

# An Optimized and Hybrid Framework for Image Processing Based Network Intrusion Detection System

Murtaza Ahmed Siddiqi and Wooguil Pak\*

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

\*Corresponding Author: Wooguil Pak. Email: wooguilpak@yu.ac.kr

Received: 06 March 2022; Accepted: 07 May 2022

**Abstract:** The network infrastructure has evolved rapidly due to the ever-increasing volume of users and data. The massive number of online devices and users has forced the network to transform and facilitate the operational necessities of consumers. Among these necessities, network security is of prime significance. Network intrusion detection systems (NIDS) are among the most suitable approaches to detect anomalies and assaults on a network. However, keeping up with the network security requirements is quite challenging due to the constant mutation in attack patterns by the intruders. This paper presents an effective and prevalent framework for NIDS by merging image processing with convolution neural networks (CNN). The proposed framework first converts non-image data from network traffic into images and then further enhances those images by using the Gabor filter. The images are then classified using a CNN classifier. To assess the efficacy of the recommended method, four benchmark datasets i.e., CSE-CIC-IDS2018, CIC-IDS-2017, ISCX-IDS 2012, and NSL-KDD were used. The proposed approach showed higher precision in contrast with the recent work on the mentioned datasets. Further, the proposed method is compared with the recent well-known image processing methods for NIDS.

**Keywords:** Anomaly detection; convolution neural networks; deep learning; image processing; intrusion detection; network intrusion detection

## 1 Introduction

As the internet becomes an integral part of life, online security and privacy have grown into a paramount concern for the consumer. According to CISCO, the internet will have 5.3 billion users by 2023 [1]. To fulfill this increasing demand for the internet, the networks are constantly expanding. With this expansion of network scale and scope, network security is becoming a major concern [2–4]. Despite the recent progress in the field of network security, the ratio of successful network attacks is reasonably high [5]. Such numbers of successful attacks, emphasize the need for an improved solution for network security. In this situation, the network Intrusion detection systems (NIDS) based on machine learning (ML) and deep learning (DL) are considered highly effective against malicious or invasive activity



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

on the network [6–8]. The reason behind the efficiency of ML and DL algorithms is their ability to learn and identify the patterns of both normal and malicious network traffic. However, the constant mutation in attack patterns is challenging even for ML and DL-based approaches. To enrich both ML and DL-based algorithms, researchers have looked into methods such as data pre-processing steps, feature engineering, the fusion of classifiers, etc. [9,10]. Practically, DL-based approaches are computationally expensive but highly accurate in NIDS. With the help of graphical processing units (GPUs), DL-based algorithms are showing promising results in terms of complexity and efficiency in several applications [11]. Among the DL algorithms, convolutional neural networks (CNN) are considered as one of the most precise and efficient algorithms for detection and recognition tasks [12]. Particularly in the area of computer vision [13–17] (i.e., object detection, pattern detection, image classification, etc.), the applications of the CNN-based models are extremely effective [18]. Due to the sensitive application of CNN, researchers are always looking to improve the detection ability of CNN models [19]. This constant improvement in CNN modeling has urged researchers to explore CNN for extracting and identifying critical patterns within the network traffic. Based on some of the recent research publications, CNN has displayed remarkable prediction ability [20–22]. To utilize CNN for NIDS, conversion of non-image network traffic into image format plays a significant part. Most of the existing published work converts non-image network traffic into images by using 1D, 2D, or 3D (dimension) matrix/arrays [23]. Whereas, some researchers enhance the form of these converted images with the help of different algorithms. Such algorithms include fast Fourier transform (FFT) [24], short-time Fourier transform (STFT), or using spectrograms based on the Fourier domain [25,26]. Apparently, the mentioned approaches do not consider the effect of features correlation when mapping 1D data to multiple dimensions. The disruption of correlation among feature representation can reduce the precision of ML/DL-based NIDS [27]. To address such concerns, the process of converting non-image data to image format should be improved.

In this paper, an improved framework to convert non-image network traffic to image format is proposed. Based on the studies conducted for this research, it was clear that the process of transforming non-image data into images plays an integral part in NIDS performance. With that in mind, the DeepInsight methodology is fused with a Gabor filter to transform non-image network traffic into images for a CNN-based classifier [28,29]. The suggested idea contributes to the domain of notable approaches in transforming non-image network traffic into an image-based dataset for NIDS. The DeepInsight [28] methodology for transforming non-image data into image format was originally proposed for medical applications. As per the authors, it is very challenging yet crucial to identify small variations in genomics data. This concern led to the development of the DeepInsight methodology. Based on the implementation, the DeepInsight methodology presented well-organized images and achieved high precision on a CNN-based classifier. This encouraged us to implement such a methodology for network traffic data for NIDS. However, non-image data from the genomic field is unlike the data from network traffic. Therefore, the DeepInsight method is fused with the Gabor filter to present an effective and precise framework for NIDS. The Gabor filter is well-known for optimal performance in spatial and frequency domains [30,31]. Merging the DeepInsight with the Gabor filter resulted in well-organized and refined images of network traffic for classification. Later the converted images from four different intrusion detection datasets are classified using a CNN-based classifier. The results highlight that the proposed method can identify different types of network attacks with high precision. The key contributions of the paper can be summarized as follows:

- Proposing a unique framework for NIDS based on image processing. The proposed method enables NIDS to achieve high precision due to the efficient conversion of non-image data to image format.

- The suggested approach also contributes to the domain of converting non-image datasets to image formats.
- Implementing and presenting a comprehensive comparison between the proposed framework and the traditional methods of converting non-image data into image format for DL-based NIDS.
- The proposed framework is implemented on multiple datasets to highlight the efficiency and general application of the suggested approach.

The remaining of the paper is arranged as follows: Section II will discuss the recent literature in the field of DL-based NIDS. The section will also include the prominent approaches that used image processing for DL-based NIDS. Section 3 presents an introduction to the DeepInsight and the Gabor filter as part of the proposed system model. The proposed methodology will be discussed in Section 4. Section 5 will highlight the experimented results of the suggested framework. The comparison and discussion between the proposed framework and recent notable approaches will be elaborated in Section 6 and Section 7 will conclude the paper.

## 2 Related Work

The ML-based NIDS has been used extensively in the past few years. However, shallow ML-based methods are struggling to counter the diversity and complexity of evolving network attacks [32]. With the rapid advancements in DL algorithms, emerging DL-based NIDS are displaying improved precision in detecting anomalies and intrusions [33]. However, achieving a high detection rate against the ever-mutating attack methods is still a challenging task for DL-based algorithms. To address this challenge researchers are always working on techniques to improve the detection rate of DL-based NIDS. For instance, Shone et al. [34] presented a hybrid approach for intrusion detection systems (IDS). The proposed IDS was a fusion of non-symmetric auto-encoder (AF) and random forest (RF). Similarly, Marir et al. [35] proposed a hybrid method for IDS. The proposed method used a fusion of deep belief networks and SVM for high precision IDS. Both the mentioned approach used a hybrid classification. The method adopted by Shone et al. [34] suffered from a low detection rate of minority classes whereas Marir et al. [35] had high computation requirements. The DNN-based hybrid method proposed by [36,37] uses a deep neural network (DNN) as the basic classifier with gated recurrent units (GRU), multi-layer perceptron (MLP), or supervised variational auto-encoder with regularization (SAVAER-DNN) for IDS. Understandably, such hybrid approaches achieve a high detection rate but suffer from high computational requirements. In the same way, Wei et al. [38], presented an approach to optimize the architecture of a deep belief network (DBN) IDS. The authors merged the clustering, searching, and other features of the fish swarm algorithm to optimize the particle swarm optimization (PSO) approach for DBN-IDS. Based on the experimentation, the optimization achieved higher detection but increased the complexity of the DBN-IDS model. On the other hand, NIDS based on image processing is a relatively new approach but is proving to be highly effective. As discussed earlier, the main concern for image-based NIDS is the transformation of non-image data into image format. Some of the most common approaches to converting non-image data to images involve multi-dimensional mappings, using the Fourier domain for image transformation, etc. For instance, in the paper [39], the authors converted the 1D non-image data to a 2D image for a CNN-based classifier. The proposed method achieved high accuracy in detecting most of the attacks, except for the low-frequency attack of U2R. Researchers Todinas et al. [12] and Kim et al. [40] transformed non-image datasets into four-channels RGB-A (Red, Green, Blue, and Alpha) and three-channels RGB (Red, Green, and Blue) images respectively. Then the converted images were classified using a

DL-ResNet50 [12] and CNN-based classifier [40]. Excluding the low-frequency attacks, both classifiers displayed high precision in detecting other attacks. The papers [24] and [25] proposed and implemented Fourier domain-based approaches to transform the non-image datasets into images. Later a CNN-based classifier was implemented for classification. Despite achieving high accuracy, the authors of these papers used only one dataset for implementing the proposed model. Raising the question of the general application of the proposed methodology for converting non-image data to images for NIDS. In summary, hybrid approaches for NIDS are highly accurate in detecting anomalies. However, computational cost and detection of low-frequency attacks are a concern for traditional ML-based NIDS. Whereas image-based NIDS are relatively new but are highly effective in identifying anomalies on the network. However, the process of transforming non-image data into an image is still an open area for improvement.

Tab. 1 presents a summary of papers discussed in this section. The Tab. 1 also highlights the absence of general application of the proposed approaches. As most of the approaches are implemented on one or two datasets. Despite the overall effectiveness and efficiency, each model does have its limitations. Studies in DL-based NIDS are relatively new and have plenty of room for improvement. To this end, we explored and implemented an image-based NIDS with CNN as the classifier. The goal is to construct a framework that is effective and efficient when it comes to anomaly detection. The following section will present a detailed introduction to the components of the proposed methodology.

**Table 1:** Summary of the research papers discussed in the related work section

Reference	Proposed method	Dataset
[29]	Non-symmetric Deep Auto-Encoder (NDAE)	KDD Cup '99, NSL-KDD
[30]	Combination of a Deep Belief Network (DBN) and multi-layer ensemble support vector machines (SVMs).	ICIDS2017, UNSW-NB15, NSL KDD
[31]	Fusion of Recurrent Neural Network (RNN), Gated Recurrent Units (GRU), and Multi-Layer Perceptron (MLP).	KDD Cup '99, NSL-KDD
[32]	Supervised Variational Auto-Encoder with Regularization (SAVAER) combined with a Wasserstein GAN + Gradient Penalty (WGAN-GP).	NSL-KDD, UNSW-NB15
[33]	Deep Belief Network-based Intrusion Detection System (DBN-IDS) fused with Particle Swarm Optimization (PSO) built on the adaptive inertia weight and learning capacity.	NSL-KDD
[34]	Multi-class NIDS model based on CNN classifier.	KDD Cup '99, NSL-KDD
[12]	The researchers suggested the transformation of non-image datasets into four-channels RGB-A images. Then the converted images were used to train a DL-ResNet50 model.	UNSW-NB15, BOUN DDoS
[35]	In this paper, the researchers recommended the conversion of non-image datasets for NIDS into three channels (red, blue, and green) and greyscale images for the CNN-based classifier.	KDD Cup '99, CSE-CIC-IDS 2018

(Continued)

**Table 1:** Continued

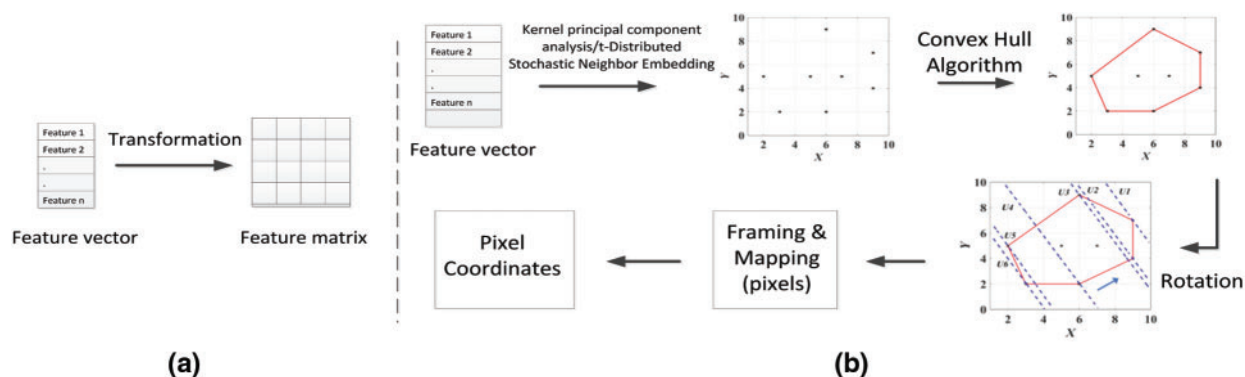
Reference	Proposed method	Dataset
[19]	In this work, the researchers suggested fast Fourier transformation (FFT) to convert non-image data to image format for a CNN-based classifier.	NSL-KDD
[20]	The researcher proposed a short-time Fourier transform (STFT) based spectrogram to convert the non-image dataset into images for a CNN-based classifier	CIC-IDS2017

### 3 System Model Components

This section presents an overview of both the DeepInsight and Gabor filters. As the proposed model is based on the fusion of the mentioned components, it is necessary to have an understanding of how these components work.

#### 3.1 DeepInsight Methodology

As mentioned earlier, the DeepInsight methodology [28] was proposed due to the absence of highly precise approaches for converting non-image data into image format. A general overview of how DeepInsight converts non-image data to image format can be seen in Fig. 1.



**Figure 1:** Graphical representation of the DeepInsight approach (a) Conversion from feature vector to a matrix. (b) Process of converting feature vector to image pixels

The procedure of transforming non-image data into images by using the DeepInsight method is based on two steps. In the first step, the feature vector is converted into a feature matrix by using transformation. The placement of features in the Cartesian coordinates is based on the resemblance of features. Once the features are plotted into a feature matrix, the second step is initiated. In this step, the feature values are mapped based on the feature matrix. This process will generate unique images of each sample (feature vector) and will result in a dataset of images. The newly created dataset can be used to train and predict different classes. For an in-depth understanding of the DeepInsight methodology, both the paper and code (Matlab, Python) can be accessed online [41].

### 3.2 Gabor Filter

In image processing, the Gabor filter is used in numerous applications i.e., feature extraction, edge detection, texture analysis, etc. Gabor being a linear filter plays a key role in modifying, extracting, improving, or representing digital graphical data [42]. 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. For instance, analyzing texture and extracting features from images are two different types of tasks. Mentioned two tasks will require different parameter settings for the Gabor filter. 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 [31]. A 2D Gabor filter can be considered as a sinusoidal signal of specific frequency and direction, regulated by a Gaussian wave. To represent the orthogonal direction, the Gabor filter has both real and imaginary components. The complex, real and imaginary equations of the Gabor filter can be represented as Eqs. (1)–(3) respectively. Both the real and imaginary components can be used separately or can be shaped into a complex number component [43].

$$g(x, y; \lambda, \theta, \phi, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \phi\right)\right) \quad (1)$$

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

$$\Im\{g(x, y; \lambda, \theta, \phi, \gamma)\} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin\left(2\pi \frac{x'}{\lambda} + \phi\right) \quad (3)$$

where:

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

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

$$\lambda = \text{Wavelength of the sinusoidal part},$$

$$\theta = \text{Controls the positioning of the Gabor function},$$

$$\gamma = \text{Spatial aspect ratio},$$

$$\sigma = \text{The standard deviation of the Gaussian covering},$$

$$\phi = \text{The phase offset error of the sinusoidal function}.$$

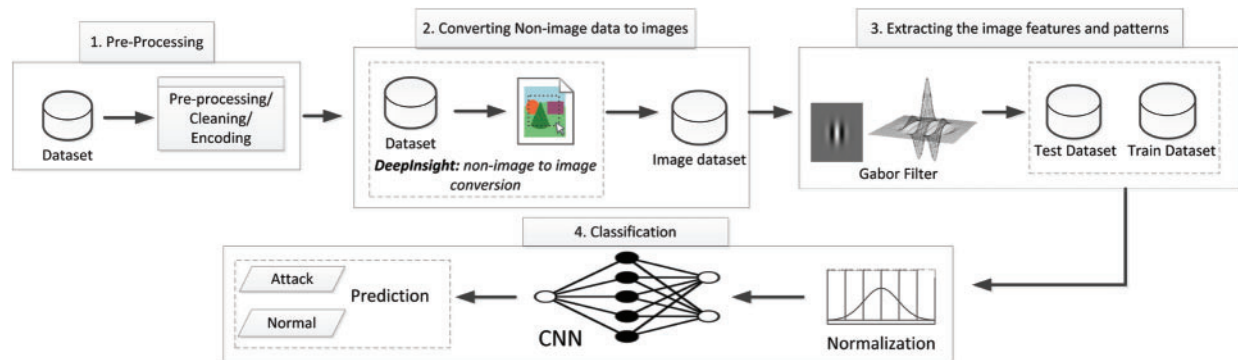
The parameters  $\lambda, \theta, \gamma, \sigma$ , and  $\phi$  define the form of the Gabor function.

## 4 Proposed Methods

In this section, each module of the proposed idea will be discussed in detail. The framework proposed for the NIDS based on the fusion of image processing, Gabor filter, and CNN is illustrated in Fig. 2. The proposed approach is composed of four modules. The modules are data pre-processing, converting non-image datasets to image datasets, extracting features and patterns from images, and CNN-based classification. The main objective of proposing this framework is based on two factors. First, to highlight the importance of using an efficient algorithm for image generation rather than simply converting 1D flow into 2D or 3D images. The second is to explore methods of image enhancements for NIDS based on image processing. Recent researchers are exploring methods such



as the Fourier domain, spectrogram-based images, frequency domain, etc. for image generation or enhancement. In this paper, the Gabor filter is fused with the image generation process to refine the quality of generated images. Further details on the Gabor filter and its significance are discussed in the upcoming section.



**Figure 2:** Illustration of the proposed framework

#### 4.1 Pre-Processing

To validate the effectiveness and generality of the proposed framework, four benchmark NIDS datasets CSE-CIC-IDS2018 [44], CICIDS-2017 [45], ISCX-IDS 2012 [46], and NSL-KDD [47] were implemented. All the mentioned datasets were cleaned by removing missing values, duplicate samples, and infinite data samples. Datasets CSE-CIC-IDS2018, CICIDS-2017, and ISCX-IDS 2012 had 79, 79, and 82 features respectively with numeric values. In the CIC-IDS 2017 dataset, two classes Heartbleed and Infiltration were dropped due to a low number of samples. The three classes representing web attacks were merged into one class of web attacks. The NSL-KDD dataset contains feature attributes that are both numerical and non-numerical. To convert the non-numeric values to numeric values the one-hot encoding was used [48]. The one-hot encoding creates additional binary columns in the dataset to replace the categorical feature. For example, in the NSL-KDD dataset, one non-numeric feature ‘protocol\\_type’ had non-numeric values i.e., tcp, udp, and icmp. To manage the non-numeric feature attributes, one-hot encoding was implemented. As a result the values of tcp, udp, and icmp were converted to 001, 010, and 100 respectively. After the encoding, the one feature column ‘protocol\\_type’ was replaced by three new feature columns. To avoid any biasing, all the datasets were balanced by using the synthetic minority over-sampling technique and edited nearest neighbors (SMOTE-ENN) [49]. Tab. 2 represents each dataset after the pre-processing stage.

**Table 2:** Details of CSE-CIC-IDS2018, CIC-IDS 2017, ISCX-IDS 2012, and NSL-KDD datasets after pre-processing

Dataset	ISCX2018	ISCX-2017	ISCX 2012	NSL-KDD
Number of features	79	79	82	123
Number of classes	10	11	5	5
Number of samples	899,441	2,519,738	1,703,784	351,133

(Continued)

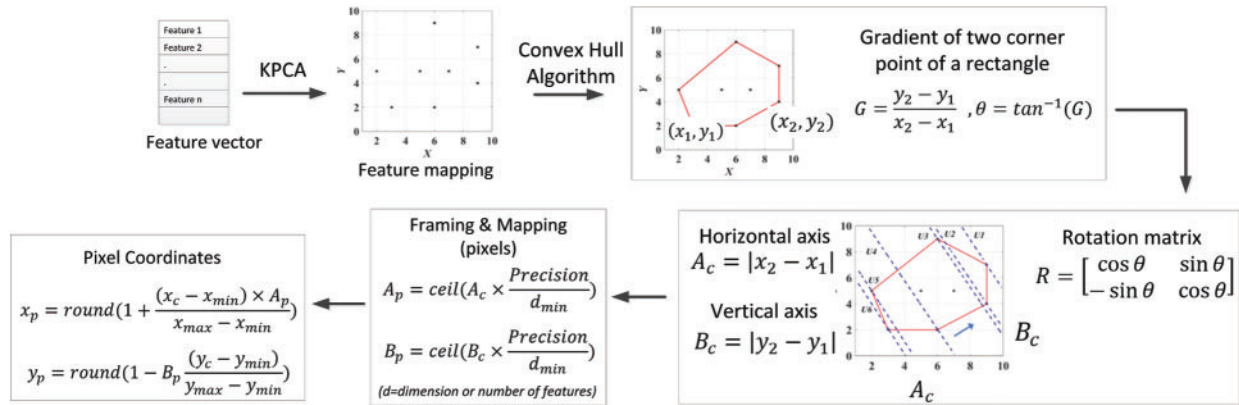
**Table 2:** Continued

Dataset	ISCX2018	ISCX-2017	ISCX 2012	NSL-KDD
Names and samples of each class	Benign: 89811	BENIGN: 225412	BruteForceSSH:	Dos: 73875
	Bot: 89991	Bot: 229731	392492	Probe: 65262
	Brute Force-Web:	DDoS: 229186	DDoS: 337141	R2L: 62433
	89942	DoS GoldenEye:	HTTPDoS: 341806	U2R: 75304
	DDOS	229713	Infiltration: 319241	normal: 74259
	attack-HOIC:	DoS Hulk: 228473	Normal: 313104	
	89999	DoS Slowhttptest:		
	DDOS attack-	229136		
	LOIC-UDP: 90000	DoS slowloris:		
	DoS attacks-	228970		
	GoldenEye: 89851	FTP-Patator:		
	DoS	229823		
	attacks-Slowloris:	PortScan: 229939		
	89887	SSH-Patator:		
	FTP-BruteForce:	229562		
	90000	Web Attack:		
	SQL Injection:	229793		
	89963			
	SSH-Bruteforce:			
	89997			

#### 4.2 Converting Non-Image Data to Image Format

The four NIDS datasets (i.e., [Tab. 1](#)) were transformed into image datasets using the pipeline illustrated in [Fig. 1](#). After the pre-processing, the datasets were transformed using the DeepInsight python tool [41]. The kernel principal component analysis (KPCA) [50,51] was used to map the dataset features from their original space into a 2D space. Due to KPCA, the dataset features are likely to be linearly separable. The obtained points in the Cartesian plane represent the features from the dataset. The acquired points only represent the location of features in 2D space, not the feature itself or its attributes. After plotting the features, the convex hull algorithm [52] is used to create the smallest rectangle possible to represent the plotted features. To facilitate the CNN classifier, the rectangle representing the features is rotated (i.e., horizontal or vertical). As the image must be presented in a framed form to the CNN classifier. Then, the Cartesian coordinates are transformed into pixels. During this transformation, some features are averaged due to the limitation of pixels. This limitation is caused by the size of the image. The resultant frame of pixels will represent the positions of the dataset features. The next step is the mapping of feature attributes to the obtained pixel locations. In the case of overlapping pixels, the respected features are averaged and assigned the same location. Therefore, it is vital to select a suitable size for image representation. An image with a small size and overlapping features may not be a good representation of a non-image dataset. Alternatively, dimension reduction can be a solution to avoid a high number of overlapping pixels. [Fig. 3](#) represents a more elaborate working of the DeepInsight methodology [53].



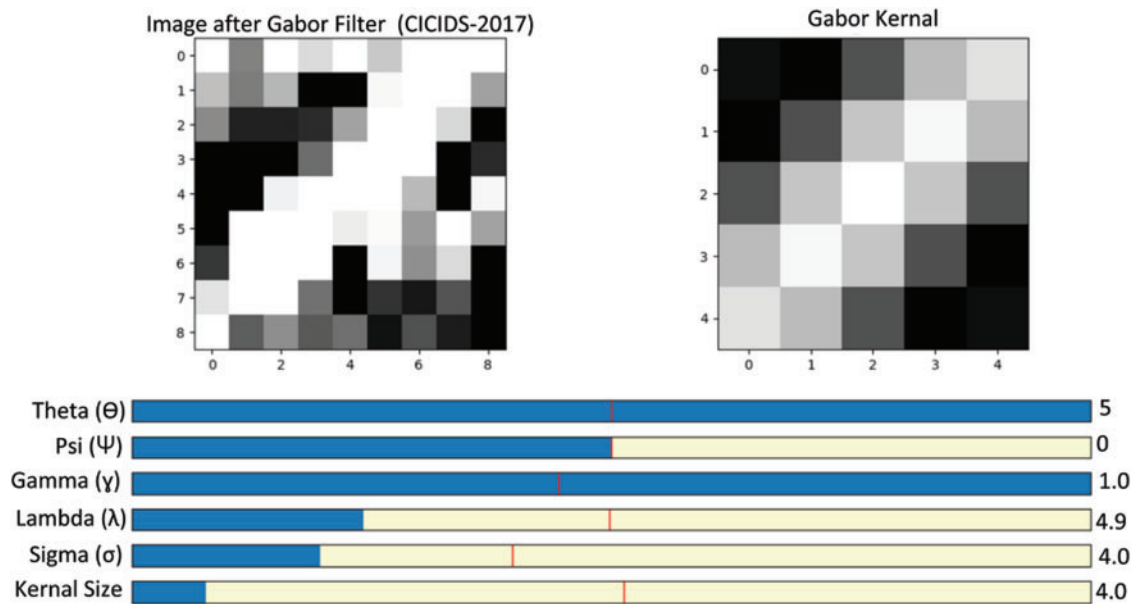


**Figure 3:** Converting non-image dataset to image dataset using the DeepInsight methodology

#### 4.3 Gabor Filter Implementation

After the conversion of non-image data to image format the third step of the proposed method is initiated. In the third step, the Gabor filter starts extracting features from the images generated by the DeepInsight methodology. To the best of our knowledge, an approach to fuse the Gabor filter with image processing for NIDS is never being done before. On the other hand, recent research on enhancing images for ML or DL-based NIDS are highlighted in the earlier section of the paper. The Fourier transform is well versatile for operating in the frequency domain. However, during the transformation, it loses information in the time domain i.e., when a signal occurs or for images where particular frequencies occur in the input image. Whereas the Gabor transformation overcomes this issue by windowing the base function of the Fourier transform with a Gaussian impulse. The resulting transformation gives a sense of when and where a particular frequency occurred. Subsequently, the Gabor filter is a generalized 2D approach of the Gabor transformation [54]. As the filter used in the Gabor transformation is the Gabor filter, which makes it a more appropriate candidate for extracting information from images. Eq. (2) represents the mathematical representation of the Gabor filter implemented for the experimentation. For the proposed framework, a bank of Gabor filter was implemented to identify the most suitable yet general filter for the four datasets. Based on the experimentation the parameter settings in Fig. 4 were identified as the most suitable parameters for the Gabor filter. The Gabor filter was then applied to the images of all four datasets.

After the completion of the third module (i.e., as shown in Fig. 2) the four non-image datasets CSE-CIC-IDS2018, CICIDS-2017, ISCX-IDS 2012, and NSL-KDD are finally converted into image datasets. The images created by datasets CSE-CIC-IDS2018, CICIDS-2017, and NSL-KDD were padded by the DeepInsight method to acquire perfect pixel representation [55]. The images dataset of ISCX-IDS 2012 did not require additional padding. Tab. 3 represents the final image size for each dataset.



**Figure 4:** Gabor filter parameters for the implementation

**Table 3:** Final Image size of each dataset for classification

Dataset	CSE-CIC-IDS 2018	CICIDS-2017	ISCX-IDS 2012	NSL-KDD
Image size	9 * 9	9 * 9	9 * 9	12 * 12

#### 4.4 CNN-Model Configuration

The fourth and final stage of the proposed framework is normalization and then classification. The normalization process scales the values of the image dataset between 0 and 1. In this paper, a CNN-based classifier is implemented due to its potential to achieve high accuracy and computational efficiency [56]. 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 3. Activation function Relu is applied on the convolutional layers and the dense layers. Whereas, softmax is used as an activation function for the output layer. 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 by using the python library Keras. Tab. 4 represents the summary of the parameter settings for the CNN model. The next section will present the implementation and results of the proposed framework.

**Table 4:** Summary of the CNN-model parameters

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

## 5 Implementation and Results

The system used for the experimentation is an Intel Xeon Gold (32 cores, 64 threads) processor with RTX2080ti GPU and 192 GB of RAM. For the framework implementation, python (v 3.6) programming language with GPU-enabled Tensor Flow (v 2.3.1) on the Keras framework is used. The DeepInsight tool based on python is publicly accessible [41]. The tool was downloaded and fused with the proposed approach. The Gabor kernel is created using the cv2 library [57]. Then filter2D method is convolved with the Gabor filter to extract the specific patterns from the images [58]. To highlight the general application of the proposed framework four different NIDS benchmark datasets were used. After converting the NIDS datasets to image datasets with the help of the proposed method, each dataset was classified using the CNN classifier. To evaluate the efficiency of the CNN classification accuracy, precision, F1-score, recall, Cohen's kappa coefficient [59], and receiver operating characteristics (ROC) [60] are measured as performance evaluation metrics. The classification accuracy, precision, recall, and the F1-score were computed using Eqs. (4)–(7).

$$Accuracy = \frac{TruePositive + TrueNegative}{Total} \quad (4)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (5)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (6)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

The accuracy represents the correlation of correctly predicted events to the total number of events. Precision is the percentage of correctly classified attacks to all the samples classified as attacks. The recall is the ratio of all the correctly predicted attack samples to all the samples that are actual attacks. The F1-score is the harmonic mean of the precision and recall. An F1 score is a statistical approach for examining the accuracy of a classification model. The TN (True Negative) and TP (True Positive) are the correctly classified attack and normal events respectively. Whereas, FN (False Negative) and FP (False Positive) are incorrectly classified events as normal and attack, respectively.

The kappa coefficient score is a handy measure to compute the efficiency of a classification model when performing multi-class classification. Mathematically kappa coefficient ( $\kappa$ ) is represented as Eq. (8).

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (8)$$

where ' $p_o$ ' is the general accuracy of the classification model and ' $p_e$ ' signifies the compromise between the classification model estimates and the accurate class values as if occurred by chance. Whereas, the ROC curve is a visual representation of the classification model at all prediction thresholds. Tab. 5 presents the classification results of each dataset based on the proposed framework. Each model is trained for 100 epochs with an 80/20 ratio of train and test datasets respectively. The mentioned epochs and ratios are based on empirical studies [7,9].

**Table 5:** Classification results of each dataset based on the proposed framework

Dataset	CSE-CIC-IDS 2018	CICIDS 2017	ISCX-IDS 2012	NSL-KDD
Accuracy	98.18%	98.70%	92.50%	99.36%
F1 score	98.18%	98.69%	92.20%	99.36%
Precision	98.22%	98.70%	92.22%	99.35%
Recall	98.18%	98.69%	92.23%	99.37%
Kappa score	97.98%	98.57%	91.61%	99.20%

### 5.1 Dataset CSE-CIC-IDS 2018

The proposed framework achieved an overall accuracy of 98.18% on the CSE-CIC-IDS 2018 dataset. Fig. 5 represents the normalized confusion matrix and ROC of each class in the dataset. Based on Fig. 5, the overall detection of each class is highly efficient. The labels in the ROC graph represent each class label as in the classification matrix respectively i.e., Class 0 is Benign, Class 1 is Bot, and so on. Based on the ROC, it is clear that the proposed framework achieved high accuracy in classifying each class.

### 5.2 Dataset CIC-IDS-2017

The proposed model was able to achieve an overall accuracy of 98.70% on the CIC-IDS-2017 dataset. The dataset has eleven classes and Fig. 6 represents the confusion matrix and ROC of each class. Based on Fig. 6, it can be seen that the proposed model was able to detect each class with high precision. The labels in the ROC graph represent each class label as in the classification matrix respectively i.e., Class 0 is BENIGN, Class 1 is Bot, and so on. Based on the ROC, it is clear that the proposed framework was able to detect each class with high accuracy.

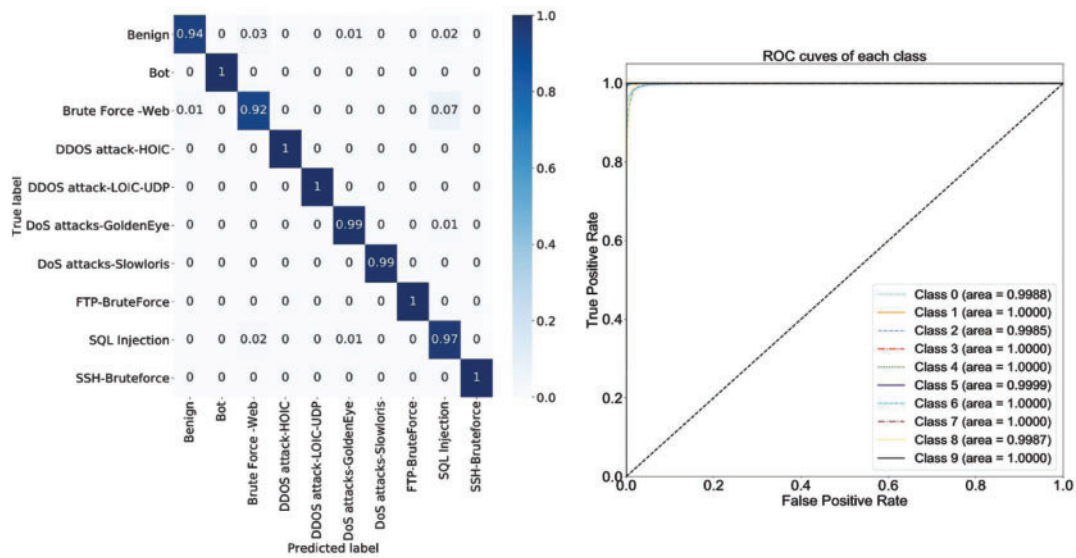


Figure 5: Confusion matrix and ROC of the CSE-CIC-IDS 2018 dataset

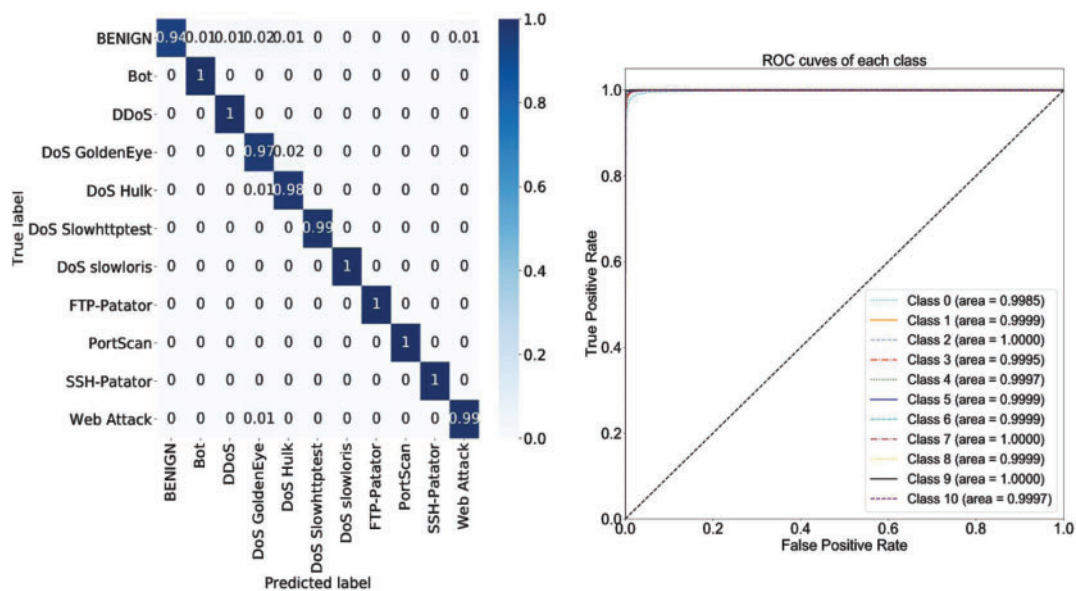
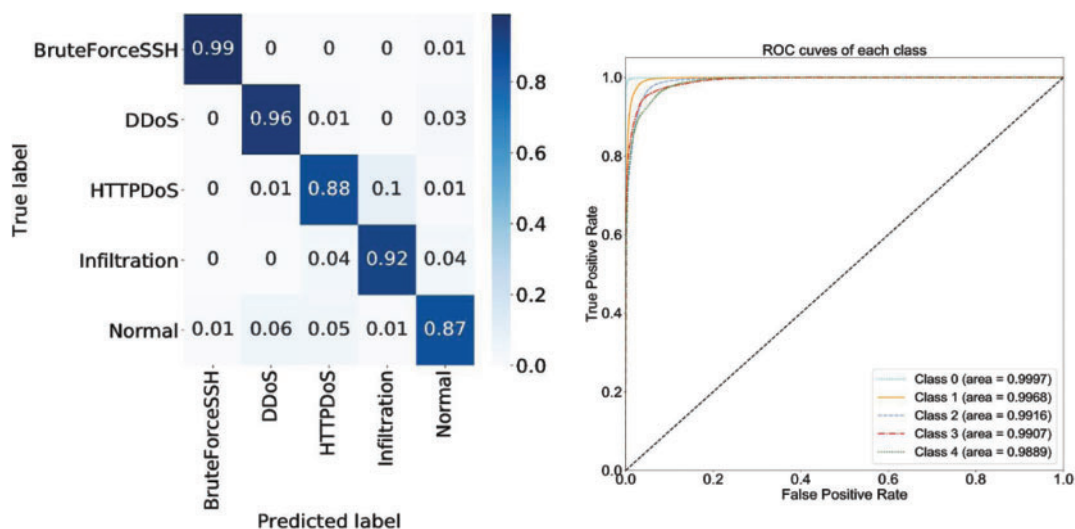


Figure 6: Confusion matrix and ROC of the CICIDS-2017 dataset

### 5.3 Dataset ISCX-IDS 2012

The overall accuracy of dataset ISCX-IDS 2012 based on the proposed model is 92.50%. Which is slightly lower as compared to the other datasets. Based on the confusion matrix in Fig. 7, the classes BruteForceSSH, DDoS, and Infiltration were detected with high precision. Whereas the classes HTTPDoS and Normal were not detected with high accuracy and achieved a detection rate of 88% and 87% respectively. The ROC curve of the dataset ISCX-IDS 2012 can also be seen in Fig. 7. The labels in the ROC graph represent each class label as in the classification matrix respectively i.e., Class

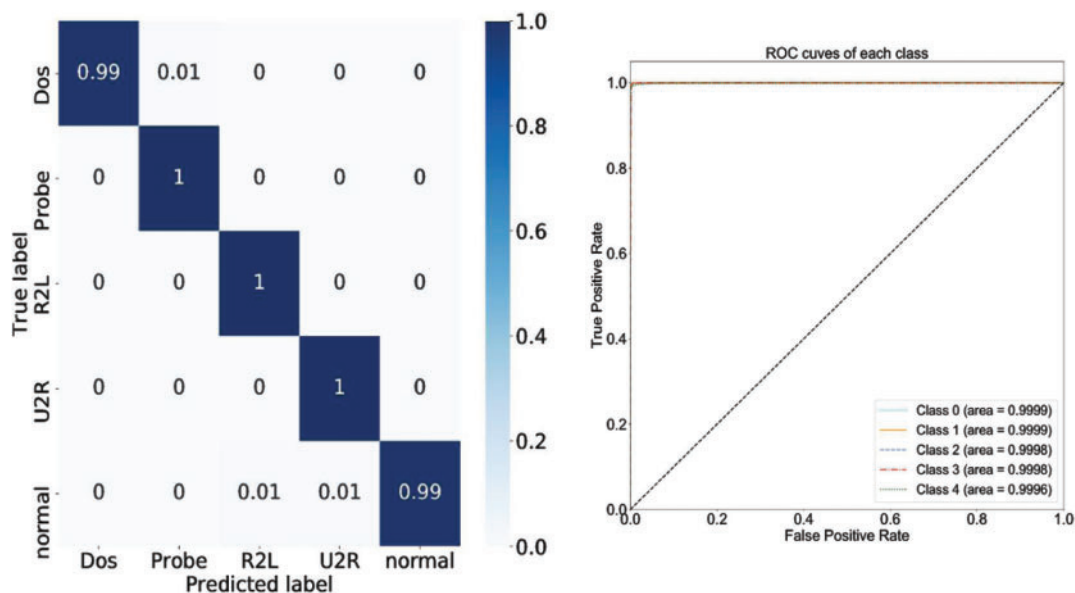
0 is BruteForceSSH, Class 1 is DDoS, and so on. Based on the ROC, the proposed framework was able to detect a majority of the classes with high precision.



**Figure 7:** Confusion matrix and ROC of the ISCX-IDS 2012 dataset

#### 5.4 Dataset NSL-KDD

The NSL-KDD dataset achieved a high precision of 99.36% on the proposed framework. Fig. 8 represents the confusion matrix and ROC of the NSL-KDD dataset. Based on Fig. 8, each class was detected with almost 100% accuracy. The labels in the ROC graph represent each class label as in the classification matrix respectively i.e., Class 0 is Dos, Class 1 is Probe, and so on. Based on the ROC graph, the proposed framework was able to detect each class of the dataset with good accuracy.



**Figure 8:** Confusion matrix and ROC of the NSL KDD dataset

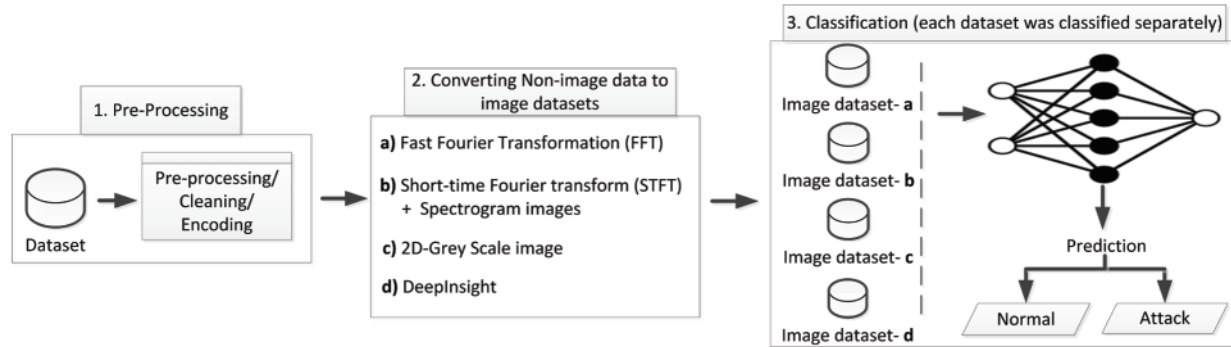


## 6 Comparison and Discussion

The implementation results indicate that the suggested framework displayed high precision on all four datasets. However, apart from the implementation results, the proposed framework must be compared with recent notable approaches. To highlight the capability of the proposed framework, four different approaches to converting non-image datasets to image format for NIDS are implemented and compared. The four approaches are:

- Using the Fast Fourier Transformation (FFT) [24] to convert non-image datasets to image format for a CNN-based NIDS.
- Using the spectrogram images generated by short-time Fourier transform (STFT) [25] to convert non-image dataset to image format for a CNN-based NIDS.
- Converting non-image dataset to a 2D gray-scale image [40] for a CNN-based NIDS.
- Using the DeepInsight methodology to convert non-image data to image format for a CNN-based NIDS.

The flow for implementing the comparison methods can be seen in Fig. 9. Tab. 2 represents the datasets presented to the comparative approaches for image conversion. To provide comparable grounds for evaluation, the proposed framework and comparison methods were implemented with the same parameter settings. Such as the datasets after 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 papers [24, 25], and [40]. The FFT is an optimized and fast algorithm of the discrete Fourier transformation (DFT). The DFT can be represented as Eq. (9). To generate the FFT-based images, 5184 sampling points were taken as per the process explained in the paper [24].



**Figure 9:** Flow of implemented comparison approaches

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} \quad k \in \{0, 1, \dots, N-1\} \quad (9)$$

where  $X(k)$  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 Eq. (10) [25].

$$x[n] = X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n] \omega_m[n-m] e^{-j\omega n} \quad (10)$$

The  $x[n] = (f_1, f_2, \dots, f_{n-1})$  represents the input 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 ( $\omega_m[n]$ ) can be seen as Eq. (11).

$$\omega_m[n] = \frac{1}{2} \left( 1 - \cos \left( 2\pi \frac{n}{N} \right) \right) \quad 0 \leq n \leq N - 1 \quad (11)$$

where 'N' presents the length of observation time. The final step of generating the spectrogram images is based on Eq. (12).

$$\text{Spectrogram} = (m, \omega) = |STFTx[n]|^2 = |X(m, \omega)|^2 \quad (12)$$

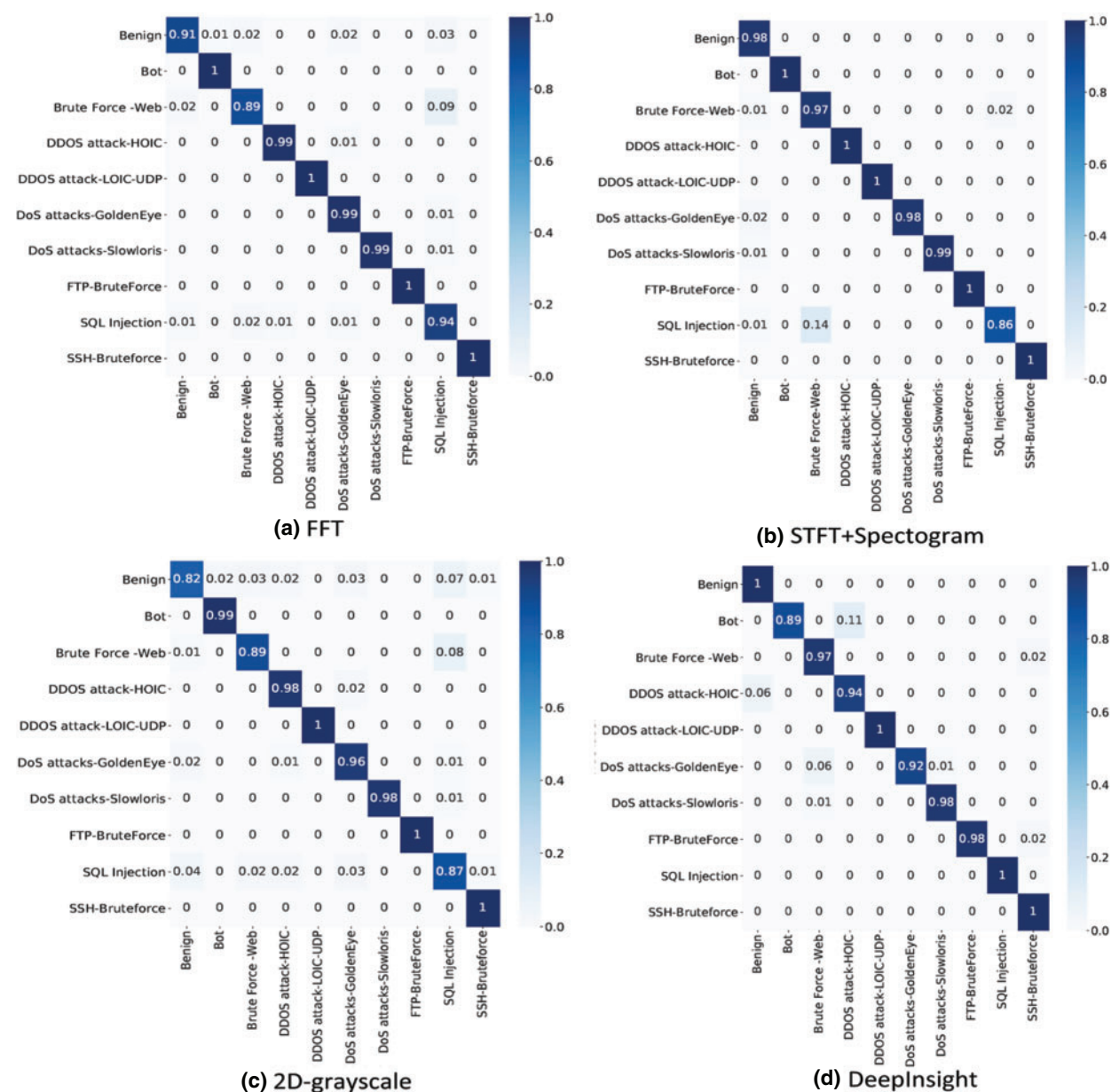
With the help of Eqs. (10) to (12), all four datasets were converted into spectrogram-based images datasets. The paper [40], presented two different methods of generating images from the non-image datasets. Method one presented an approach to generate a 3-channel RGB (Red, Green, Blue) image. While method two presented a one-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  $13 \times 9$  and  $13 \times 6$  pixels are generated for the NSL-KDD and CSE-CIC-IDS 2018 datasets respectively. For comparison purposes, the 2D gray-scale images of four datasets were generated based on the process defined by authors Kim et al. [40].

The fourth approach implemented is based on the DeepInsight methodology of converting non-image data to image format. This implementation highlights the image classification results without the fusion of the Gabor filter. The results highlight that the fusion of the Gabor filter with DeepInsight was able to achieve a much higher classification accuracy as compared to the images based on the DeepInsight approach. 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 proposed framework is generally applicable and can achieve high precision results. Tab. 6, represents the classification results and kappa score of each comparative approach implemented on the CSE-CIC-IDS 2018 dataset. The proposed algorithm achieved an accuracy of 98.18%, which is the highest among all the implemented comparison approaches.

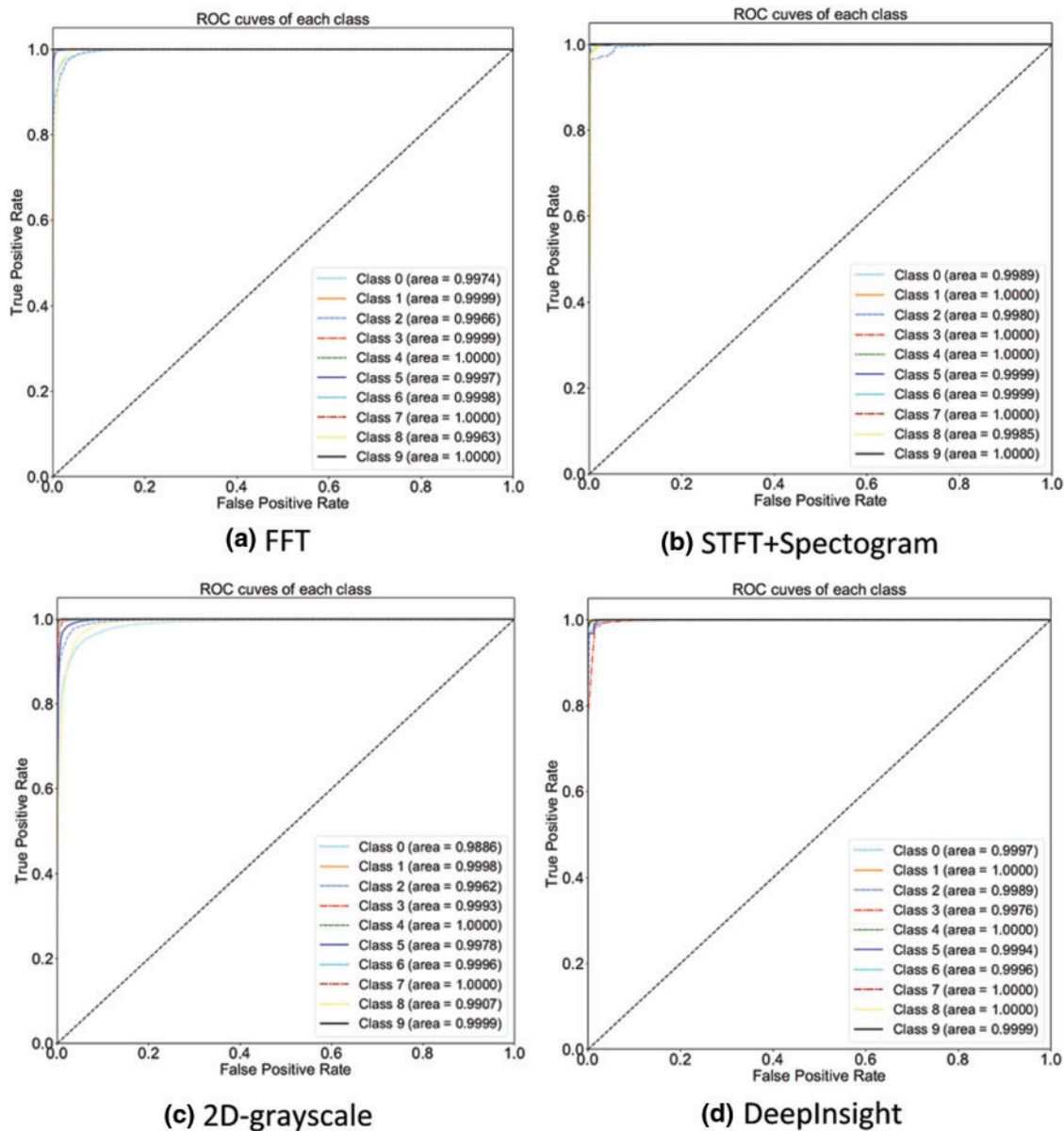
**Table 6:** Results of the proposed framework and implemented comparative approaches on dataset CSE-CIC-IDS 2018

CSE-CICIDS2018	Proposed	FFT	STFT + Spectrogram	2D-grayscale	DeepInsight
Accuracy	98.18%	97.12%	97.80%	97.09%	96.79%
F1 score	98.18%	97.12%	97.74%	97.07%	96.78%
Precision	98.22%	97.19%	97.88%	97.08%	96.73%
Recall	98.18%	97.13%	97.72%	97.08%	96.78%
Kappa score	97.98%	96.80%	97.56%	96.77%	96.43%

The confusion matrix and ROC of the proposed framework can be seen in Fig. 5. Whereas the confusion matrix of each implemented comparative approach on the CSE-CIC-IDS 2018 is shown in Fig. 10. The ROC graph of the implemented approaches are highlighted in Fig. 11.



**Figure 10:** Confusion matrix of each implemented comparative approach on dataset CSE-CIC-IDS 2018



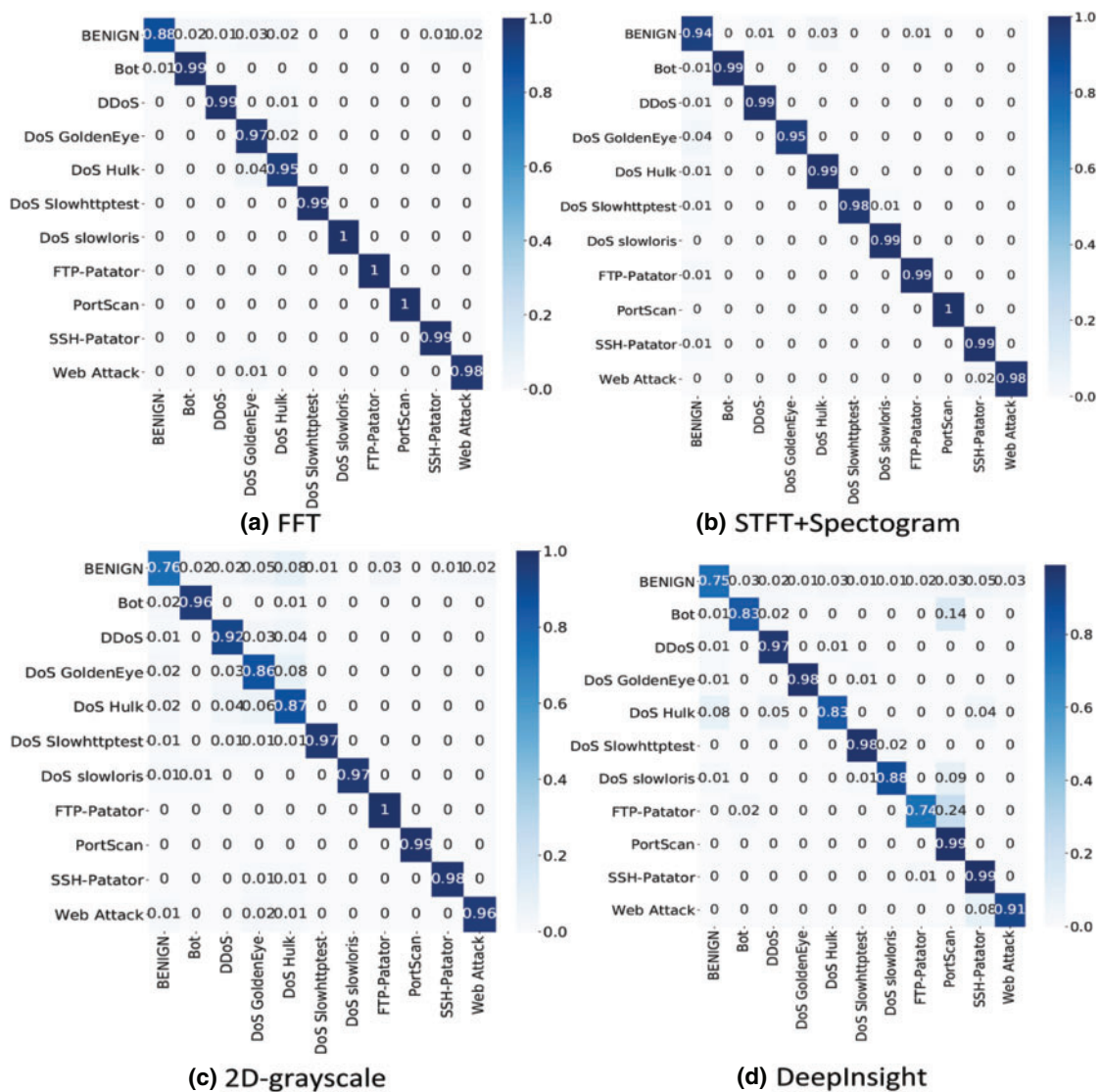
**Figure 11:** The ROC of each implemented comparative approach on dataset CSE-CIC-IDS 2018

Tab. 7 represents the classification results and kappa score of each comparative approach implemented on the CIC-IDS 2017 dataset. The proposed algorithm achieved an accuracy of 98.70%, which is the highest among all the implemented comparison approaches. The 2D-gray scale image implementation was also able to achieve an accuracy of 98.49%, which is very good.

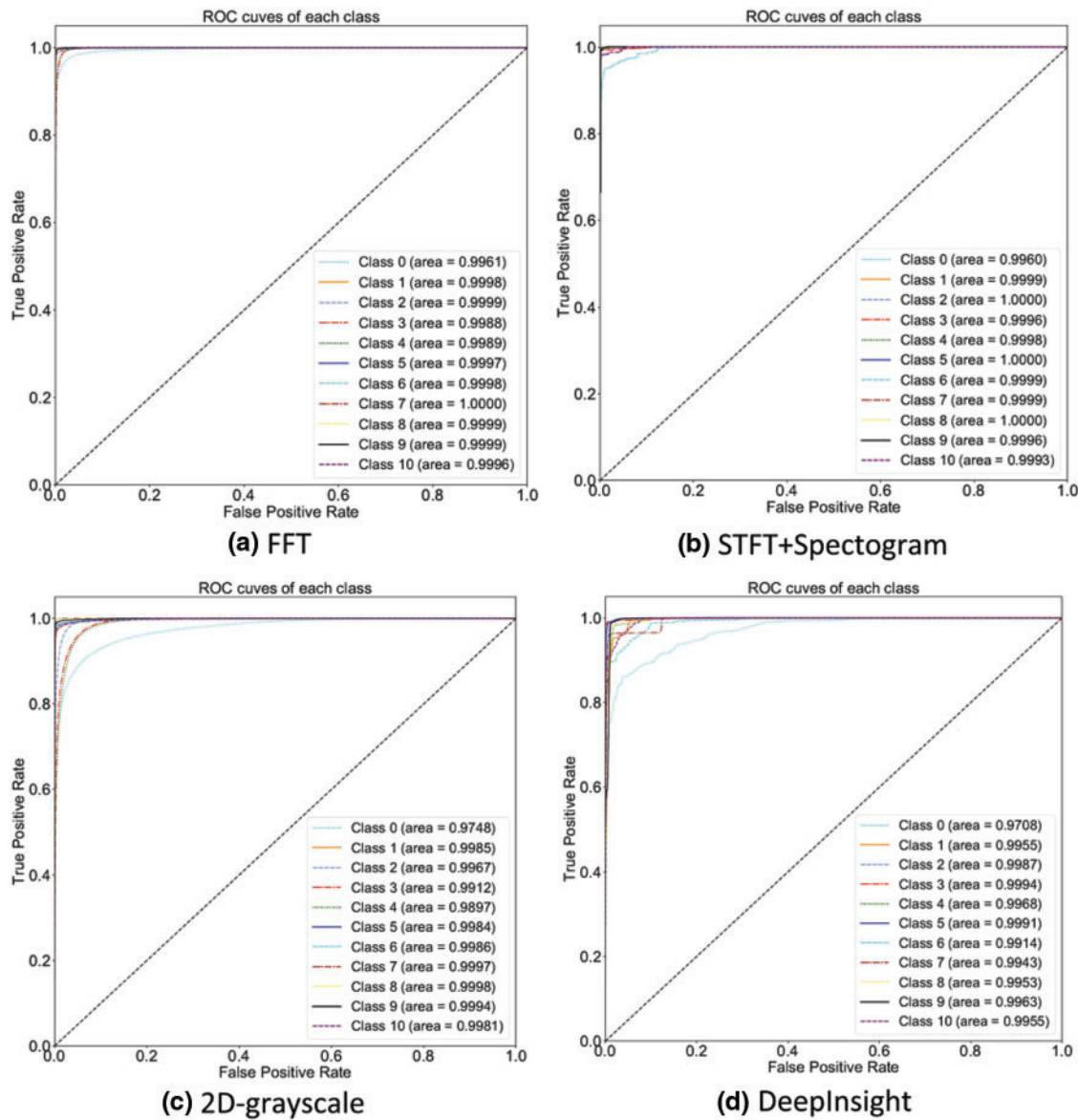
The confusion matrix and ROC of each implemented comparative approach on the CIC-IDS 2017 can be seen in Figs. 12 and 13 respectively.

**Table 7:** Results of the proposed framework and implemented comparative approaches on dataset CIC-IDS 2017

CIC-IDS 2017	Proposed	FFT	STFT + Spectrogram	2D-grayscale	DeepInsight
Accuracy	98.70%	97.65%	98.18%	98.49%	89.59%
F1 score	98.69%	97.62%	98.14%	98.48%	89.66%
Precision	98.70%	97.67%	98.15%	98.49%	90.93%
Recall	98.69%	97.63%	98.16%	98.48%	89.57%
Kappa score	98.57%	97.41%	98.01%	98.25%	88.55%

**Figure 12:** Confusion matrix of each implemented comparative approach on dataset CIC-IDS 2017





**Figure 13:** The ROC of each implemented comparative approach on dataset CIC-IDS 2017

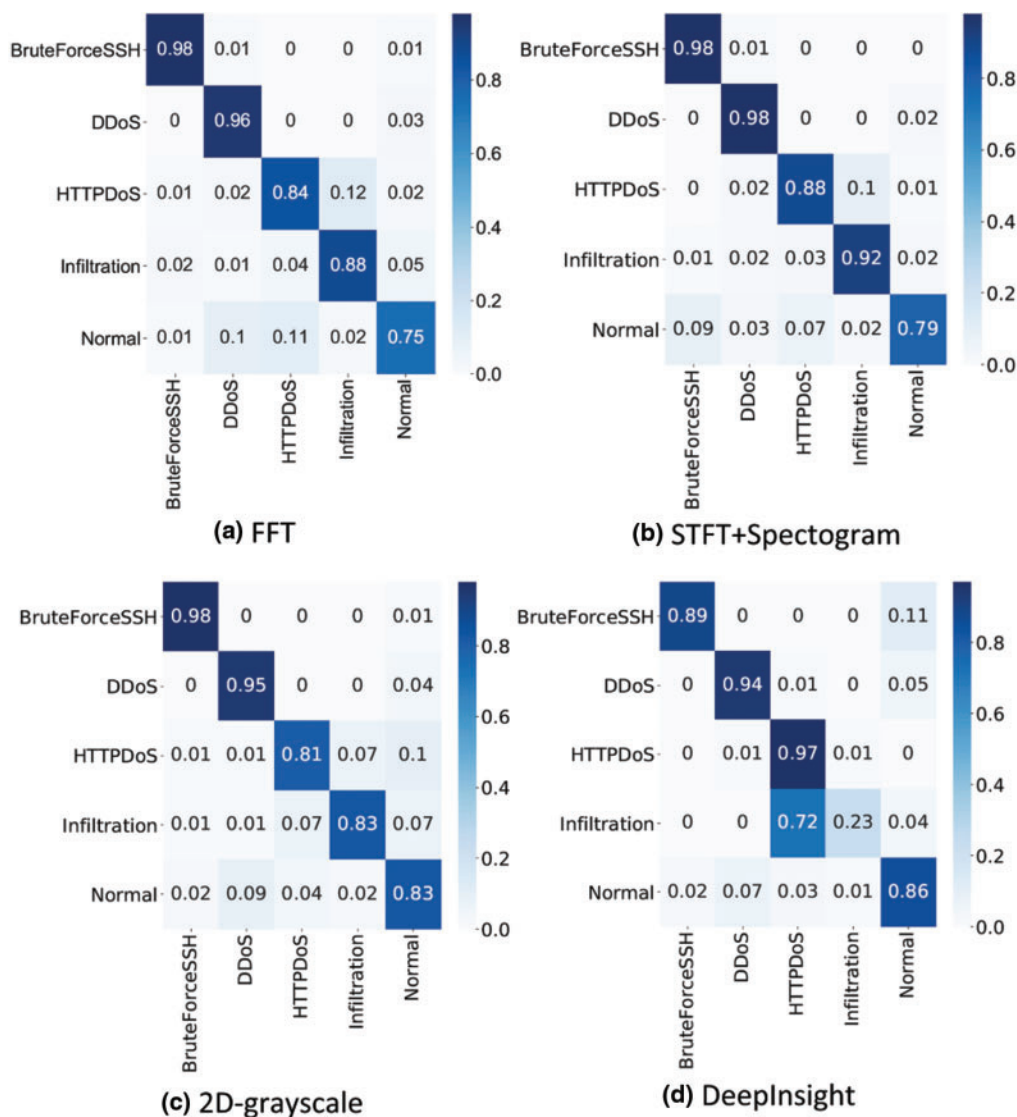
Tab. 8 represents the classification results and kappa score of each comparative approach implemented on the ISCX-IDS 2012 dataset. The proposed algorithm achieved an accuracy of 92.50%, which is the highest among all the implemented comparison approaches.

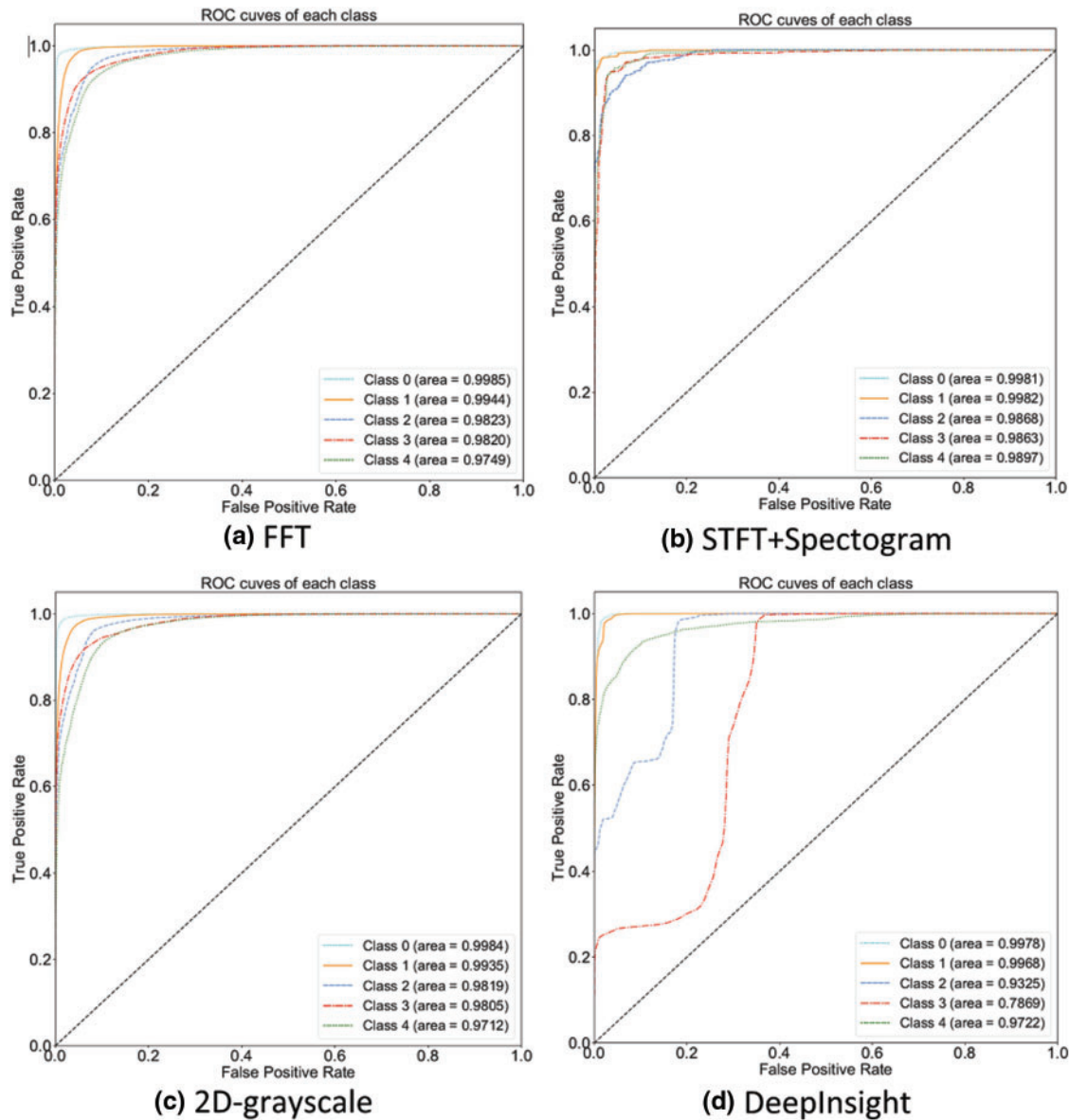
The confusion matrix and ROC of each implemented comparative approach on the ISCX-IDS 2012 can be seen in Figs. 14 and 15 respectively.



**Table 8:** Results of the proposed framework and implemented comparative approaches on dataset ISCX-IDS 2012

ISCX-IDS 2012	Proposed	FFT	STFT + Spectrogram	2D-grayscale	DeepInsight
Accuracy	92.50%	88.85%	91.50%	92.09%	78.78%
F1 score	92.20%	88.30%	91.30%	91.79%	75.56%
Precision	92.22%	88.54%	91.47%	91.95%	82.99%
Recall	92.23%	88.36%	90.91%	91.91%	77.91%
Kappa score	91.61%	86.03%	90.46%	90.10%	73.40%

**Figure 14:** Confusion matrix of each implemented comparative approach on dataset ISCX-IDS 2012



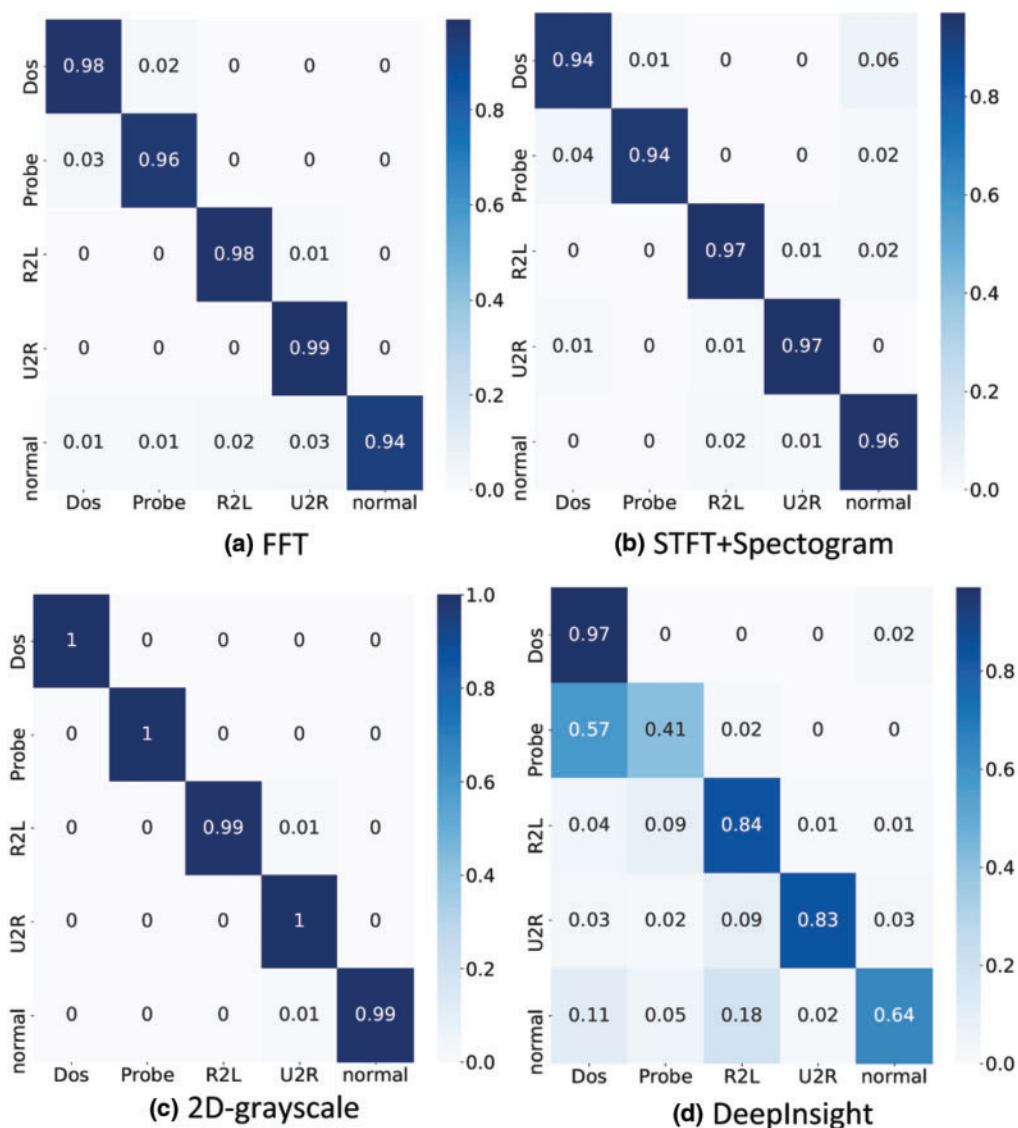
**Figure 15:** The ROC of each implemented comparative approach on dataset ISCX-IDS 2012

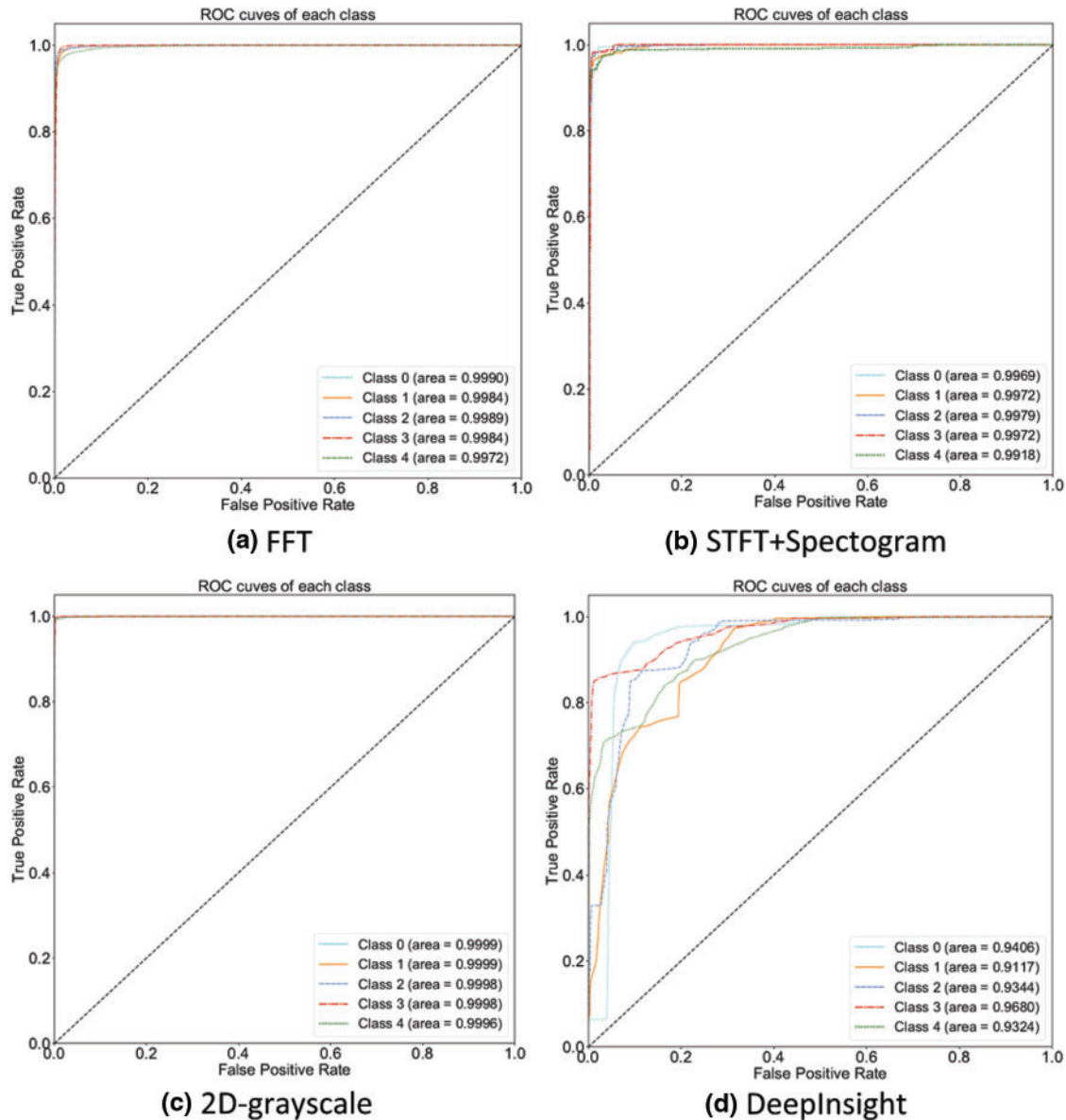
Tab. 9, represents the classification results and kappa score of each comparative approach implemented on the NSL-KDD dataset. The proposed algorithm achieved an accuracy of 99.36%, which is the highest among all the implemented comparison approaches.

The confusion matrix and ROC of each implemented comparative approach on the NSL-KDD can be seen in Figs. 16 and 17 respectively.

**Table 9:** Results of the proposed framework and implemented comparative approaches on dataset NSL-KDD

NSL-KDD	Proposed	FFT	STFT + Spectrogram	2D-grayscale	DeepInsight
Accuracy	99.36%	96.89%	95.55%	84.45%	74.28%
F1 score	99.36%	96.89%	95.55%	84.26%	73.22%
Precision	99.35%	96.90%	95.59%	84.92%	77.33%
Recall	99.37%	96.92%	95.57%	84.19%	73.86%
Kappa score	99.20%	96.11%	94.43%	80.53%	70.64%

**Figure 16:** Confusion matrix of each implemented comparative approach on dataset NSL-KDD



**Figure 17:** The ROC of each implemented comparative approach on dataset NSL-KDD

Based on the results of the implemented competitors, the proposed framework achieved higher precision. The main cause behind the enhanced performance of the proposed methodology can be attributed to the transformation of non-image data into images. As the proposed methods use KPCA for better mapping of features as compared to the other approaches. Due to the KPCA, the images created are an improved representation of the non-image dataset. For instance, the KPCA takes into account the correlation between features of the dataset. Further, the use of the Gabor filter helps the CNN in identifying the most prominent features of each class. In terms of complexity, the Gabor filter plays a crucial part in the proposed framework. For the Gabor filter, the complexity depends on the size of the image. The computing and representation of the filter response at a single point is  $O(M^2)$ , where  $M$  represents the height and width of the particular point. If the filter is applied to an image of

$N \times N$  size, the complexity can be represented as  $O(M^2N^2)$  [61]. Tab. 10 represents the comparative time complexity of the proposed algorithm. Python function ‘time’ is implemented to compute the run time of each algorithm. The time computation is performed for the phase of converting non-image data to image format. As the remaining processes of all the competitive methods are the same i.e., data preprocessing, classification using CNN, etc. Based on Tab. 10, it can be seen that the proposed approach is efficient in terms of time. On the contrary, the 2D-grayscale and DeepInsight approaches are efficient in time consumption but they lack the effectiveness in identifying network anomalies.

**Table 10:** Time (s) consumed by each algorithm in converting non-image data to image format

Dataset	Proposed	FFT	STFT + Spectrogram	2D-grayscale	DeepInsight
CSE-CICIDS2018	9.52	110.36	34.9	5.38	9.41
CIC-IDS-2017	25.97	324.98	36.27	12.54	25.1
ISCX-IDS 2012	16.89	1727.22	209.48	8.41	16.38
NSL-KDD	4.5	44.24	15.5	3.1	4

In the proposed framework, the images created by the four datasets were relatively small i.e., Tab. 3. Due to the small size of the images, the complexity is relatively low. Still, recent research has significantly reduced the complexity of the Gabor filter [62,63]. Such advancements can lead to a much-improved yet low computational implementation of the Gabor filter. Research in the use of image processing with DL for NIDS is relatively new and holds great potential for improvement. Due to the phenomenal growth and developments in the fields of image processing and DL, studies such as the framework presented in this paper can encourage researchers to explore and improve methods of converting non-image data into image format.

## 7 Conclusion

In this study, a new framework for transforming non-image datasets into image datasets is presented. The transformation is based on the fusion of the DeepInsight methodology with the Gabor filter. The transformed images present an efficient interpretation of non-image datasets. Such representation can assist a CNN in understanding deep and useful patterns from the presented images. Based on the understanding, a CNN model can predict network anomalies and threats with high accuracy. To evaluate the efficiency and general application of the recommended framework, four different intrusion detection datasets were implemented. The proposed framework achieved high accuracy on all four datasets. In contrast with the existing work the proposed method can effectively predict low-frequency attacks with low computational requirements. Further, the implementation of four different datasets highlights the general application of the proposed model. The paper also presents an implementation and comparison of recent methods that transform non-image datasets into image format for NIDS. 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.

**Acknowledgement:** M.A.S., and W.P. have written this paper and have done the research.



**Funding Statement:** This work was supported by the National Research Foundation of Korea (NRF) NRF-2022R1A2C1011774.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] CISCO, "Cisco annual internet report (2018–2023) white paper," (Accessed 12 Jan 2022), 2020. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [2] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security: Emerging trends and recent developments," *Elsevier*, vol. 7, no. 8, pp. 8176–8186, 2021.
- [3] M. A. Siddiqi and N. Ghani, "Critical analysis on advanced persistent threats," *International Journal of Computer Applications*, vol. 141, no. 13, pp. 46–50, 2016.
- [4] L. Feiner, "Google, Microsoft plan to spend billions on cybersecurity after meeting with Biden, CNBC," (Accessed 12 Jan 2022), 2021. [Online]. Available: <https://www.cnbc.com/2021/08/25/google-microsoft-plan-to-spend-billions-on-cybersecurity-after-meeting-with-biden.html>.
- [5] M. Bristow, "A SANS 2021 survey: OT/ICS Cybersecurity," (Accessed 20 Feb 2022), 2021. [Online]. Available: <https://www.sans.org/white-papers/SANS-2021-Survey-OTICS-Cybersecurity/>.
- [6] S. Kumar, S. Gupta and S. Arora, "Research trends in network-based intrusion detection systems: A review," *IEEE Access*, vol. 9, pp. 157761–157779, 2021.
- [7] 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.
- [8] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, no. 1, pp. 239–247, 2021.
- [9] M. A. Siddiqi and W. Pak, "Optimizing filter-based feature selection method flow for intrusion detection system," *Electronics*, vol. 9, no. 12, pp. 2114, 2020.
- [10] M. A. Siddiqi and W. Pak, "Efficient filter based feature selection flow for intrusion detection system," in *Proc. of 2020 IWE-ICT*, Gyeongsan, South Korea, 2020.
- [11] M. Chatterjee, "Top 20 applications of deep learning in 2022 across industries, Great Learning," (Accessed 13 Jan 2022), 2019. [Online]. Available: <https://www.mygreatlearning.com/blog/deep-learning-applications/>.
- [12] J. Toldinas, A. Venckauskas, R. Damaševicius, Š. Grigaliunas, N. Morkevicius *et al.*, "A novel approach for network intrusion detection using multistage deep learning image recognition," *Electronics*, vol. 10, no. 15, p. 1854, 2021. <https://doi.org/10.3390/electronics10151854>.
- [13] T. H. Zhao and M. I. Khan and C. Yu-Ming, "Artificial neural networking (ANN) analysis for heat and entropy generation in flow of non-Newtonian fluid between two rotating disks," *Mathematical Methods in the Applied Sciences*, vol. 119, pp. 273, 2021.
- [14] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.*, "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.
- [15] F. Wang, M. N. Khan, I. Ahmad, H. Ahmad, H. Abu-Zinadah *et al.*, "Numerical solution of traveling waves in chemical kinetics: Time-fractional fishers equations," *Economy and Health Sciences*, vol. 30, no. 2, pp. 8898309, 2022.
- [16] J. Zhang, Z. Wang, Y. Zheng and G. Zhang, "Design of network cascade structure for image super-resolution," *Journal of New Media*, vol. 3, no. 1, pp. 29–39, 2021.
- [17] M. J. Ebadi and A. Ebrahimi, "Video data compression by progressive iterative approximation," *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)*, vol. 6, no. 6, pp. 189–195, 2021.



- [18] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021. <https://doi.org/10.1109/TNNLS.2021.3084827>.
- [19] S. Fouladi, M. J. Ebadi, A. A. Safaei, M. Y. Bajuri, A. Ahmadian *et al.*, "Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio," *Computer Communications*, vol. 176, no. 2, pp. 234–248, 2021.
- [20] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Generation Computer Systems*, vol. 113, no. 11, pp. 418–427, 2020.
- [21] G. Andresini, A. Appice and D. Malerba, "Nearest cluster-based intrusion detection through convolutional neural networks," *Knowledge-Based Systems*, vol. 216, no. 4, pp. 106798, 2021.
- [22] A. D. Ossowicka, M. Pietrolaj and J. Rumiński, "A survey of neural networks usage for intrusion detection systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 497–514, 2021.
- [23] K. Wu, Z. Chen and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.
- [24] W. Liu, X. Liu, X. Di and H. Qi, "A novel network intrusion detection algorithm based on fast fourier transformation," in *Proc. of 2019 1st IAI*, Shenyang, China, 2019.
- [25] A. S. Khan, Z. Ahmad, J. Abdulah and F. Ahmad, "A spectrogram image-based network anomaly detection system using deep convolutional neural network," *IEEE Access*, vol. 9, pp. 87079–87093, 2021.
- [26] P. Harder, F. J. Pfreundt, M. Keuper and J. Keuper, "SpectralDefense: Detecting adversarial attacks on CNNs in the fourier domain," in *Proc. of 2021 IJCNN*, Shenzhen, China, 2021.
- [27] C. Wu and W. Li, "Enhancing intrusion detection with feature selection and neural network," *International Journal of Intelligent systems*, Wiley, vol. 36, no. 7, pp. 3087–3105, 2021.
- [28] 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," *Scientific Reports*, vol. 9, Article number, no. 11399, pp. 832, 2019.
- [29] H. S. Munawar, R. Aggarwal, Z. Qadir, S. I. Khan and A. Z. Kouzani, "A gabor filter-based protocol for automated image-based building detection," *Buildings*, vol. 11, no. 7, pp. 302, 2021.
- [30] J. Hu, Z. He, J. Li, L. He and Y. Wang, "3D-gabor inspired multiview active learning for spectral-spatial hyperspectral image classification," *Remote Sensing*, vol. 10, no. 7, pp. 1070, 2018.
- [31] A. Shah, "Through the eyes of gabor filter, Medium," Accessed 18 Jan 2022, 2018. [Online]. Available: [https://medium.com/@anuj\\_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97](https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97).
- [32] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, pp. 4396, 2019.
- [33] H. Zhang, L. Huang, C. Q. Wu and Z. Li, "An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177, no. 18, pp. 107315, 2020.
- [34] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [35] N. Marir, H. Wang, G. Feng, B. Li and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018.
- [36] C. Xu, J. Shen, X. Du and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48697–48707, 2018.
- [37] Y. Yang, K. Zheng, B. Wu, Y. Yang and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020.
- [38] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li *et al.*, "An optimization method for intrusion detection classification model based on deep belief network," *IEEE Access*, vol. 7, pp. 87593–87605, 2019.
- [39] G. Liu and J. Zhang, "CNID: Research of network intrusion detection based on convolutional neural network," *Hindawi*, vol. 2020, no. 2, pp. Article ID: 4705982, 2020.
- [40] J. Kim, J. Kim, H. Kim, M. Shim and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, pp. 916, 2020. <https://doi.org/10.3390/electronics9060916>.

- [41] A. Sharma, "List of published tools, ALOK AI LAB," (Accessed 26 Jan 2022), 2017. [Online]. Available: <http://www.alok-ai-lab.com/tools.php>.
- [42] A. C. Bovik and S. T. Acton, "3.1-Basic linear filtering with application to image enhancement," in *Handbook of Image and Video Processing*, 2<sup>nd</sup> ed., USA: Academic Press, pp. 99–108, 2005.
- [43] D. Gupta, U. Agrawal, J. Arora and A. Khanna, "Chapter 11-Bat-inspired algorithm for feature selection and white blood cell classification," in *Nature-Inspired Computation and Swarm Intelligence*, 1<sup>st</sup> ed., USA: Academic Press, pp. 179–197, 2020.
- [44] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. of 4th ICISSP*, Funchal, Madeira, Portugal, pp. 108–116, 2018.
- [45] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. B. Idris, A. M. Bamhdi *et al.*, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.
- [46] A. Shiravi, H. Shiravi, M. Tavallaee and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.
- [47] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. of 2009 IEEE SCISDA*, Ottawa, ON, Canada, 2009.
- [48] D. Yadav, "Categorical encoding using label-encoding and one-hot-encoder, Towards Data Science," (Accessed 27 April 2021), 2019. [Online]. Available: <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd>.
- [49] R. A. A. Viadinugroho, "Imbalanced classification in python: smote-enn method, Towards Data Science," (Accessed 28 Jan 2022), 2021. [Online]. Available: <https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50>.
- [50] M. M. Hossain and M. A. Hossain, "Feature reduction and classification of hyperspectral image based on multiple kernel PCA and deep learning," in *Proc. of 2019 IEEE RAAICON*, Dhaka, Bangladesh, 2019.
- [51] R. Zhang and G. Zhi, "Using KPCA feature selection and fusion for intrusion detection," in *Proc. of 2010 Sixth ICNC*, Yantai, China, 2010.
- [52] H. Sikchi, "Convex hulls: Explained, Medium," (Accessed 1 Feb 2022), 2017. [Online]. Available: <https://medium.com/@harshitsikchi/convex-hulls-explained-baab662c4e94>.
- [53] D. A. Sharma, "Applying CNN to non-image data by deepInsight method," (Accessed 19 April 2022), 2020. [Online]. Available: <https://www.matlabexpo.com/jp/2020/proceedings.html>.
- [54] U. A. Ahmad, K. Kidiyo and R. Joseph, "Texture features based on fourier transform and gabor filters: An empirical comparison," in *Proc. of 2007 ICMV*, Islamabad, Pakistan, 2007.
- [55] A. Rehman, "Basic operations on images," (Accessed 5 Feb 2022), 2016. [Online]. Available: [https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_core/py\\_basic\\_ops/py\\_basic\\_ops.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_core/py_basic_ops/py_basic_ops.html).
- [56] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan *et al.*, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 53, pp. 307, 2021.
- [57] S. Eksi, "Gabor filter, Kaggle," (Accessed 5 Feb 2022), 2021. [Online]. Available: <https://www.kaggle.com/sameteki/gabor-filter>.
- [58] Sanjay, "Python opencv-filter2D() function, Geeks for Geeks," (Accessed 5 Feb 2022), 2021. [Online]. Available: <https://www.geeksforgeeks.org/python-opencv-filter2d-function/>.
- [59] K. Pykes, "Cohen's kappa: Understanding cohen's kappa coefficient, Towards Data Science," (Accessed 29 April 2021), 2020. [Online]. Available: <https://towardsdatascience.com/cohens-kappa-9786ceceab58>.
- [60] P. N. Tan, "Receiver operating characteristic," in *Encyclopedia of Database Systems*, vol. 1, Boston, US: Springer, 2016.
- [61] G. Amayeh, A. Tavakkoli and G. Bebis, "Accurate and efficient computation of gabor features in real-time applications," in *Proc. of 5th ISVC 2009*, Las Vegas, NV, USA, pp. 243–252, 2009.

- [62] G. G. Lee, C. H. Huang, C. F. Chen and T. P. Wang, “Complexity-aware gabor filter bank architecture using principal component analysis,” *Journal of Signal Processing Systems*, vol. 89, no. 3, pp. 431–444, 2017.
- [63] V. Tadic, T. L. Turukalo, A. Odry, Z. Trpovski, A. Toth *et al.*, “A note on advantages of the fuzzy gabor filter in object and text detection,” *Symmetry*, vol. 13, no. 4, pp. 678, 2021.