IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 March 2023, accepted 23 April 2023, date of publication 28 April 2023, date of current version 5 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271408

RESEARCH ARTICLE

An Attention-Based Convolutional Neural Network for Intrusion Detection Model

ZHEN WANG[1,2] **AND FUAD A. GHALEB**[1]

[1]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia
[2]School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou, Zhejiang 325035, China

Corresponding author: Fuad A. Ghaleb (abdulgaleel@utm.my)

This work was supported by UTM RA Iconic Grant (UTMIcon) 2020 under Grant Q.J130000.4351.09G62.

ABSTRACT Network technology has had a distinctive impact on the entire human civilization and has become an important factor of production in many countries and regions. However, with the widespread popularity of network technology, security flaws have been scattered in various fields, and potential crises may break out by attackers at any time. Therefore, it is crucial to establish a traffic monitoring mechanism for network systems. Some researchers have already implemented intrusion detection models by convolutional neural networks (CNNs) combined with attention mechanisms and achieved good results. However, few attempts have been made to improve the computational efficiency of the model by organizing the appropriate image structure, and the integration of attention mechanisms could be further enhanced. In this study, an attention-based CNN intrusion detection model has been proposed. Together with the image generation methods described in this paper, an efficient and accurate processing flow is formed. To optimize the use of the feature information in the experiments, the feature fields in the experimental images were arranged according to the results of their importance analysis. And a more integrated attention mechanism has been applied to the CNN for building the detection model. A series of comparative experiments were conducted on a subset of the CSE-CIC-IDS2018 dataset, and the results show that the detection process and model proposed in this paper can swiftly complete the detection procedure while maintaining high accuracy.

INDEX TERMS Intrusion detection, convolutional neural network, attention mechanism.

I. INTRODUCTION

There is no doubt in our mind that network technology is enjoying unprecedented popularity in our society now. In addition to being so popular, network technology is becoming a target for ill-intentioned people who are trying to exploit its flaws. Countless cyber-attacks take place around the world every day [1]. For a network system, it is hard to overstate the importance of an effective intrusion detection system.

In response to these cyber crises, many researchers have designed a series of intrusion detection systems (IDSs) based on machine learning and deep learning models. Machine learning-based IDSs [2], [3], [4] perform well in several scenarios. The problem with such methods is that they are not effective in detecting complex attack patterns [5]. Furthermore, they all require excellent feature extraction methods to lay the foundation for the accuracy of subsequent models [6], which can require a lot of experience and expertise, as well as subjective thinking. Deep learning models, relatively speaking, have stronger feature representation [6], [7]. In recent years, with the dramatic growth of data volume and the rapid climb of computing power, more and more researchers have started to apply deep learning models to build IDSs [8], [9], [10]. CNN is an effective deep learning algorithm for complex tasks [11], [12] such as recognizing faces and objects and operating autonomous vehicles [7]. The advantages of CNN can be used to study data features if a suitable way is found to convert the data to be studied into the form of images. Several researchers have practiced this method [13], [14], [15] and have confirmed its effectiveness. Since the introduction of attention mechanisms [16], applications in many fields have started to integrate this idea, which includes IDSs. Attention mechanisms are actually like the way people make decisions daily. When going to judge an object, one will mainly focus on some key features of it, while some unimportant parts can even be ignored. When

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

the attention mechanism is used, it is designed to assess the importance of numerous features to facilitate a more rational perception of the object under consideration [17]. Attention mechanisms have appeared in many forms in intrusion detection systems and have shown good results [18], [19], [20]. Given its good performance, various attention mechanisms are integrated into this paper to better uncover the latent information of the data.

Despite deep learning's strong ability to learn data, there are still some problems in its application [21], which ultimately may result in a lower-than-ideal correct rate and inefficient handling process [22]. The well-designed processes and strategies in this paper enable the model to make efficient use of the data information and accomplish high-quality classification prediction. This study makes the following key contributions.

1) The image is generated by converting most of the features into the form of sparse matrixes. It saves memory space and speeds up computation during the subsequent storage and computation process [23].
2) With the model proposed in this paper, as multiple attention mechanisms are incorporated into the CNN, the input features were evaluated from various perspectives and assigned appropriate weight values, leading to an overall improvement in prediction performance.
3) Such image generation rules, and the model proposed in this paper can work well together to form an efficient and accurate processing flow. While improving the computational speed of the model, it guarantees its classification correctness as much as possible.
4) Finally, a comprehensive experimental analysis and performance evaluation was conducted, summarizing the experimental results and looking into future possibilities.

Following is a summary of the remainder of this study. Section II begins with a description of the background and related works. In Section III, we describe our proposed methodology in detail. Section IV describes the data processing flow and the training and validation process of the model in detail. Furthermore, the experimental results are also discussed and analyzed. In Section V, we summarize the outcomes of the research and suggest possible directions for future research.

II. RELATED WORK

Security is a constant topic, and the technology and means to confront both sides will continue to evolve. Network technologies have received a great deal of attention from researchers because of their widespread popularity and involvement in many dimensions of people's lives. CNNs, recurrent neural networks (RNNs), machine learning, and hybrid models are some of the most common measures that are used in the field of intrusion detection. In this paper, the main focus is on research material related to CNNs and attention mechanisms. The application of CNNs to image and video tasks is widely used, with excellent results in image classification, target recognition, etc. The attention mechanism is very effective for model optimization and can be used in a wide range of models with a variety of application modes. Therefore, a proper combination of CNN models and attention mechanisms should be able to collide with new sparks and probably get better results.

It has been reported that CNN models alone have yielded good detection results for some researchers. Meliboev et al. [24] with one-dimensional CNNs proposed a deep-learning approach for developing efficient and flexible IDSs. With supervised learning, packets are organized into predetermined time frames as intrusion traffic for intrusion detection, and normal and abnormal network traffic is classified and labeled in 1D-CNN. In the comparison experiments, the performance of RF and SVM models are compared in addition to 1D-CNNs with different network parameters and structures. Since CNN can extract high-level features, the experimental results show that it performs better. Wooyeon et al. [13] with the minimum protocol information, field size, and offset proposed the first preprocessing method called ''direct'' for network IDS. Furthermore, they proposed two additional preprocessing mechanisms: ''weighted'' and ''compressed''. An IDS based on CNN and the proposed preprocessing method exhibits meaningful performance on the NSL-KDD dataset. Jiyeon et al. [14] developed and evaluated a CNN-based model on the KDD CUP 1999 dataset and the CSE-CIC-IDS 2018 dataset focusing on various types of Denial of Service (DoS) attacks. According to the experimental results, CNNs perform better at detection than RNNs. Mahmoud et al. [25] used two popular regularization techniques to solve the overfitting problem in IDS, resulting in improved accuracy. The InSDN dataset was used to train and evaluate the performance, and the results show that regularization can improve the accuracy of anomaly detection models using CNN. Using feature correlation analysis and surrounding correlation matrix matrices, Liu et al. [26] convert NetFlow data into NetFlow images to extract and encode features. As a result of feeding the resulting NetFlow images into the designed CNN model, 95.86% accuracy is achieved in detecting the intrusion. OKEY et al. [27] trained five pretrained CNNs using the CIC-IDS2017 and CSE-CICIDS2018 datasets, including VGG16, VGG19, MobileNet, Inception, and EfficientNets. After a series of data preprocessing processes, three models (InceptionV3, MobileNetV3Small, and EfficientNetV2B0) were selected based on their performance to develop an efficient-lightweight ensemble transfer learning (ELETL-IDS) model. According to the evaluation, ELETL-IDS performed better than existing state-of-the-art proposals on common evaluation metrics, achieving 100% accuracy, precision, recall, and F-score.

CNN models can be combined with other deep learning models to obtain more comprehensive feature information for intrusion detection. CNN and Long Short-Term Memory (LSTM) is a popular combination. In ASMAA et al. [28], hybrid intrusion detection systems were developed to take

advantage of the ability of CNNs to extract spatial information and the ability of LSTM networks to extract temporal information. To improve the model's performance, they added batch normalization and dropout layers. Training and validation of the model were performed on CIC-IDS 2017, UNSW-NB15, and WSN-DS datasets, demonstrating good performance in both dichotomous and multiple classification scenarios. In the study by Ruizhe et al. [5], a model consisting of CNN and LSTM components, which are connected in the form of cross-layers, was also proposed. A series of experiments based on the KDD Cup 99 and NSL-KDD datasets showed that the proposed cross-layer feature-fusion CNN-LSTM model outperformed other methods. To extract features and data from large-scale car network data traffic for detecting intrusions and finding anomalous behaviors, Ali et al. [29] constructed a detection model combining CNN and LSTM networks. The experimental results show that the model has a high accuracy of 99.7%. CNNLSTM IDS, developed by Rajak et al. [30], is an IDS framework that can detect attacks of various types using the CNN and LSTM methods. The HIKARI-2021 dataset is used to train the architecture, yielding an accuracy of 93.27. To detect intrusions, Deore et al. [31] use Chimp Chicken Swarm Optimization-driven Deep Long Short-Term Memory (ChCSO-driven Deep LSTM). A CNN is used for the feature extraction process and the Deep LSTM is applied for network intrusion detection. A designed optimization technique is used to train the Deep LSTM and enhance its detection performance. Compared to other algorithms, the present ChCSO algorithm achieved better performance using input data from BoT-IoT and NSL-KDD databases.

Attentional mechanisms are also very active in the field of intrusion detection. A new framework for the 0-day problem was proposed by Jinghong et al. [32] based on a Hierarchical Attention-based Triplet Network with Unsupervised Domain Adaptation (HAT-UDA). To detect unknown attacks, a single-class support vector machine model is trained using benign embeddings. As part of HAT-UDA implementation in new network scenarios, an adversarial unsupervised domain adaptation module is proposed to reduce false positives. Several experiments were designed, and the results proved that HAT-UDA improves detection rates of unknown attacks significantly. Yongjie et al. [33] analyzed network traffic data over time. Their model was based on bi-directional long and short-term memory (Bi-LSTM) and was endowed with an attention mechanism to classify traffic data. It was finally experimentally validated against the UNSW-NB15 benchmark dataset. The results show that the model outperforms numerous leading-edge network IDSs using machine learning models. The Laghrissi et al. [17] study proposed another detection model using LSTM and attention mechanisms. They used four reduction algorithms, named: Chi-Square, UMAP, Principal Components Analysis (PCA), and Mutual Information (MI) algorithms. According to the results of the experiments, using attention with all features and using PCA with 03 components produced the best results on the NSL-KDD dataset. Based on a hierarchical attention mechanism and a bidirectional gated recurrent unit (GRU), CHANG et al. [18] present a network intrusion detection model. The hierarchical attention model is tested on the dataset UNSW-NB15, achieving a detection accuracy of more than 98.76% and a false alarm rate (FAR) of less than 1.2%. An in-vehicle network intrusion detection model called TCAN-IDS was proposed by Pengzhou et al. [34] using a temporal convolutional network with global attention. This model enabled global critical region attention by ignoring unimportant byte changes. Tests demonstrated that TCAN-IDS model is capable of performing real-time monitoring and detecting attacks on publicly known datasets. It has been proposed by Hou et al. [35] to detect intrusions using hierarchical LSTMs combined with attention mechanisms (HLSTM + Attention). A sequence feature is extracted from the network record sequence using the HLSTM utilizing multiple hierarchical structures. Afterward, the attention layer is used to capture the correlation between the features, and it re-distributes the weights of the features, thereby adjusting the network learning process to adapt to each feature's importance in relation to different attack categories. Results of the verification experiment conducted on the intrusion detection benchmark dataset NSL-KDD indicate that the proposed algorithm has a better detection performance. An CNN-attention-BILSTM network model is proposed by Chi et al. [36]. The performance is evaluated according to accuracy, false alarm rate, processing performance, and completeness compared with a convolutional neural network, attention-BILSTM network, and other technologies. In conclusion, the CNN-attention-BILSTM model performed the best on KDD99 when compared with the other models.

Of course, there are also many machine learning models [1], [37], [38], [39], [40], [41] for intrusion detection, but in recent years, with the rapid growth in data volume and the steep climb in computational performance, the advantages of deep learning will become more and more prominent.

III. PROPOSED SYSTEM MODEL

Figure 1 illustrates the learning process of the model. After obtaining the dataset, the data first needs to be preprocessed. First, data cleaning is performed, as the sample size is sufficient, and when each sample is checked, if there are any invalid fields, the sample is discarded. Then, due to a significant imbalance in the sample size, in this study, 5000 samples of each type were randomly selected for the subsequent experiments. Finally, by analyzing the importance of these features and selecting some of the relatively important ones from them, they were organized into an image format in a suitable form.

After converting all the data records into image format, they were used to train and validate the proposed model. The structure of the model proposed in this paper is shown in Figure 2. The block marked by green A in the figure indicates the convolutional neural network. It has n convolutional layers, denoted by L(1) to L(n), respectively. Between every

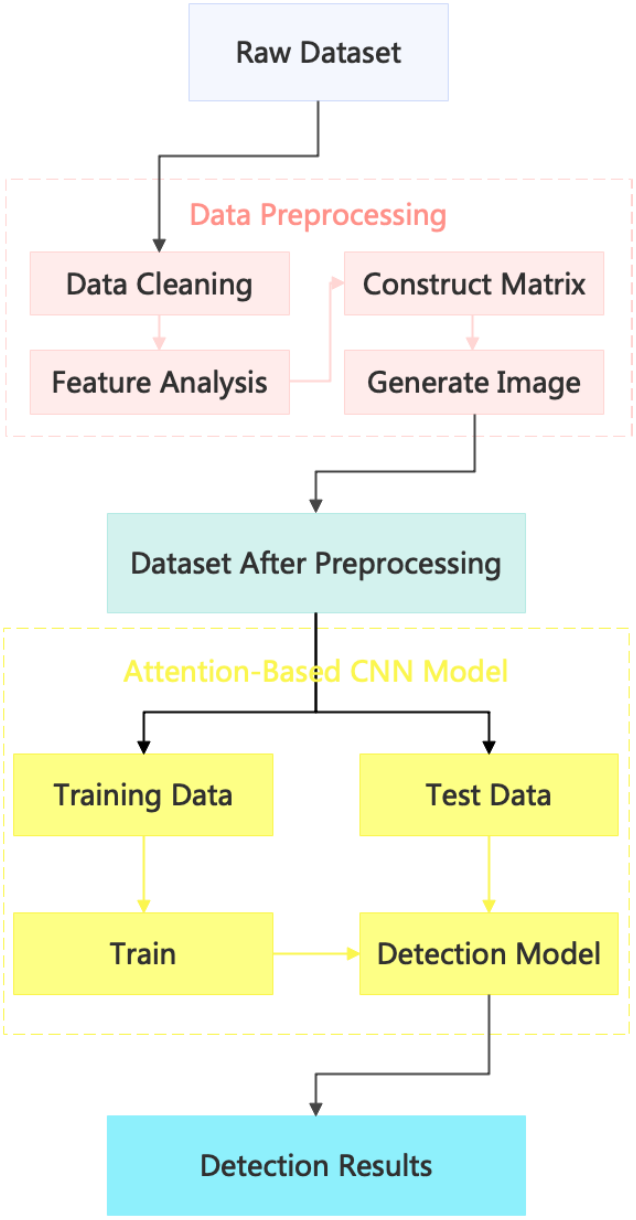

FIGURE 1. Model learning process.

two convolutional layers, there is a Local Attention Module. The Local Attention Module mainly completes the channel and spatial attention mechanisms within the current convolutional layer, and the specific implementation is described as follows. Purple blocks represent the attention logic attached to each convolutional layer B(1) to B(n). The attention tensor obtained from all convolutional layers and the resulting feature tensor output by the CNN make up the attention mechanism in the model. These tensors, which constitute the attention mechanism, are finally connected to find the appropriate attention parameters for that dataset through a training process, thus fulfilling their mission.

The processing logic of the Local Attention Module is illustrated in Figure 3. It contains two sub-modules, called Spatial Attention and Channel Attention. The two sub-modules process the original feature tensor separately to obtain two intermediate tensors and then use these two intermediate tensors and the original features to perform element-wise multiplication operations to finally obtain the refined feature tensor.

The structural details of the two sub-modules Channel Attention and Spatial Attention are illustrated in Figure 4 and Figure 5, respectively. In the Channel Attention sub-module, Global AvgPool and Global MaxPool were applied to this feature tensor respectively. Immediately afterward they both pass through a fully connected layer, and then the two intermediate tensors are summed the element-wise. Finally, the summation result and the original feature tensor are multiplied element-wise to obtain the optimization tensor with the channel attention mechanism attached. In the Spatial Attention sub-module, AvgPool was applied to this feature tensor. Then the new features were extracted by the next convolutional layer. Finally, the new feature tensor and the original feature tensor are multiplied element-wise to obtain the optimization tensor with the spatial attention mechanism attached.

The implementation logic of Spatial Attention can be represented by the following equation.

$$\mathrm{SA}(f) = f \otimes \mathrm{Conv}(\mathrm{AvgPool}(f)) \tag{1}$$

where f is the input feature, $\otimes$ indicates element-level multiplication. Conv stands for convolutional layer operation, and AvgPool is the average pooling.

The Channel Attention can be expressed as the following equations.

$$\mathrm{Path1}(f) = \mathrm{Dense}(\mathrm{GlobalAvgPool}(f)) \tag{2}$$
$$\mathrm{Path2}(f) = \mathrm{Dense}(\mathrm{GlobalMaxPool}(f)) \tag{3}$$
$$\mathrm{CA}(f) = f \otimes (\mathrm{Path1}(f) \oplus \mathrm{Path2}(f)) \tag{4}$$

where Dense, GlobalAvgPool, GlobalMaxPool denote the full connection operation, global average pooling and global maximum pooling, respectively. $\oplus$ stands for elemental-level addition.

Combining the previous expressions, the Local Attention Module can be represented by the following equation.

$$\mathrm{LAM}(f) = f \otimes \mathrm{SA}(f) \otimes \mathrm{CA}(f) \tag{5}$$

The Multi-layer Attention mechanism can be represented by the following equations.

$$f_{i+1} = \mathrm{LAM}(\mathrm{L_i}(f_i)) \tag{6}$$
$$\mathrm{MLA}(f) = \mathrm{Concat}_1^n(\mathrm{Dense}(\mathrm{Conv}(\mathrm{L_i}(f_i)))) \tag{7}$$

where $f_i$is the input information of the i-th convolutional layer. $\mathrm{L_i}$ is the convolution operation at i-th layer. Concat_1^n indicates that the objects in the range are joined together.

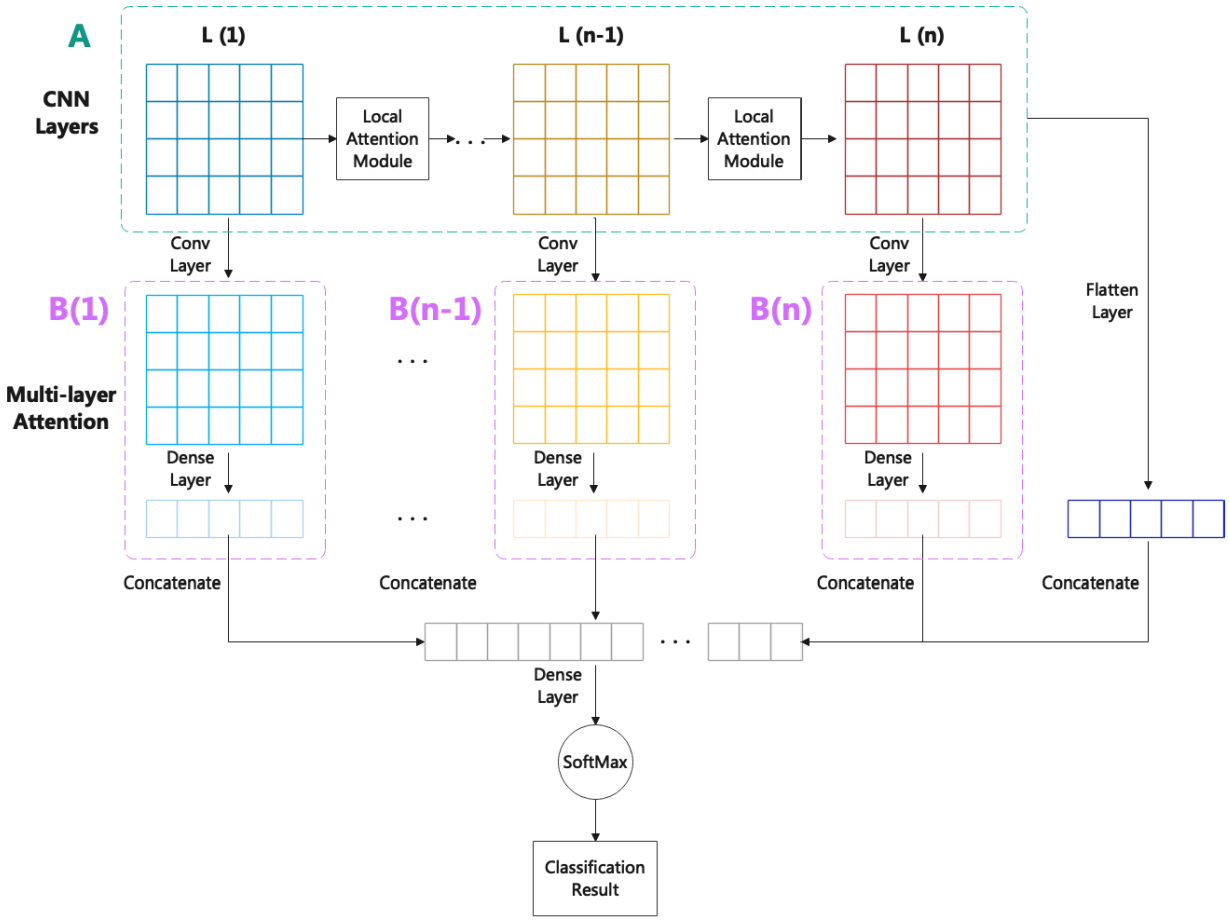

FIGURE 2. Architecture of the proposed model.

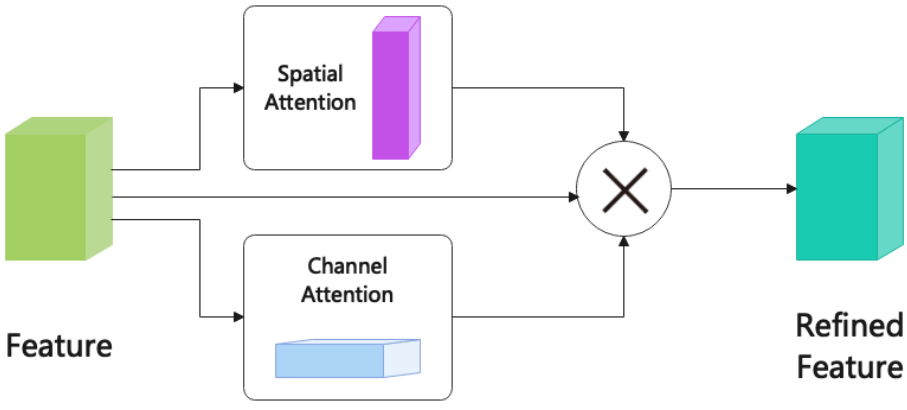

FIGURE 3. Local attention module.

The processing logic of the overall model can be represented by the following equations.

$$\mathrm{RES} = \mathrm{Flatten}\left(\mathrm{L_n}\left(f_n\right)\right) \tag{8}$$

$$\mathrm{MODEL}\left(f\right) = \mathrm{SoftMax}\left(\mathrm{Dense}\left(\mathrm{Concat}\left(\mathrm{MLA}\left(f\right), \mathrm{RES}\right)\right)\right) \tag{9}$$

where Flatten is a multidimensional to one-dimensional conversion operation. RES is the one-dimensional information output from the last convolutional layer. The SoftMax function is used to determine the classification with the highest probability.

IV. EXPERIMENTS AND RESULTS

For programming, this study uses Python as the coding language, numpy, scipy, pandas, imblearn, matplotlib, and the machine learning library sklearn, as well as the deep learning libraries keras and tensorflow to implement all the operations

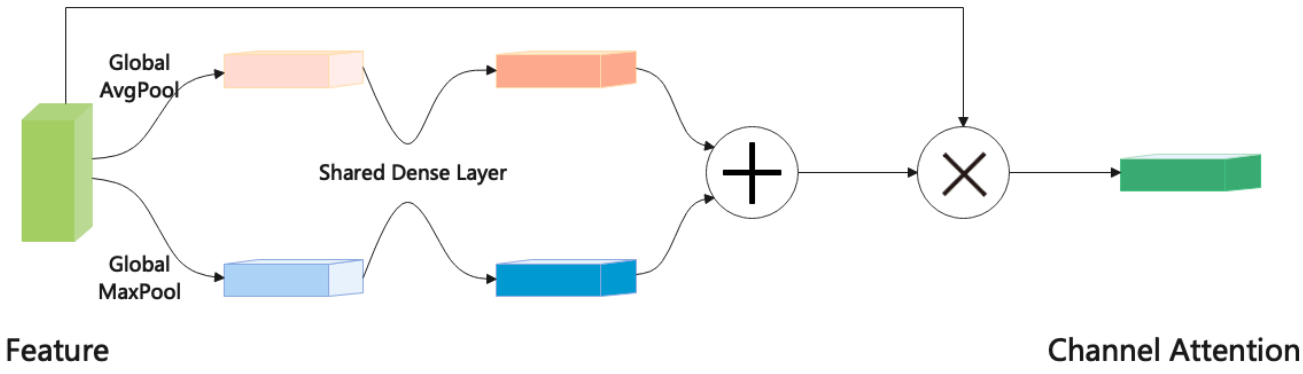

FIGURE 4. Illustration of channel attention.

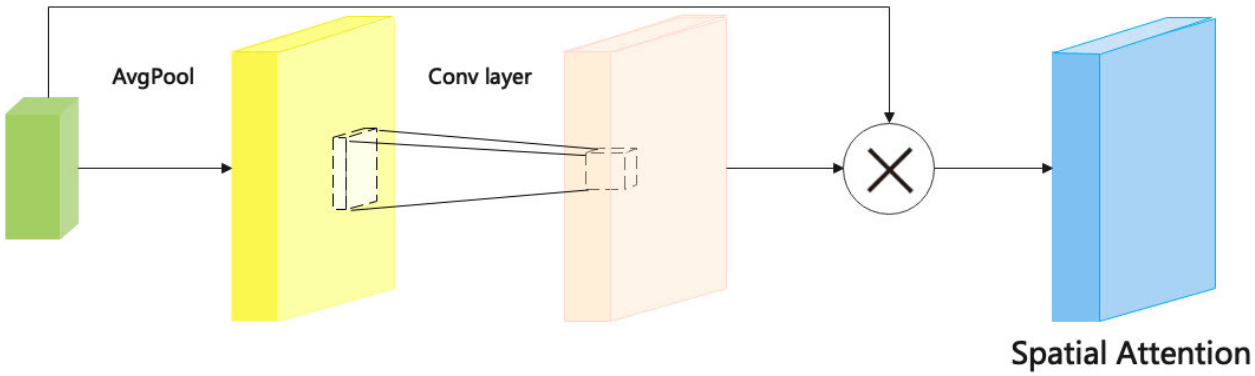

FIGURE 5. Illustration of spatial attention.

in the experiments. The code was executed on a computer running on Intel(R) Xeon(R) CPU @ 2.20GHz, 2200Mhz, 4 Core(s), 32 GB (RAM), Chromium OS 14.0.0, 64-bit and NVIDIA Tesla P100 (16GB) GPU.

Five types of data were extracted from the CSE-CIC-IDS2018 [42] dataset, namely Botnet (1.8%), Infiltration (1.0%), DoS (4.1%), DDoS (8.0%) and Benign (85.1%). The total sample size is 15851066.

Due to the large volume of the datasets, samples containing missing values and invalid values were dropped. Such as the values nan, inf, infinity, etc. There are also negative values where there should be positive values, which are also considered invalid. Additionally, duplicate samples were removed.

The number of different types of samples in the dataset shows an extreme imbalance. Due to the large sample size, this study uses random sampling to avoid the impact of this problem on the model learning process. A sample of 5000 was taken from each type of data for subsequent experiments. For features, SourceIP, SourcePort, and Timestamp these 3 were dropped because an attack can happen at any time and any place. Then 64 features were selected using the SelectKBest algorithm in sklearn. The selected features are numeric, and for each feature, 16 bits are allocated to store its information, which is subsequently organized into a 4×4 matrix. Binary encoding is performed for discrete-valued features, and values over 65535 are treated as 65535. For continuous-valued features, they are normalized to the range [0, 100].

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \times 100 \tag{10}$$

However, before performing normalization, these values need to be further processed because they are found to be very unevenly distributed when analyzing the features. Figure 6 shows the results of the Isolation Forest (iForest) [43], [44] algorithm analysis for a particular feature.

iForest builds an iTree collection for a given dataset and then determines whether a sample is anomalous by calculating the anomaly score.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{11}$$

s(x, n) defines how the anomaly score is calculated for sample x. Where n is the number of training samples, E(h(x)) is the average of h(x) from a collection of isolated trees, and h(x) is determined by the number of edges crossed by x from the root node to the leaf nodes of the iTree.

$$c(n) = 2H(n-1) - (2(n-1)/n) \tag{12}$$

$$H(k) = \ln(k) + \xi \tag{13}$$

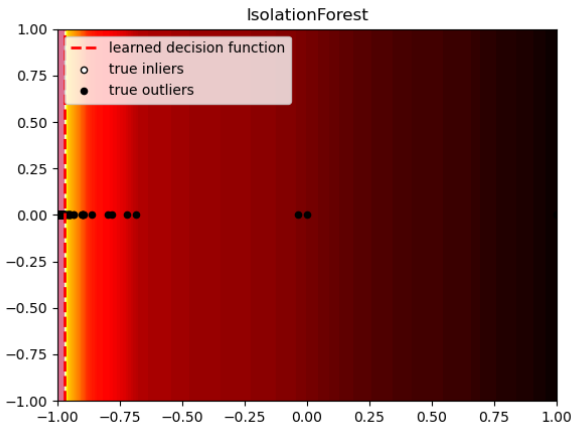

FIGURE 6. Isolation forest analysis.

The definition of c(n) is illustrated by the two equations above, where ξ is Euler's constant.

Figure 6 shows the analysis result obtained when the abnormal sample ratio is set to 0.1. It can be seen that a small number of outliers makes the overall range of values grow tremendously. This situation was very unfavorable for the subsequent experiments of this study, so the outliers were treated as follows.

$$\begin{cases} P_{outlier} = \text{Min}\,(P_{inlier})\,, & P_{outlier} < \text{Min}\,(P_{inlier}) \\ P_{outlier} = \text{Max}\,(P_{inlier})\,, & P_{outlier} > \text{Max}\,(P_{inlier}) \end{cases} \tag{14}$$

where P_{inlier}, and $P_{outlier}$ denote the normal and abnormal points obtained after iForest analysis, respectively. Then, they are normalized according to Equation 10.

In the next step, a one-hot code is created from the range of 0-100 values. That is, when the value reaches a certain level, the corresponding binary position becomes one, while all other bits become zeros. The binary representation of each feature is organized into a 4×4 matrix. A total of 64 features are selected, which eventually form a 32×32 matrix to represent the corresponding sample. When organizing the matrix, each bit obtained from the feature calculation is treated as an element of the matrix. That is, all the elements in the matrix are 0 or 1. The matrix is then converted to image format. 0 and 1 are converted to pixel values of 0 and 255, respectively. The exact organization of the matrix is shown in Figure 7.

The serial numbers 1-64 in Figure 7 are the results obtained by sorting the selected features according to SelectKBest in sklearn.

In the overall structure of the proposed model, there are 3 layers of the convolutional neural network, which will extract image feature information from different granularity for subsequent classification operations. These 3 main convolutional layers are equipped with inter-layer attention mechanisms to allow the model to make better use of the important information in the image. Also in each convolutional layer, the attention mechanism of this layer is embedded. It is mainly divided into two parts: one is Spatial Attention; the other is Channel Attention. This is also to

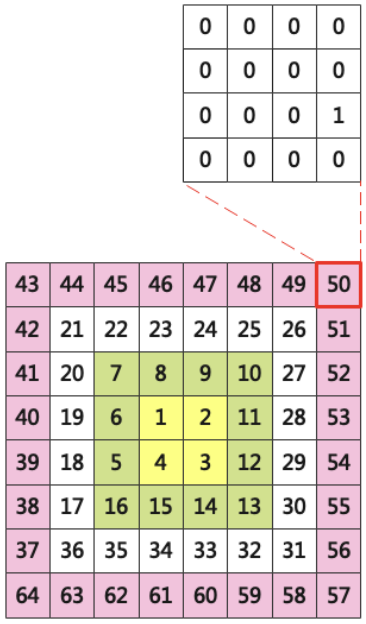

FIGURE 7. Organization of matrix.

enable the model to make full use of the valid information in each layer of convolution operations and to improve the overall model performance. In the course of the experiment, dropout layers were added to some of the layer connections of the model to mitigate overfitting.

In the 3 main convolutional layers, the number of channels is 32, the convolutional kernel size is 4, and the step size is 4. These layers all apply the relu activation function, initialize the parameters by a random standard normal distribution, and use the padding strategy of SAME. The dropout rate for all dropout layers is 25%. For the setting of the model learning rate, the initial value is 0.001, and when the training reaches 20, and 40 rounds, the value is set to one-tenth of the previous value. The total number of training rounds is 50 rounds.

The category identifiers used in the experiments, 0 for Benign and 1-4 for each of the four attack types, which are DoS, DDoS, Botnet, and Infiltration. The evaluation indicators used in this study are as follows.

$$\text{True Positive Rate} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{16}$$

$$\text{False Negative Rate} = \frac{FN}{FN + TP} \tag{17}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{19}$$

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

A true positive is shown as a TP while a true negative is shown as a TN, while a false positive and a false negative are shown as FP and FN respectively.

TABLE 1. Proposed model validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.969	0.0020	0.99	0.97	0.98
1	1.000	0.0000	1.00	1.00	1.00
2	1.000	0.0000	1.00	1.00	1.00
3	1.000	0.0000	1.00	1.00	1.00
4	0.992	0.0059	0.98	0.99	0.99

TABLE 2. Davit validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.959	0.0050	0.97	0.96	0.96
1	0.992	0.0068	0.98	0.99	0.98
2	0.987	0.0000	1.00	0.99	0.99
3	1.000	0.0014	1.00	1.00	1.00
4	0.976	0.0065	0.98	0.98	0.98

TABLE 3. Beit validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.956	0.0153	0.92	0.96	0.94
1	1.000	0.0011	1.00	1.00	1.00
2	1.000	0.0003	1.00	1.00	1.00
3	0.909	0.0003	1.00	0.91	0.95
4	0.936	0.0321	0.89	0.94	0.91

TABLE 4. Halonet validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.950	0.0013	0.99	0.95	0.97
1	1.000	0.0000	1.00	1.00	1.00
2	1.000	0.0000	1.00	1.00	1.00
3	1.000	0.0000	1.00	1.00	1.00
4	0.995	0.0095	0.97	0.99	0.98

TABLE 5. DenseNet121 validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.966	0.0018	0.99	0.97	0.98
1	1.000	0.0000	1.00	1.00	1.00
2	1.000	0.0000	1.00	1.00	1.00
3	0.999	0.0000	1.00	1.00	1.00
4	0.993	0.0068	0.98	0.99	0.98

In addition to the model proposed in this paper, seven other deep learning-based image classification models were used as base learners. Four of the benchmark models were imported from the application module of the Keras platform, namely DenseNet121, NASNetMobile, MobileNet, and ResNet50. The other three models are implemented according to the papers and they are Davit [45], BEiT [46], and HaloNet [47]. All seven models adopted have excellent performance in image classification. Figure 8 shows the validation of all models after each training round during the experiment.

Figure 9 shows the confusion matrix for the validation results of the proposed model in this paper. Table 1 gives the details of the validation results for each type. Figure 10 to Figure 16 show the confusion matrixes for the other seven models' validation results, and Tables 2 to 8 show their specific classification validations, respectively. Table 9 shows the overall performance of all models. It contains four performance metrics, Classification Accuracy, False Positive Rate, and False Negative Rate, which are mainly to examine the detection accuracy of the model, and the fourth metric is to measure the detection efficiency of the model.

The classification accuracy of most of the models can reach above 0.96, which indicates that enough information in the original dataset is retained in the images. All models were able to process more than 350 samples per second, and most of them were able to process more than 500 samples

TABLE 6. MobileNet validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.963	0.0030	0.98	0.96	0.97
1	1.000	0.0000	1.00	1.00	1.00
2	1.000	0.0000	1.00	1.00	1.00
3	0.922	0.0000	1.00	0.92	0.96
4	0.988	0.0281	0.90	0.99	0.94

TABLE 7. NASNetMobile validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.964	0.0015	0.99	0.96	0.98
1	1.000	0.0003	1.00	1.00	1.00
2	1.000	0.0000	1.00	1.00	1.00
3	1.000	0.0000	1.00	1.00	1.00
4	0.994	0.0065	0.98	0.99	0.99

TABLE 8. ResNet50 validation.

Category	True Positive Rate	False Positive Rate	Precision	Recall	F1-score
0	0.963	0.1085	0.61	0.96	0.75
1	0.000	0.0000	0.00	0.00	0.00
2	1.000	0.0000	1.00	1.00	1.00
3	0.993	0.0000	1.00	0.99	1.00
4	0.993	0.1637	0.62	0.99	0.76

TABLE 9. Overall performance.

Model	Classification Accuracy	False Positive Rate	False Negative Rate	Samples/s
Davit	0.9843	0.0039	0.0173	476
BEiT	0.9604	0.0098	0.0398	526
HaloNet	0.9915	0.0022	0.0110	667
ResNet50	0.7788	0.0544	0.2102	833
MobileNet	0.9753	0.0062	0.0254	2500
DenseNet121	0.9932	0.0017	0.0084	513
NASNetMobile	0.9934	0.0017	0.0083	351
Proposed Model	0.9936	0.0016	0.0079	2857

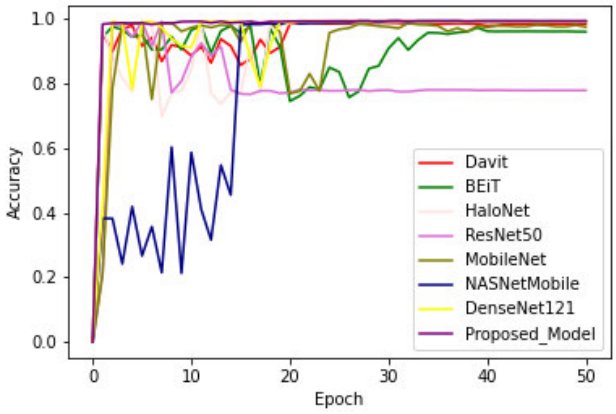

FIGURE 8. Training and validation.

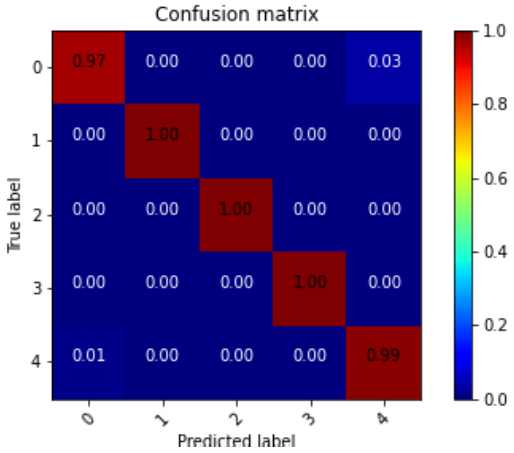

FIGURE 9. Proposed model validation.

per second, with two of them reaching 2500 samples per second (with GPU acceleration during the experiments). The image data constructed in the way described in this paper is very convenient for various types of image processing models to perform calculations. With these evaluation metrics

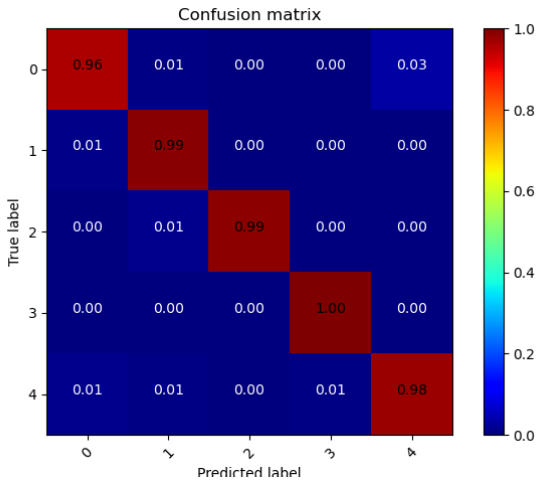

FIGURE 10. Davit validation.

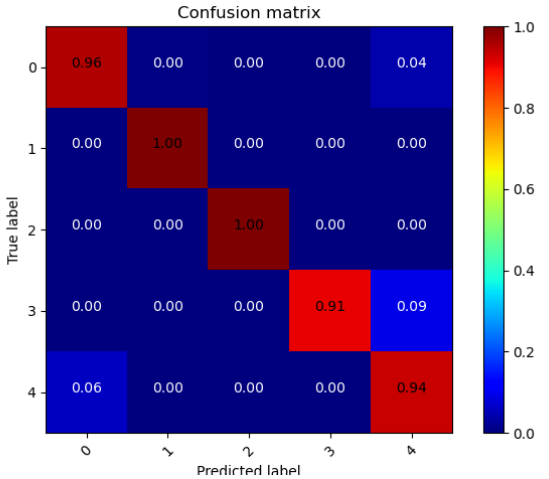

FIGURE 11. BEiT validation.

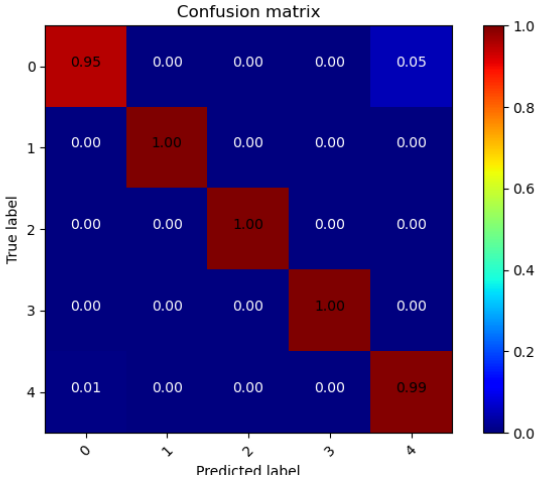

FIGURE 12. HaloNet validation.

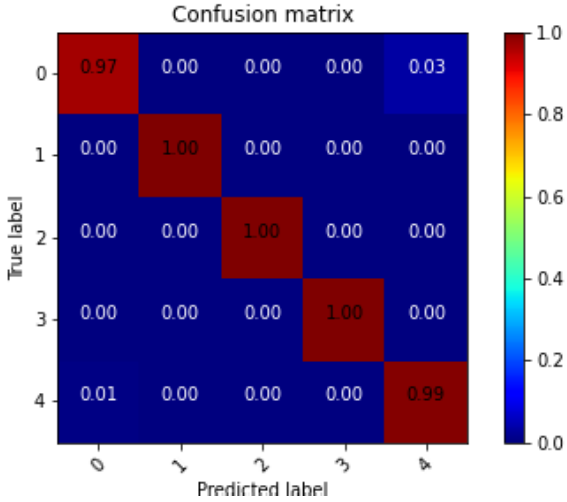

FIGURE 13. DenseNet121 validation.

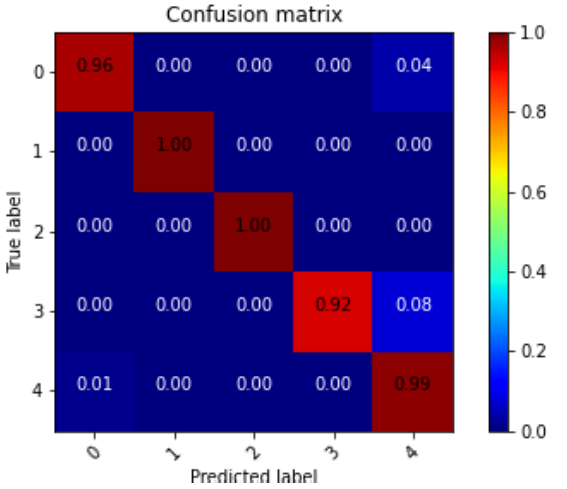

FIGURE 14. MobileNet validation.

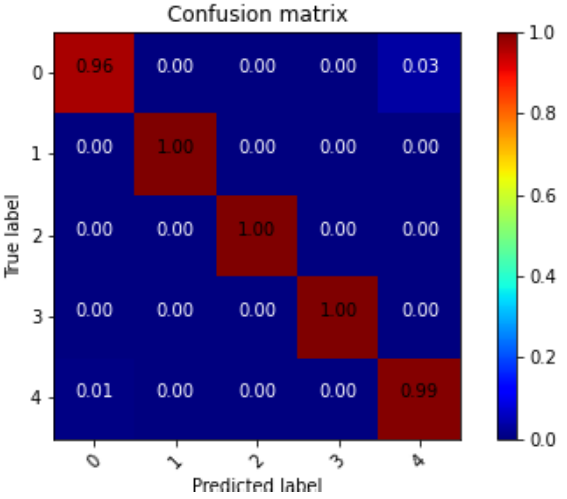

FIGURE 15. NASNetMobile validation.

given in this paper, the model proposed is compared in a comprehensive way with current well-known models as well as new models that have just been proposed in recent years that perform well in the field of image classification. Based on the experimental results, the proposed method can efficiently

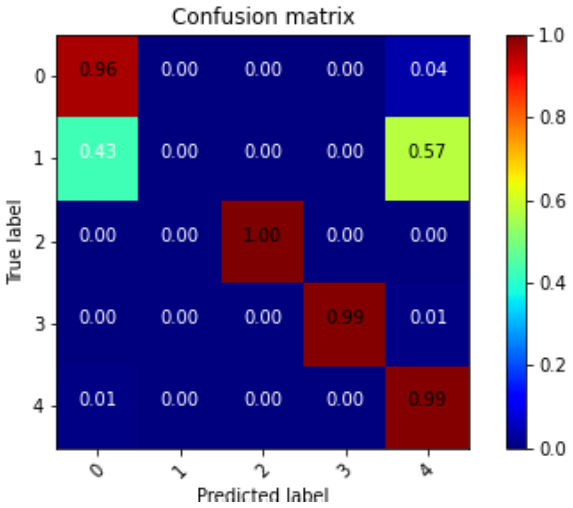

FIGURE 16. ResNet50 validation.

compute sample data while maintaining high classification accuracy.

V. CONCLUSION AND FUTURE WORK

During the experiment session, in addition to the model presented in this paper, a total of seven other models that perform well in the classification of images are also tested on the target dataset. From the experimental results, it can be seen that the image organization described in this paper can well preserve the important information in the original dataset. Moreover, most of the pixels in the composed image data with a value of 0 at the time of computation, which can enhance the execution efficiency of the image computation process. This explains the high execution speed that these models can exhibit when processing data. If these images are stored compressed, they can also exhibit good compression ratios and greatly reduce their space footprint. Reviewing the model that was proposed in this paper, it appears to be a good performer both from the point of view of classification accuracy and from the point of view of execution speed when compared to other models. As can be seen, the overall processing flow and strategy described in this paper can be applied efficiently and accurately to the target dataset to perform classification operations within it.

In future work, we will explore whether it is possible to further compress the data while ensuring that important characteristics are preserved during image composition. Further attention will also be paid to lightweight intrusion detection strategies. In many scenarios, target devices lack enough storage space and computing power, but they are equally vulnerable to attacks. These devices are likely to be the various types of infrastructure that facilitate our lives and should be given sufficient attention.

ACKNOWLEDGMENT

The authors wish to thank Universiti Teknologi Malaysia for supporting this work.

REFERENCES

[1] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Comput. Secur.*, vol. 121, Oct. 2022, Art. no. 102861.
[2] D. Gumusbas and T. Yildirim, "AI for cybersecurity: ML-based techniques for intrusion detection systems," in *Advances in Machine Learning/Deep Learning-based Technologies* (Learning and Analytics in Intelligent Systems), vol. 2, G. Bourbakis, Ed. Cham, Switzerland: Springer, Aug. 2021, pp. 117–140.
[3] L. Canete-Sifuentes, R. Monroy, and M. A. Medina-Perez, "A review and experimental comparison of multivariate decision trees," *IEEE Access*, vol. 9, pp. 110451–110479, 2021.
[4] Y. Pan, W. Zhai, W. Gao, and X. Shen, "If-SVM: Iterative factoring support vector machine," *Multimedia Tools Appl.*, vol. 79, nos. 35–36, pp. 25441–25461, Sep. 2020.
[5] R. Yao, N. Wang, Z. Liu, P. Chen, and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: A cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, no. 2, p. 626, Jan. 2021.
[6] M. S. ElSayed, N.-A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103160.
[7] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of CNN-based network intrusion detection," *Appl. Sci.*, vol. 12, no. 16, p. 8162, Aug. 2022.
[8] Z. Ahmad, A. S. Khan, C. W. 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.
[9] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[11] T. Arias-Vergara, P. Klumpp, J. C. Vasquez-Correa, E. Nöth, J. R. Orozco-Arroyave, and M. Schuster, "Multi-channel spectrograms for speech processing applications using deep learning methods," *Pattern Anal. Appl.*, vol. 24, no. 2, pp. 423–431, May 2021.
[12] N. Lopac, F. Hrzic, I. P. Vuksanovic, and J. Lerga, "Detection of non-stationary GW signals in high noise from Cohen's class of time-frequency representations using deep learning," *IEEE Access*, vol. 10, pp. 2408–2428, 2022.
[13] W. Jo, S. Kim, C. Lee, and T. Shon, "Packet preprocessing in CNN-based network intrusion detection system," *Electronics*, vol. 9, no. 7, p. 1151, Jul. 2020.
[14] 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.
[15] L. Mohammadpour, T. C. Ling, C. S. Liew, and C. Y. Chong, "A convolutional neural network for network intrusion detection system," in *Proc. Asia–Pacific Adv. Netw.*, vol. 46, Aug. 2018, pp. 50–55.
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[17] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "IDS-attention: An efficient algorithm for intrusion detection systems using attention mechanism," *J. Big Data*, vol. 8, no. 1, pp. 1–21, Dec. 2021.
[18] C. Liu, Y. Liu, Y. Yan, and J. Wang, "An intrusion detection model with hierarchical attention mechanism," *IEEE Access*, vol. 8, pp. 67542–67554, 2020.
[19] P. Zhao, Z. Fan, Z. Cao, and X. Li, "Intrusion detection model using temporal convolutional network blend into attention mechanism," *Int. J. Inf. Secur. Privacy*, vol. 16, no. 1, pp. 1–20, Oct. 2021.
[20] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102923.
[21] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021.
[22] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking state-of-the-art deep learning software tools," in *Proc. 7th Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 99–104.

[23] Y. Nagasaka, A. Nukada, and S. Matsuoka, "High-performance and memory-saving sparse general matrix-matrix multiplication for NVIDIA Pascal GPU," in *Proc. 46th Int. Conf. Parallel Process. (ICPP)*, Aug. 2017, pp. 101–110.
[24] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 218–224.
[25] M. S. Elsayed, H. Z. Jahromi, M. M. Nazir, and A. D. Jurcut, "The role of CNN for intrusion detection systems: An improved CNN learning approach for SDNs," in *Proc. Int. Conf. Future Access Enablers Ubiquitous Intell. Infrastruct.*, Jun. 2021, pp. 91–104.
[26] X. Liu, Z. Tang, and B. Yang, "Predicting network attacks with CNN by constructing images from NetFlow data," in *Proc. IEEE 5th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2019.
[27] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt, and D. Z. Rodriguez, "Transfer learning approach to IDS on cloud IoT devices using optimized CNN," *IEEE Access*, vol. 11, pp. 1023–1038, 2023.
[28] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.
[29] A. Alferaidi, K. Yadav, Y. Alharbi, N. Razmjooy, W. Viriyasitavat, K. Gulati, S. Kautish, and G. Dhiman, "Distributed deep CNN-LSTM model for intrusion detection method in IoT-based vehicles," *Math. Problems Eng.*, vol. 2022, pp. 1–8, Mar. 2022.
[30] P. Rajak, J. Lachure, and R. Doriya, "CNN-LSTM-based IDS on precision farming for IIoT data," in *Proc. IEEE 4th Int. Conf. Cybern., Cognition Mach. Learn. Appl. (ICCCMLA)*, Oct. 2022, pp. 99–103.
[31] B. Deore and S. Bhosale, "Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection," *IEEE Access*, vol. 10, pp. 65611–65622, 2022.
[32] J. Lan, X. Liu, B. Li, and J. Zhao, "A novel hierarchical attention-based triplet network with unsupervised domain adaptation for network intrusion detection," *Int. J. Speech Technol.*, pp. 1–22, Sep. 2022.
[33] Y. Yang, S. Tu, R. H. Ali, H. Alasmary, M. Waqas, and M. N. Amjad, "Intrusion detection based on bidirectional long short-term memory with attention mechanism," *Comput., Mater. Continua*, vol. 74, no. 1, pp. 801–815, 2023.
[34] P. Cheng, K. Xu, S. Li, and M. Han, "TCAN-IDS: Intrusion detection system for Internet of Vehicle using temporal convolutional attention network," *Symmetry*, vol. 14, no. 2, p. 310, Feb. 2022.
[35] H. Hou, Z. Di, M. Zhang, and D. Yuan, "An intrusion detection method for cyber monintoring using attention based hierarchical LSTM," in *Proc. IEEE 8th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2022, pp. 125–130.
[36] H. Chi and C. Lin, "Industrial intrusion detection system based on CNN-attention—BILSTM network," in *Proc. Int. Conf. Blockchain Technol. Inf. Secur. (ICBCTIS)*, Jul. 2022, pp. 32–39.
[37] E. Alshahrani, D. Alghazzawi, R. Alotaibi, and O. Rabie, "Adversarial attacks against supervised machine learning based network intrusion detection systems," *PLoS ONE*, vol. 17, no. 10, Oct. 2022, Art. no. e0275971.
[38] A. Singh, J. Amutha, J. Nagar, S. Sharma, and C.-C. Lee, "AutoML-ID: Automated machine learning model for intrusion detection using wireless sensor network," *Sci. Rep.*, vol. 12, no. 1, pp. 1–14, May 2022.
[39] M. S. Farooq, S. Abbas, K. Sultan, M. A. Khan, and A. Mosavi, "A fused machine learning approach for intrusion detection system," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 2607–2623, 2023.
[40] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Exp. Syst. Appl.*, vol. 39, no. 1, pp. 129–141, 2012.
[41] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. S. Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Commun.*, vol. 14, no. 5, pp. 888–895, Mar. 2020.
[42] I. Sharafaldin, A. H. 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*, 2018, pp. 108–116.
[43] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
[44] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, pp. 1–39, Mar. 2012.
[45] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, "DaViT: Dual attention vision transformers," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science), Nov. 2022, pp. 74–92.
[46] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," 2021, *arXiv:2106.08254*.
[47] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12894–12904.

ZHEN WANG received the bachelor's degree in software engineering from Neijiang Normal University, Sichuan, China, in 2013, and the master's degree in computer technology from Chongqing University, Chongqing, China, in 2017. He is currently pursuing the Ph.D. degree with Universiti Teknologi Malaysia.

His research interests include cyber security, intrusion detection, data science, deep learning, and knowledge discovery.

FUAD A. GHALEB received the B.Sc. degree in computer engineering from the Faculty of Engineering, Sana'a University, Yemen, in 2003, and the M.Sc. and Ph.D. degrees in computer science (information security) from the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia (UTM), Johor, Malaysia, in 2014 and 2018, respectively. He is currently a Senior Lecturer with the Faculty of Engineering, School of Computing, UTM. His research interests include vehicular network security, cyber security, intrusion detection, data science, data mining, and artificial intelligence. He was a recipient of many awards and recognitions, such as the Postdoctoral Fellowship Award, the Best Postgraduate Student Award, the Excellence Awards, and the Best Presenter Award from the School of Computing, Faculty of Engineering, UTM, and the best paper awards from many international conferences.