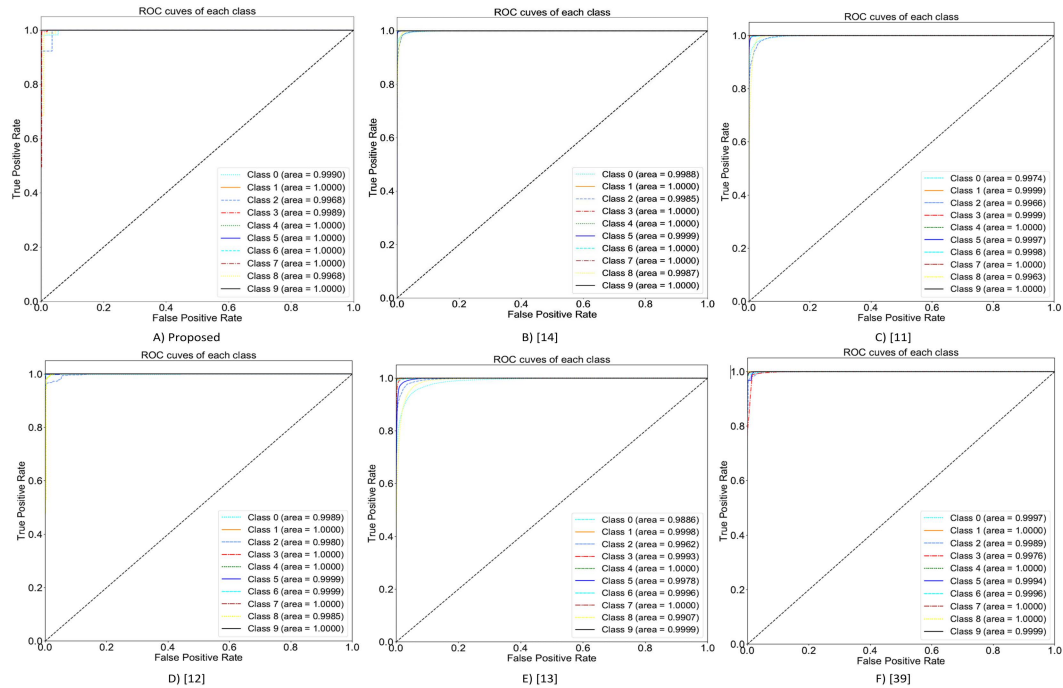


FIGURE 11. ROC of the proposed and comparative approaches based on CSE-CIC-IDS 2018 dataset.

proposed framework is generally applicable and can achieve high precision results. Table 8 represents the results of the proposed framework in contrast with the implemented comparative approaches.

The confusion matrix and ROC of the proposed and comparative approaches for the dataset ISCX-IDS 2012 can be seen in Figures 6 and 7 correspondingly. In Figure 7 (i.e ROC), Class 0 represents BruteForceSSH, Class 1 represents DDoS and similarly, the remaining Class labels in ROC are in sequence with the confusion matrix labels (i.e Figure 5).

Figures 8 and 9 represent the respective confusion matrix and ROC of the proposed and comparative approaches for the dataset CIC-IDS 2017. The ‘Class’ labels in the ROC (i.e. Figure 9) are in the same sequence as in Figure 8. That is Class 0 represents BENIGN, Class 1 represents Bot, and onwards.

The confusion matrix and ROC of the proposed and comparative approaches for the dataset CSE-CIC-IDS 2018 can be seen in Figures 10 and 11 respectively. As mentioned earlier the ‘Class’ labels in the ROC (i.e. Figure 11) are in the same sequence as in Figure 10. That is Class 0 represents Benign, Class 1 represents Bot, and onwards.

As the focus of this study is to achieve an optimized NIDS framework. A comparison of the time consumed by each competitive approach and the proposed framework is also computed. The time contrast is only focused on the time consumed by each method in transforming non-image data into image format. As the rest of the steps by each comparative methodology is the same. The python function ‘time’ [46] is

used for time computation. Table 9 shows the time used by each implemented approach.

Understandably a time-based evaluation may not be a standard approach to signify the efficiency of the proposed framework. For instance, factors like hardware can influence the time dynamic of implemented methodology. However, for this study, all the approaches are implemented in the same environment. Therefore the time-based comparison can provide a rough intuition for the efficiency of the proposed and compared frameworks.

VI. DISCUSSION

Based on the results highlighted in the earlier section, it can be seen that the proposed NIDS framework was able to achieve competitive results. In this section, the results achieved by each dataset are discussed separately. Starting with the results of the dataset CSE CIC IDS 2018. The proposed framework was able to achieve a precision of almost 98% on the dataset with only 72 features. In contrast to our earlier work [17], which achieved a slightly higher precision on the same dataset but with 79 features. While the other competitors were not able to achieve a precision higher than the suggested framework. Despite using all the 79 features of the dataset. The results of the recommended framework on the CIC-IDS 2017 dataset are the highest among the comparative approaches. Even though the proposed method used only 61 features as compared to the 79 features used by all the comparative approaches. The dataset ISCX-IDS 2012 attained the highest precision as compared to the implemented methodologies. The ISCX-IDS 2012 dataset was able

to achieve the highest result with only 41 features. Whereas the competitive methods used 82 features of the dataset. As discussed earlier, in the era of big data a reduced number of features can play a vital role in optimizing an ML-based NIDS. The core purpose of this study was to attain an optimized framework for image processing-based NIDS. The implementation results highlight that the suggested system can play a significant role in optimizing image processing-based NIDS.

VII. CONCLUSION

The NIDS is among the most fundamental part of providing network security. NIDS based on ML and DL is considered highly effective against illusive attacks on the network. DL algorithms are considered highly efficient in understanding the patterns of normal and ab-normal behaviors on a network. Due to the advancements in the field of image processing, security experts are exploring the possibilities of building efficient NIDS based on image processing. In this study, a new framework for NIDS based on image processing is presented. The proposed framework follows a three-tier approach to generate a refined and improved representation of the non-image-based NIDS dataset. The framework reduces the number of features to achieve low computational with high precision. The feature selection process also normalizes the data for better interpretation of features for DL-based models. Although in image processing larger image means higher precision. However, the proposed framework reduces the number of features and employs a fusion of DeepInsight with the Gabor filter to generate highly representative images of the non-image-based dataset. Such representation can assist a CNN in understanding deep and useful patterns from the images. To evaluate the efficiency and general application of the recommended framework, three different network intrusion detection datasets were implemented. The proposed framework achieved high accuracy on the implemented datasets. For future work, it is planned to explore methods that can assist in identifying appropriate parameters for implementing the Gabor filter on network flow. Identifying such an approach can avoid the need to implement a bank of Gabor filters on the non-image datasets. Further, we plan to evaluate the potential of the proposed framework with a variety of other ML-based classifiers and inspect methods that can identify attacks in live network traffic.

REFERENCES

- [1] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Proc. Comput. Sci.*, vol. 185, pp. 239–247, 2021, doi: [10.1016/j.procs.2021.05.025](https://doi.org/10.1016/j.procs.2021.05.025).
- [2] M. A. Siddiqi and N. Ghani, "Critical analysis on advanced persistent threats," *Int. J. Comput. Appl.*, vol. 141, no. 13, pp. 46–50, 2016.
- [3] L. Irwin. (Jan. 20, 2022). Data Breaches and Cyber Attacks in 2021: 5.1 Billion Breached Records. IT Governance. Accessed: May 6, 2022. [Online]. Available: <https://www.itgovernance.co.uk/blog/data-breaches-and-cyber-attacks-in-2021-5-1-billion-breached-records>
- [4] FBI. (2021). *2021 Internet Crime Report*. Accessed: May 5, 2022. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf
- [5] M. A. Siddiqi, W. Pak, and M. A. Siddiqi, "A study on the psychology of social engineering-based cyberattacks and existing countermeasures," *Appl. Sci.*, vol. 12, no. 12, p. 6042, Jun. 2022.
- [6] V. Mohanraj, R. Sakthivel, A. Paul, and S. Rho, "High performance GCM architecture for the security of high speed network," *Int. J. Parallel Program.*, vol. 46, no. 5, pp. 904–922, 2018.
- [7] M. A. Siddiqi and W. Pak, "Optimizing filter-based feature selection method flow for intrusion detection system," *Electronics*, vol. 9, no. 12, p. 2114, Dec. 2020.
- [8] M. A. Siddiqi and W. Pak, "An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection," *IEEE Access*, vol. 9, pp. 137494–137513, 2021.
- [9] S. Shyla, V. Bhatnagar, V. Bali, and S. Bali, "Optimization of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm," *Electronics*, vol. 11, no. 4, p. 507, Feb. 2022.
- [10] H. Alyami, M. T. J. Ansari, A. Alharbi, W. Alosaimi, M. Alshammari, D. Pandey, A. Agrawal, R. Kumar, and R. A. Khan, "Effectiveness evaluation of different IDSs using integrated fuzzy MCDM model," *Electronics*, vol. 11, no. 6, p. 859, Mar. 2022.
- [11] M. A. Siddiqi and W. Pak, "Efficient filter based feature selection flow for intrusion detection system," in *Proc. Int. Workshop Emerg. (ICT) Gyeongsan*, South Korea, 2020, pp. 1–13.
- [12] M. M. Rathore, A. Ahmad, and A. Paul, "Real time intrusion detection system for ultra-high-speed big data environments," *J. Supercomput.*, vol. 72, no. 9, pp. 3489–3510, Sep. 2016.
- [13] J. Toldinas, A. Venckauskas, R. Damaševicius, Š. Grigaliunas, N. Morkevicius, and E. Baranauskas, "A novel approach for network intrusion detection using multistage deep learning image recognition," *Electronics*, vol. 10, no. 15, p. 1854, Aug. 2021.
- [14] W. Liu, X. Liu, X. Di, and H. Qi, "A novel network intrusion detection algorithm based on fast Fourier transformation," in *Proc. 1st Int. Conf. Ind. Artif. Intell. (IAI)*, Shenyang, China, Jul. 2019, pp. 1–6.
- [15] A. S. Khan, Z. Ahmad, J. Abdullah, and F. Ahmad, "A spectrogram image-based network anomaly detection system using deep convolutional neural network," *IEEE Access*, vol. 9, pp. 87079–87093, 2021.
- [16] J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, p. 916, Jun. 2020.
- [17] M. A. Siddiqi and W. Pak, "An optimized and hybrid framework for image processing based network intrusion detection system," *Comput., Mater. Continua*, vol. 73, no. 2, pp. 3921–3949, 2022.
- [18] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, Q.-V. Pham, and N.-N. Dao, "A LSTM-FCNN based multi-class intrusion detection using scalable framework," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107720, doi: [10.1016/j.compeleceng.2022.107720](https://doi.org/10.1016/j.compeleceng.2022.107720).
- [19] J. Brownlee. (Apr. 10, 2020). How Do Convolutional Layers Work in Deep Learning Neural Networks?. Machine Learning Mastery. Accessed: May 9, 2022. [Online]. Available: <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
- [20] A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review," *Remote Sens.*, vol. 13, no. 13, p. 2450, Jun. 2021, doi: [10.3390/rs13132450](https://doi.org/10.3390/rs13132450).
- [21] M. A. R. Ahad, S. Kobashi, and J. M. R. S. Tavares, "Advancements of image processing and vision in healthcare," *J. Healthcare Eng.*, vol. 2018, pp. 1–3, Jan. 2018, doi: [10.1155/2018/8458024](https://doi.org/10.1155/2018/8458024).
- [22] J. Seo and H. Park, "Object recognition in very low resolution images using deep collaborative learning," *IEEE Access*, vol. 7, pp. 134071–134082, 2019.
- [23] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Funchal, Madeira, Portugal, 2018, pp. 108–116.
- [24] D. Kurniabudi, D. Stiawan, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.
- [25] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012.
- [26] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021, doi: [10.1002/ett.4150](https://doi.org/10.1002/ett.4150).

- [27] Y. Xu, X. Zhang, C. Lu, Z. Qiu, C. Bi, Y. Lai, J. Qiu, and H. Zhang, "Network threat detection based on group CNN for privacy protection," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–18, Sep. 2021, doi: [10.1155/2021/3697536](https://doi.org/10.1155/2021/3697536).
- [28] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.
- [29] G. Liu and J. Zhang, "CNID: Research of network intrusion detection based on convolutional neural network," *Discrete Dyn. Nature Soc.*, vol. 2020, pp. 1–11, May 2020, doi: [10.1155/2020/4705982](https://doi.org/10.1155/2020/4705982).
- [30] J. Man and G. Sun, "A residual learning-based network intrusion detection system," *Secur. Commun. Netw.*, vol. 2021, pp. 1–9, Mar. 2021, doi: [10.1155/2021/5593435](https://doi.org/10.1155/2021/5593435).
- [31] L. Yang and A. Shami, "A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2022, pp. 1–6.
- [32] W. Nadr. (Jan. 18, 2019). Why Feature Correlation Matters. A Lot! Towards Data Science. Accessed: May 17, 2022. [Online]. Available: <https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4>
- [33] E. Rajaby and S. M. Sayedi, "A structured review of sparse fast Fourier transform algorithms," *Digit. Signal Process.*, vol. 123, Apr. 2022, Art. no. 103403, doi: [10.1016/j.dsp.2022.103403](https://doi.org/10.1016/j.dsp.2022.103403).
- [34] X. Zhang, J. Chen, and Y. Zhou, "A multiple-layer representation learning model for network-based attack detection," *IEEE Access*, vol. 7, pp. 91992–92008, 2019.
- [35] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2022.
- [36] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cyber-security*, vol. 2, no. 1, p. 20, Dec. 2019, doi: [10.1186/s42400-019-0038-7](https://doi.org/10.1186/s42400-019-0038-7).
- [37] J. Raymaekers and P. J. Rousseeuw, "Transforming variables to central normality," 2020, *arXiv:2005.07946*.
- [38] D. Suprina. (Jan. 7, 2013). The Importance of Data Normalization in IPS. Help Net Security. Accessed: Mar. 3, 2022. [Online]. Available: <https://www.helpnetsecurity.com/2013/01/07/the-importance-of-data-normalization-in-ips/>
- [39] J. Brownlee. (Jun. 10, 2020). *How to Use StandardScaler MinMaxScaler Transforms Python*. Accessed: Apr. 28, 2022. [Online]. Available: <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>
- [40] S. A. Sohail. (Jul. 31, 2021). Feature Scaling in Machine Learning: Robust Scaler and MinMax Scaler With K-Means Clustering—A Python Tutorial. Medium. Accessed: May 7, 2022. [Online]. Available: <https://medium.com/@syedar.sohail/outlier-handling-using-robust-scaler-a-python-tutorial-613d174b58eb>
- [41] R. Karim. (Dec. 27, 2018). *Intuitions L1 L2 Regularisation*. Accessed: Apr. 9, 2021. [Online]. Available: <https://towardsdatascience.com/intuitions-on-l1-and-l2-regularisation-235f2db4c261#f810>
- [42] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, doi: [10.1038/s41598-019-47765-6](https://doi.org/10.1038/s41598-019-47765-6).
- [43] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," 2012, *arXiv:1207.3538*.
- [44] S. Eksi. (Apr. 5, 2021). Gabor Filter. Kaggle. Accessed: Feb. 5, 2022. [Online]. Available: <https://www.kaggle.com/sameteki/gabor-filter>
- [45] A. Shah. (Jun. 17, 2018). Through The Eyes Gabor Filter. Medium. Accessed: Jan. 18, 2022. [Online]. Available: https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97
- [46] G. A. Hjelle. (Mar. 21, 2022). *Python Timer Functions: Three Ways to Monitor Your Code*. Real Python. Accessed: Jun. 23, 2022. [Online]. Available: <https://realpython.com/python-timer/>



MURTAZA AHMED SIDDIQI received the B.S. degree from Greenwich University, Pakistan, the M.S. degree from Mohammad Ali Jinnah University, Pakistan, and the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, Republic of Korea. He is currently working as a Postdoctoral Researcher at Yeungnam University. His research interests include network and system security, network intrusion detection based on machine learning, and data analysis.



WOOGUIL PAK received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in electrical engineering and computer science from Seoul National University, in 1999, 2001, and 2009, respectively. In 2010, he joined the Jangwee Research Institute for National Defence, as a Research Professor, and Keimyung University, Daegu, South Korea, in 2013. Since 2019, he has been an Associate Professor with Yeungnam University, Gyeongsan, South Korea. His current research interests include network and system security, blockchain, and real-time network intrusion prevention based on machine learning for over 1Tbps networks.

• • •