applied sciences

MDPI

Communication

Deep Learning-Based Network Intrusion Detection Using Multiple Image Transformers

Taehoon Kim and Wooguil Pak *

Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea
* Correspondence: wooguilpak@yu.ac.kr; Tel.: +82-53-810-3092

Abstract: The development of computer vision-based deep learning models for accurate two-dimensional (2D) image classification has enabled us to surpass existing machine learning-based classifiers and human classification capabilities. Recently, steady efforts have been made to apply these sophisticated vision-based deep learning models as network intrusion detection domains, and various experimental results have confirmed their applicability and limitations. In this paper, we present an optimized method for processing network intrusion detection system (NIDS) datasets using vision-based deep learning models by further expanding existing studies to overcome these limitations. In the proposed method, the NIDS dataset can further enhance the performance of existing deep-learning-based intrusion detection by converting the dataset into 2D images through various image transformers and then integrating into three-channel RGB color images, unlike the existing method. Various performance evaluations confirm that the proposed method can significantly improve intrusion detection performance over the recent method using grayscale images, and existing NIDSs without the use of images. As network intrusion is increasingly evolving in complexity and variety, we anticipate that the intrusion detection algorithm outlined in this study will facilitate network security.

Keywords: network intrusion detection; multiple image transformers; deep learning; three-channel RGB color image

Citation: Kim, T.; Pak, W. Deep Learning-Based Network Intrusion Detection Using Multiple Image Transformers. *Appl. Sci.* **2023**, *13*, 2754. https://doi.org/10.3390/app13052754

Academic Editors: Tarek Gaber, Shu-Chuan Chu and Chin-Shiuh Shieh

Received: 23 January 2023
Revised: 17 February 2023
Accepted: 18 February 2023
Published: 21 February 2023

1. Introduction

For several decades, various methods have been applied for accurate network intrusion detection. Early threshold- or pattern-based methods offer high accuracy in detecting known intrusions but are susceptible to new, zero-day attacks [1,2]. As zero-day attacks are becoming more prominent, early network intrusion detection systems (NIDSs) were eventually replaced by machine learning (ML)-based NIDSs [3]. Early ML algorithms applied to NIDS were traditional decision tree-based models, such as J48, random forest (RF), and Adaboost [4–8]. All of them possess a rapid rate of learning, yet their primary attribute is their high accuracy and speed of classification, thus making them suitable for NIDS, which requires the processing of high-volume traffic at a fast rate. However, as deep learning (DL) models in image recognition have made remarkable progress over existing traditional models, attempts have been made to apply DL models to NIDS [9]. In the early days, the statistical characteristics of intrusion and normal sessions were expressed as feature vectors, and a simple approach that used them as inputs to DL models, such as deep neural networks (DNNs), was mainly used [10].

Utilizing existing features as inputs of the DL model yields satisfactory accuracy in terms of detection performance. Finally, we must develop an approach to image non-image datasets while minimizing the loss of information. To solve this problem, studies using dimension-reduction algorithms have been conducted. Using principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), the non-image dataset

was converted into a dataset with two-dimensional (2D) images, and it was proved that convolution neural networks (CNNs), which are vision-based DL models, were applied to the dataset, resulting in significantly improved classification accuracy [11–13].

The previous study developed a novel and quite effective approach; however, further research is needed on the process of converting the dataset of NIDS into 2D images using various 2D image conversion methods. Dimension-reduction algorithms such as t-SNE, PCA, kernel PCA (kPCA), and uniform manifold approximation and projection (UMAP) exist [14–16]. kPCA can apply diverse nonlinear kernel functions, such as radial basis function (RBF), cosine, and polynomial [17]. Thus, considering the candidates of 2D image conversion methods, it is necessary to investigate which method is best for transforming NIDS datasets into 2D images, and whether one algorithm or a combination of algorithms is preferable. The throughput and detection accuracy of 2D image conversion-based NIDS cannot be improved optimally unless sufficient research results are obtained. This is a very important issue given that network intrusions are becoming more diverse and data traffic is growing rapidly.

This study devised guidelines in NIDS research by presenting the most suitable 2D image conversion method for the NIDS domain using and evaluating various dimension reduction algorithms to address these problems. We also verify whether the performance can be improved using a combination of various algorithms instead of a single-dimension reduction algorithm. Through this process, we finally introduced the best 2D image-conversion algorithm for non-image-based NIDS datasets. The contributions of this study are as follows:

(1) It enables NIDS development that can provide optimal detection performance by affording guidelines for applying image DL models to NIDS through a comprehensive analysis of various dimension reduction algorithms and classification performance.
(2) The classification performance of NIDS can be maximized, enabling accurate intrusion detection by devising a method for converting non-image-based NIDS datasets into 2D images that are optimized for image DL models.

The remainder of this paper is organized as follows: Section 2 explains the related research in detail. Section 3 performs a NIDS performance analysis based on the 2D image conversion method and proposes a novel NIDS using this result. Section 4 analyzes the performance of the proposed NIDS by comparing it with existing state-of-the-art methods. Finally, Section 5 presents the conclusions of this study.

2. Related Work

Previously, NIDS adopted a simple rule-based method, but as intrusion methods became more diverse and zero-day attacks became more prominent, ML was actively used to detect intrusion based on prior knowledge of intrusion to minimize administrator intervention and increase the detection rate. Currently, we will consider the direction of technology development from early NIDS to recent deep-learning-based NIDS.

2.1. Threshold/Pattern-Matching-Based NIDS

Early NIDS used simple rules to detect or block intrusion based on a simple analysis of traffic. For example, if an ICMP echo response packet is received at over 10 per second, it is detected as a Smurf attack, or if the number of half-open sessions is over five per second, it is evaluated as a synchronize sequence number (SYN) packet scan attack [18,19].

Gradually, as network attacks gradually shift from layer 3 and layer 4 attacks, NIDS requires deep packet inspection [20]. A NIDS which supports deep packet inspection attempts to match pre-implemented patterns for the application layer payload, extract information regarding the application layer through matching results, and use them to determine whether to attack and control traffic accordingly. A typical example of deep packet inspection is malicious code detection. To perform a cross-site scripting attack (XSS), specific code is injected into the website, which is then found by pattern matching using regular expressions [21].

Threshold- and pattern-matching-based approaches can efficiently reduce false positives as they leverage methods designed to identify network intrusions based on actual data. It can easily optimize pattern search using an advanced pattern-matching algorithm that can efficiently search for multiple rules when constructing set pattern-matching rules. "Aho–Corasick algorithm [22]". Therefore, although the threshold-based and pattern-matching methods are initial methods, they are still employed in recent NIDS. In particular, the pattern matching-based NIDS is being expanded so that it can use behavioral characteristics as a pattern, not just data characteristics; thus, the NIDS is further improved [23].

However, because thresholds and patterns are created based on information from prior attacks, an attacker can estimate thresholds and patterns in reverse. An attacker can also modify the attack pattern so that it cannot be detected by exploiting the estimated result. Here, network administrators should steadily collect information regarding variant attacks and add new thresholds and patterns. Consequently, the administrator is overwhelmed, resulting in a sluggish response to new attacks.

2.2. Early ML-Based NIDS

Various traditional ML models have been applied to NIDS to address the high management cost and slow response time of NIDS, which are used to detect intrusion based on existing thresholds or patterns. Detecting intrusions using thresholds or patterns can be slightly vulnerable to detection bypass attacks; therefore, a ML-based NIDS detects intrusions using overall characteristics rather than some characteristics of intrusive traffic [23–25]. For example, traffic is divided into sessions, and various statistical characteristics are extracted for each session and used to distinguish between intrusive and normal traffic. Because a ML-based NIDS uses approximately 20–90 features, it comprehensively uses the overall characteristics of the session to detect intrusion [24–26]. Hence, the detection efficiency is not affected by only a few parameters, e.g., the threshold or pattern, rendering it unfeasible for attackers to circumvent a ML-based NIDS.

In the early days, many simple decision tree algorithms were applied, and high-speed classifier algorithms, such as J48, C4.5, and C5.0, were selected because NIDS required a fast classification speed to handle large amounts of traffic [27,28]. NIDS and firewalls require a packet classification process. In contrast to routing or switching, which is classified based on one existing field, packet classification in a NIDS or firewall is performed using five tuples. Thus, the packet classification process was implemented using a sophisticated high-speed algorithm. The high-speed packet classification algorithm is similar to the decision tree-based algorithm with a field number of five. Therefore, the decision tree algorithm can be easily implemented using high-speed packet classification algorithms; thus, the decision tree algorithms are most suitable for early ML-based NIDS.

As the CPU/GPU hardware platform using multiple threads was applied in the NIDS, it even used RF to apply multiple decision trees. RF is a superior alternative for increasing the performance of NIDS, as it can prevent the overfitting of decision trees and increase classification accuracy. In addition to RF, attempts have been made to increase classification performance by applying the ensemble technique to various MLs. Most studies have been conducted to increase classification accuracy rather than classification speed.

2.3. Advance DL-Based NIDS

Owing to the successful application of early ML-based NIDS, it has been expanded and developed into a technology that detects anomalies using simple network traffic and user behavior characteristics. The problem is that existing MLs have limitations in improving the detection rate of NIDS, and feature engineering is still required; therefore, it is not possible to completely eliminate user intervention. To solve this problem, many researches have been conducted to apply deep learning models such as DNN, LSTM, and CNN to NIDSs instead of the initial ML model. Deep learning models not only require more computation and memory than existing ML models, but also have difficulty improving detection performance due to the large differences in terms of datasets. Therefore, the

purpose of most researches was to reduce the complexity of deep learning models while increasing the detection accuracy of NIDS [29–35].

DL models for computer vision, which have been actively researched in the vision field, achieved improved classification accuracy over existing ML models, prompting the use of DL models in ML-based network intrusion detection systems. DL models, such as CNN, are known to be one of the most promising DL models because they exhibit extremely high classification accuracy by self-extracting important features for numerous features [14]. DL models for computer vision are designed to classify images, a data type consisting of pixels, rendering them unsuitable for NIDS datasets. NIDS and image datasets differ in their characteristics in the following ways:

- The NIDS dataset has an extremely small feature size compared with the image dataset.
- Image data are represented as a dense matrix, but NIDS data are represented as a sparse one.
- The data in the image dataset consist only of the same data called pixels, whereas the NIDS data consist of different types of data.
- The two pixels in the image have a higher correlation as the distance decreases, while the two features in the NIDS data have independent properties regardless of the distance.

Eventually, DL models optimized for 2D images can only be applied to the NIDS dataset after being modified to suit 1D data processing [27]. Many related studies have been conducted because even these degraded DL models exhibited a higher performance than the existing ML-based NIDS. Recently, however, a novel approach was proposed to maximize the performance of DL models for computer vision by converting non-image data into 2D images and using DL models for computer vision without modification [13].

Although there is some loss of information in converting existing non-image data into image data, the proposed method shows further improvement over degraded DL models, as DL models for computer vision perform best in 2D images. Various algorithms, such as LDA, UMAP, t-SNE, and PCA, exist in dimension reduction techniques used to convert NIDS datasets into 2D images. In previous studies, there was a limit to improving performance because only one algorithm was applied and converted into an image. Consequently, research into various 2D image conversion methods must be undertaken to enable the application of DL technologies for highly advanced computer vision to NIDS. Through this, a new NIDS dataset conversion algorithm optimized for DL model for computer vision must be developed.

3. Proposed Algorithm

Dataset D for a NIDS consisting of N data points are defined as follows: $D = \{D_1, D_2, \ldots, D_N\}$ where $D_k \in \mathbb{R}^n$. The proposed image transformation algorithm $F : D \rightarrow V$, where $V = \{V_1, V_2, \ldots, V_N\}$ and $V_k \in \mathbb{R}^2$. It converts the dataset into a 2D image, but not a grayscale image, and to a three-channel RGB color image. The proposed algorithm generates one grayscale image using each dimensionality reduction algorithm, analyzes the quality of every generated image, selects the best three images, and integrates them to create one RGB color image. Let us now explain each step in detail.

3.1. Image Transformation

The dimensionality reduction algorithms used are t-SNE, UMAP, PCA, PCA with RBF kernel, i.e., kPCA(rbf), PCA with cosine kernel, i.e., kPCA(cosine), PCA with the polynomial kernel, i.e., kPCA(polynomial), and PCA with the sigmoid kernel, i.e., kPCA(sigmoid). Each dimensionality reduction algorithm maps the entire NIDS dataset onto a 2D space. Subsequently, it determines the smallest rectangle containing the transformed 2D data before rotating and cropping it. The data converted in this manner are normalized to a predefined size, regardless of the algorithm. Finally, data consisting of 80 features in the dataset are converted into a $120 \times 120 \times 1$ image. Based on this result, the conversion function F_p is generated, where $p \in \{t\text{-}SNE, UMAP, PCA, kPCA(rbf), kPCA(cosine), kPCA(polynomial), PCA(sigmoid)\}$.

Let F_p be an image transformer [13]. Figure 1 shows how the existing NIDS dataset is converted into a dataset composed of images from each image transformer.

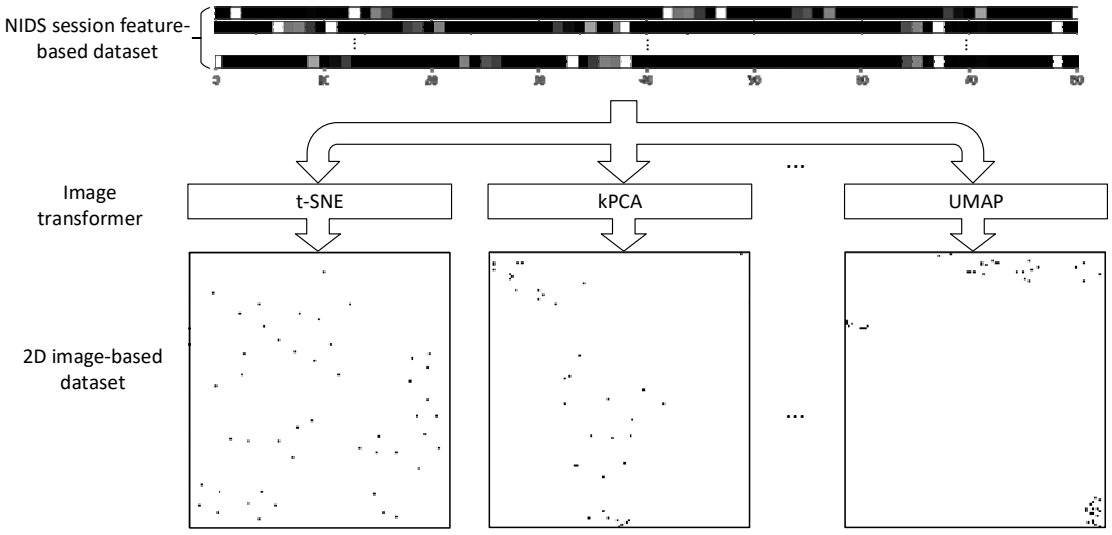


Figure 1. Results of converting NIDS datasets into 2D image-based datasets using each image transformer.

3.2. Multiple Image Transformer-Based Algorithms

The proposed scheme generates V_p with $F_p(D)$ for each algorithm p, trains a CNN using V_p, and calculates the F1-score for training data. Based on the F1-score, the three algorithms with the highest scores are selected. Subsequently, the grayscale image of each selected algorithm is mapped to each channel of the three-channel RGB image. Figure 2 shows the synthesis of an RGB image using the results of the selected image transformers. Figure 3 shows the final RGB image built from the results shown in Figure 1.

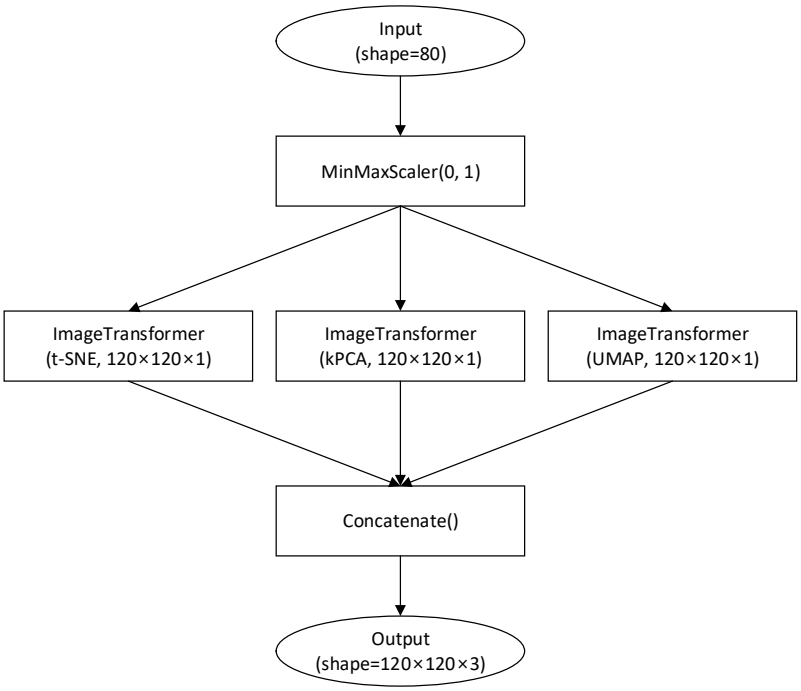


Figure 2. Process of generating a 3-channel RGB color image when t-SNE, kPCA(cosine), or UMAP are selected as the image transformer with the highest F1-score values.

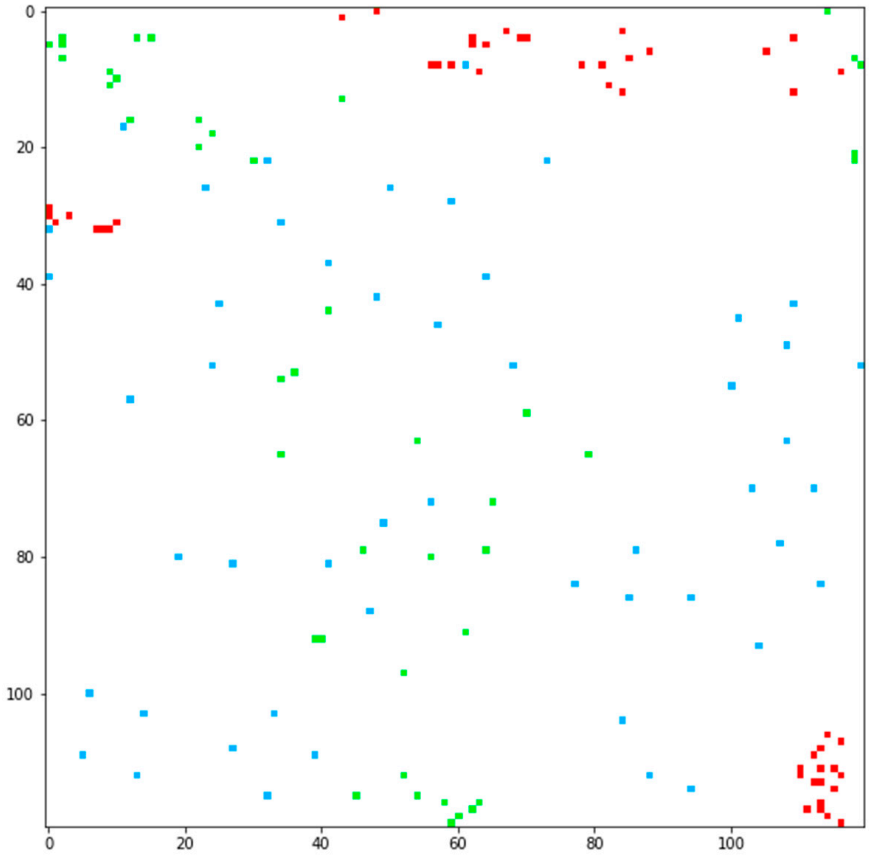


Figure 3. Final 3-channel RGB color image created using grayscale images generated by t-SNE, kPCA (cosine), and UMAP, which are the image transformers with the highest F1-score values.

3.3. 2D Image-Based DL Classifier

The generated image is used to train the four-layer dual CNN model shown in Figure 4. This CNN model has two four-layer CNN models with different filter sizes in parallel to improve intrusion detection performance. The selected filter sizes are 3×3 and 5×5, and for each layer of the CNN model, the padding option is set to be valid, and ReLU is applied as an activation function. MaxPooling is constantly applied to the convolution layer output. The outputs of the two four-layer CNNs applied Global Average Pooling, and are combined through the concatenate layer, and the final classification is performed through the fully connected network.

It is a well-known fact through research on existing inception network that exploiting various kernel sizes in CNN is of great help in improving classification performance [inception]. However, since inception network is far more complicated and slower than other deep learning models such as CNN and RNN, it is difficult to apply them to high performance NIDSs. Therefore, in order to minimize the degradation of the classification speed while having the advantage of multiple kernel sizes, the proposed method uses only two kernel sizes, concatenates the results, and uses them for final classification.

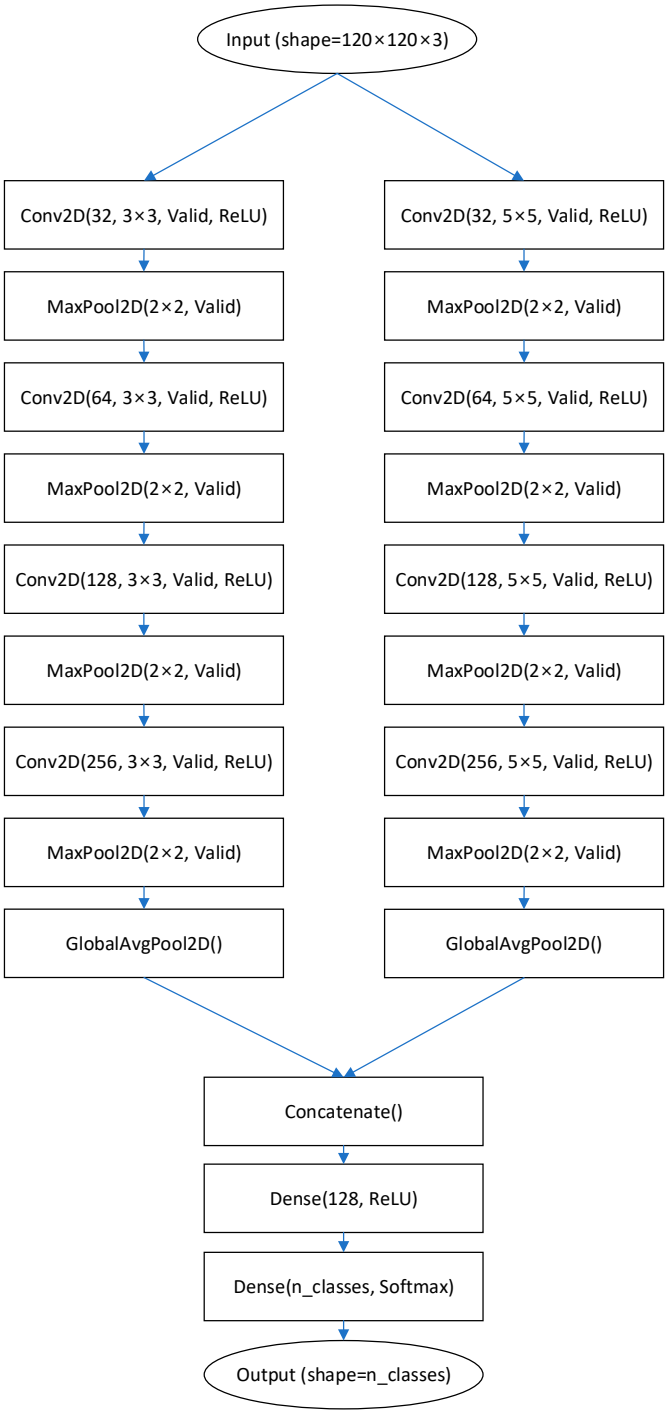


Figure 4. Structure of 4-layer dual CNN classifier.

4. Performance Evaluation

4.1. Evaluation Environment

To accurately evaluate the performance of the proposed algorithm, it is compared with two existing methods: one is the latest method that classifies each session by deep learning using session feature without image conversion (labeled as 'Non-image'), and the other is the method that uses only one channel image (labeled as '1 Channel') [13,36,37]. To evaluate the performance in various environments, the ISCIDS2012 and CSE-CIC-IDS2018 datasets were used, and each dataset was divided into training, verification, and test datasets at a size of 3:1:1. Detailed information about each dataset are as follows:

4.1.1. ISCXIDS2012 Dataset [25]

This dataset consisted of four intrusion classes and one normal class. Table 1 lists the names and sizes of each class in the dataset.

Table 1. Each class size in ISCXIDS2012 dataset.

Label	Training Dataset	Validation Dataset	Testing Dataset
BruteForceSSH	3000	1000	1000
DDoS	36,000	12,000	12,000
HTTPDoS	1500	500	500
Infiltration	2700	900	900
Normal	60,000	20,000	20,000
Total	103,200	34,400	34,400

4.1.2. CSE-CIC-IDS2018 Dataset [38]

The dataset originally had 12 classes of attacks and one class of normal attacks. However, because the amount of data for the BruteForce-XSS and SQL-Injection classes are extremely small, only 10 intrusion classes and one benign class were used for performance evaluation. Table 2 lists the names and sizes of each class used in the evaluation.

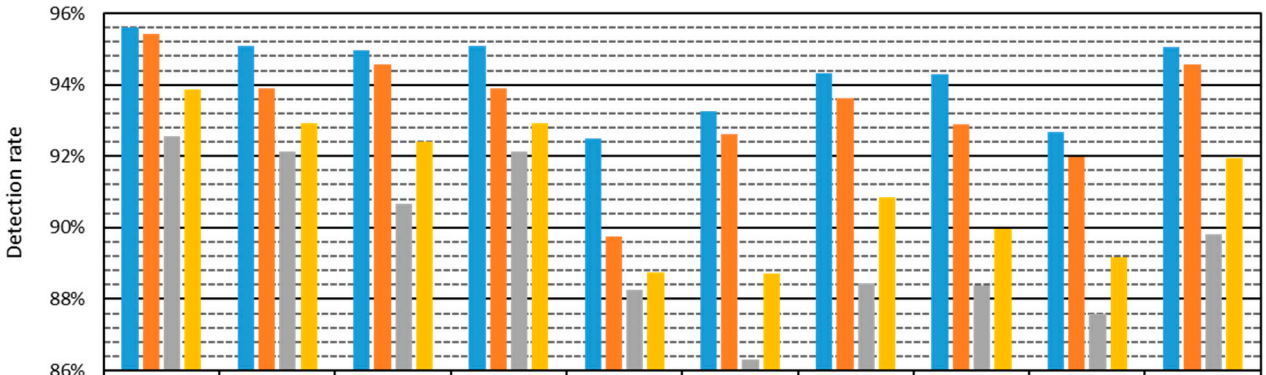
Table 2. Each class size in CSE-CIC-IDS2018.

Label	Training Dataset	Validation Dataset	Testing Dataset
Benign	30,000	10,000	10,000
Bot	6000	2000	2000
BruteForce-FTP	5400	1800	1800
BruteForce-SSH	5400	1800	1800
BruteForce-WEB	360	120	120
DDoS-HOIC	7200	2400	2400
DDoS-LOIC-HTTP	7200	2400	2400
DoS-GoldenEye	2400	800	800
DoS-Hulk	7200	2400	2400
DoS-SlowHTTPTest	5400	1800	1800
DoS-Slowloris	1500	500	500
Total	78,060	26,020	26,020

4.2. Detection Rate

To accurately compare the detection rate for each algorithm, the results of all image transformers and the existing '1 channel' algorithm and 'Non-image' algorithm were measured. Figure 5 shows the measurement results of the detection rates for each algorithm for the ISCXIDS2012 dataset. Since the 'Non-Image' algorithm has higher precision than the '1 channel' algorithm, the probability of classifying a normal session as an intrusion is low. The '1 channel' algorithm has a higher recall value than the 'Non-image' algorithm and is superior in detecting actual intrusions. The proposed algorithm achieves the highest performance for all metrics such as accuracy, precision, recall, and F1-score compared to all image transformers including the 'Non-image' algorithm and the '1 channel' algorithm. Figure 5 shows that the performance of image transformers varies considerably, and that the overall performance is mostly inferior to that of the '1 channel' and 'Non-image' algorithms. However, the proposed algorithm achieved the highest performance when using these

image transformers. The proposed algorithm significantly improves the detection accuracy by integrating the results of the three best image transformers.

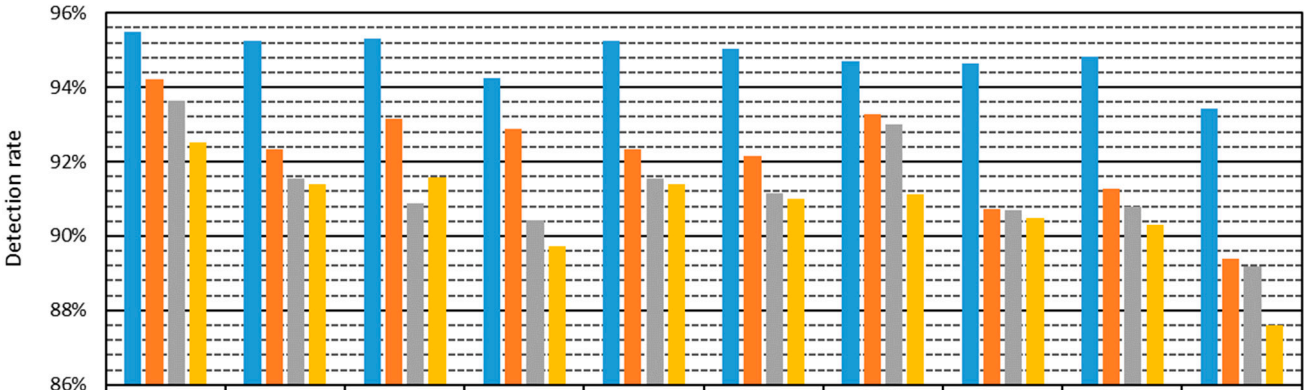


	Proposed	1 Channel	Non-image	t-SNE	PCA	kPCA (RBF)	kPCA (Cosine)	kPCA (Polynomial)	kPCA (Sigmoid)	UMAP
Accuracy	95.60%	95.08%	94.95%	95.08%	92.48%	93.24%	94.34%	94.31%	92.68%	95.07%
Precision	95.41%	93.90%	94.56%	93.90%	89.74%	92.62%	93.63%	92.90%	91.96%	94.57%
Recall	92.55%	92.12%	90.67%	92.12%	88.25%	86.30%	88.42%	88.38%	87.60%	89.82%
F1-Score	93.86%	92.91%	92.41%	92.91%	88.73%	88.73%	90.85%	89.97%	89.18%	91.95%

Figure 5. Detection rate for each algorithm on the ISCXIDS2012 dataset.

It can be confirmed that the RGB color image used by the proposed method is significantly more effective than the grayscale image through the results using the three-channel image and the one-channel image. In addition, when generating a color image, the selection algorithm that determines an image for each channel plays a major role in generating a good quality image.

Figure 6 shows the detection performance for the CSE-CIC-IDS2018 dataset and a trend similar to that of the ISCXIDS2012 dataset. However, noting that the 'Non-image' algorithm has a higher F1-score than the '1 channel' algorithm, it cannot be guaranteed that the image transformer improves intrusion detection performance compared to the existing method using session features. However, the proposed algorithm still showed the best performance compared to the '1 channel' and 'Non-image' algorithms. Therefore, it is concluded that the proposed algorithm exhibits the highest performance compared with competing algorithms in various environments.



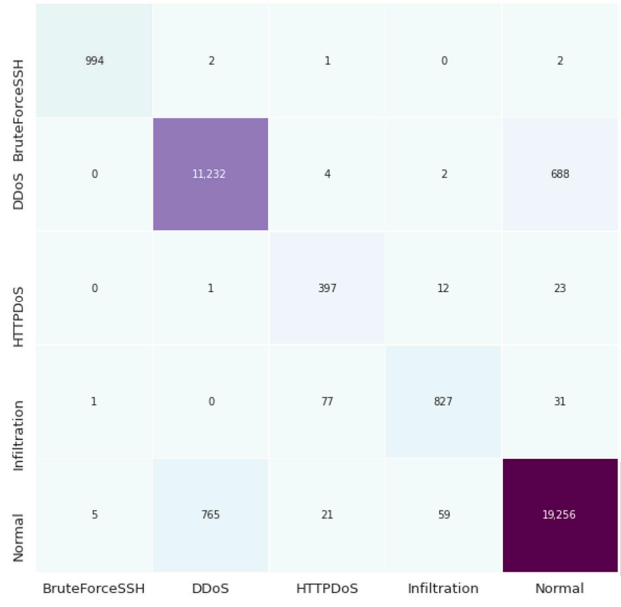
	Proposed	1 Channel	Non-image	t-SNE	PCA	kPCA (RBF)	kPCA (Cosine)	kPCA (Polynomial)	kPCA (Sigmoid)	UMAP
Accuracy	95.50%	95.26%	95.30%	94.23%	95.26%	95.02%	94.70%	94.65%	94.81%	93.42%
Precision	94.22%	92.34%	93.16%	92.88%	92.34%	92.14%	93.26%	90.73%	91.26%	89.38%
Recall	93.63%	91.54%	90.87%	90.43%	91.54%	91.16%	93.01%	90.69%	90.78%	89.19%
F1-Score	92.52%	91.40%	91.59%	89.73%	91.40%	91.00%	91.11%	90.50%	90.31%	87.60%

Figure 6. Detection rate for each algorithm on the CSE-CIC-IDS2018 dataset.

4.3. Confusion Matrix

Figures 7 and 8 show the confusion matrix for the proposed algorithm and the '1 channel' and 'Non-image' algorithms. As mentioned above, the overall performance of the proposed method was confirmed to be superior to those of the two existing algorithms. Figures 7 and 8 also show that the detection rate of the proposed algorithm is higher than or at least similar to that of the existing algorithms for most classes. For example, the proposed algorithm has higher or similar detection rates in all classes except infiltration for ISCXIDS2012 and DoS-SlowHttpTest for CSE-CIC-IDS2018 compared with the '1 channel' algorithm. Furthermore, the proposed algorithm has a high detection rate in all cases except for the DoS-Slowloris of CSE-CIC-IDS2018.

(**a**) Proposed



(**b**) 1 channel

(**c**) Non-image

Figure 7. ISCIDS2012 Confusion matrix. Each row shows predicted result for each class.

(**a**) Proposed

(**b**) 1 channel

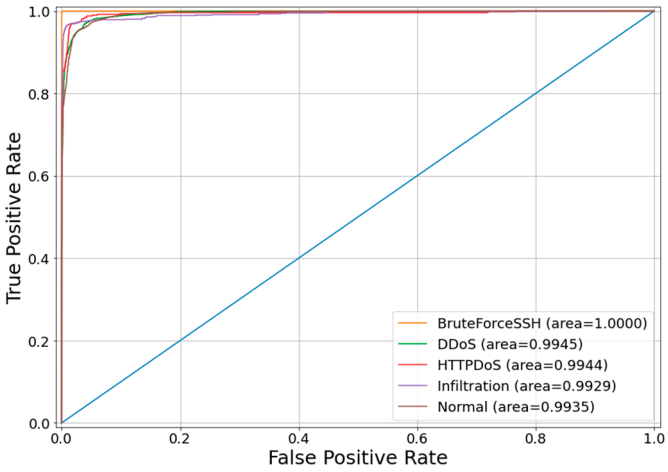
(**c**) Non-image

Figure 8. CSE-CIC-IDS2018 Confusion matrix. Each row shows predicted result for each class.

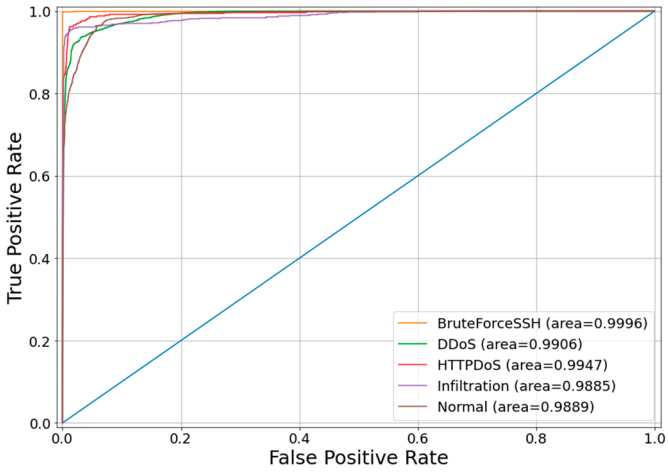
From the confusion matrix, it can be seen that the proposed method not only has the highest overall accuracy, but also the highest accuracy for individual classes, especially the normal class. If a normal class is falsely detected as an intrusion, the corresponding normal service is greatly hindered, causing great inconvenience to users. Therefore, the detection accuracy for the normal class is more important than for other classes. In this aspect, the proposed method also has a very good characteristic compared to other algorithms.

4.4. ROC Curve

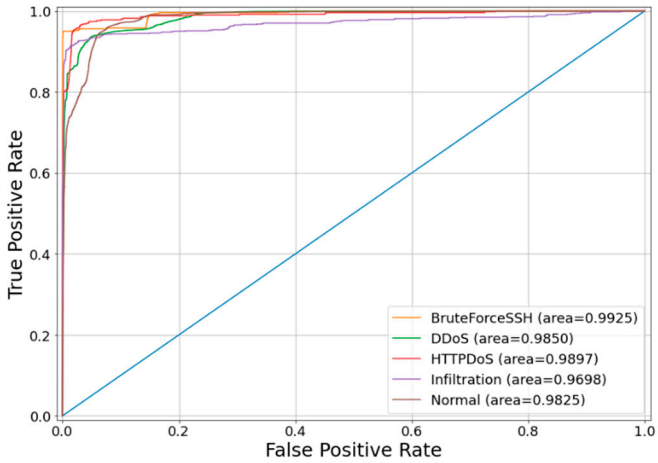
Figures 9 and 10 show the ROC curves for each algorithm. As shown in the detection rate and confusion matrix described above, it is confirmed that the performance of each class of the proposed algorithm is superior or similar to that of the existing algorithms.



(**a**) Proposed

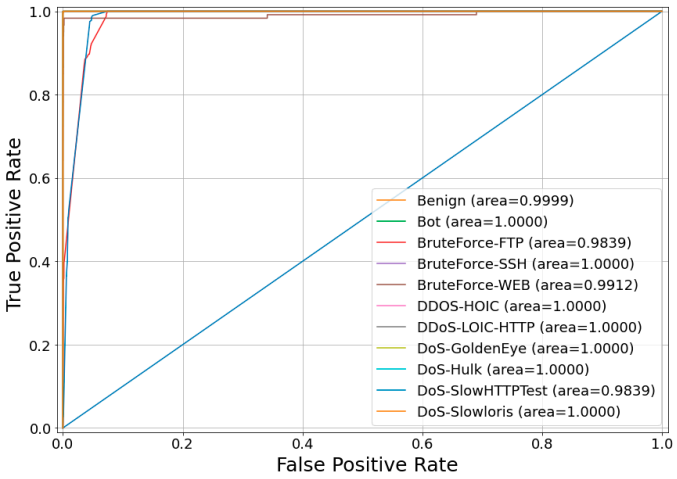


(**b**) 1 channel

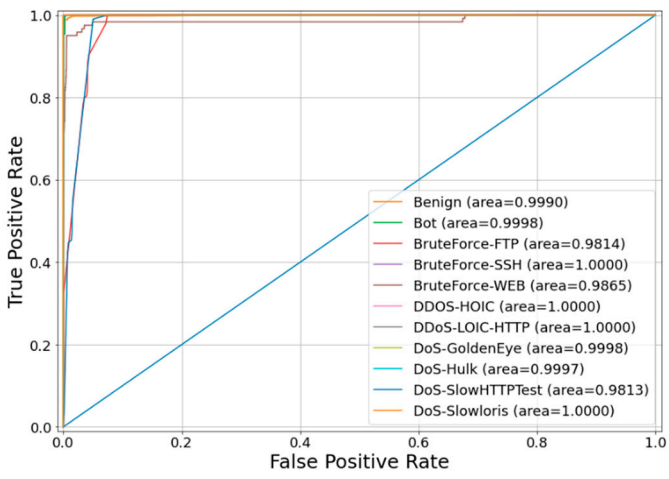


(**c**) Non-image

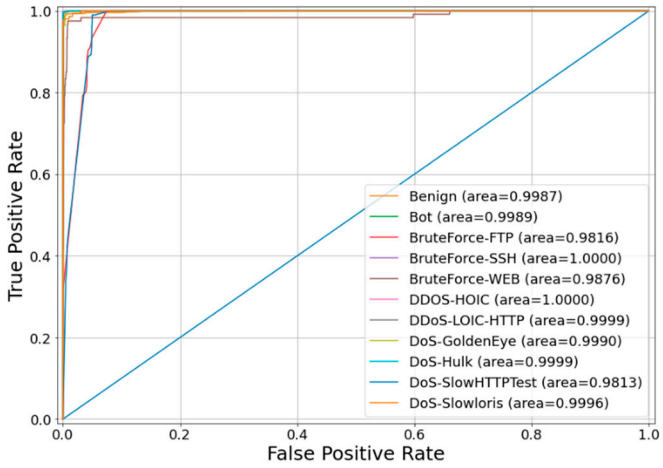
Figure 9. ISCXIDS2012 ROC curves.



(**a**) Proposed



(**b**) 1 channel



(**c**) Non-image

Figure 10. CSE-CIC-IDS2018 ROC curves.

5. Conclusions

Recently, DL models have been introduced to improve NIDS detection performance in NIDS research. However, because the NIDS dataset is not composed of image-based data, there is a technical limit to applying modern DL models that have shown high performance in image classification to NIDS. Although the existing method converts it into grayscale images, the proposed algorithm converts the NIDS dataset into three-channel RGB color images, allowing DL models for existing image classification to be applied to NIDS without performance degradation owing to non-image data or grayscale image data. This has a very high detection performance, which is difficult to achieve using the existing methods. Currently, threats to network security are steadily increasing, and as zero-day attacks spread, simple rule-based or pattern-based NIDS can no longer safely protect networks from cyber threats. We expect that the proposed algorithm will be instrumental in introducing a DL model to the NIDS domain to surmount existing limitations.

However, there are problems to be solved regarding the proposed algorithm before it can be deployed in real networks. Converting NIDS features to a three-channel RGB color image has a higher time complexity than a one-channel grayscale image. This causes a limitation for the proposed NIDS to handle high-capacity network traffic. First, additional research on software optimization techniques, along with hardware structures such as

parallel processing structures to improve the proposed NIDS throughput, are needed. Second, features from a three-channel RGB color image require higher memory usage than features from a one-channel or non-image. Memory requirement is highly correlated with image size, so research on the optimal size of image through analysis of image size and performance is also necessary. Through these additional studies, we hope that the proposed algorithm will become a more stable and efficient one.

Author Contributions: T.K. and W.P. wrote the paper and conducted the research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF), NRF2022R1A2C1011774.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets utilized in this paper are ISCXIDS2012 dataset (https://www.unb.ca/cic/datasets/ids.html (accessed on 17 February 2023)) and CSE-CIC-IDS2018 dataset (https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 17 February 2023)).

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

References

1. Faizal, M.A.; Mohd Zaki, M.; Shahrin, S.; Robiah, Y.; Siti Rahayu, S.; Nazrulazhar, B. Threshold Verification Technique for Network Intrusion Detection System. *arXiv* **2009**. [CrossRef]
2. Chunyue, Z.; Yun, L.; Hongke, Z. A Pattern matching based Network Intrusion Detection System. In Proceedings of the 2006 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5–8 December 2006; pp. 1–4. [CrossRef]
3. Bilge, L.; Dumitras, T. Before we knew it: An empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 833–844. [CrossRef]
4. Kruegel, C.; Toth, T. Using decision trees to improve signature-based intrusion detection. In Proceedings of the 2003 International Workshop on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 8–10 September 2003; pp. 173–191. [CrossRef]
5. Sahu, S.; Mehtre, B.M. Network intrusion detection system using J48 Decision Tree. In Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 August 2015; pp. 2023–2026. [CrossRef]
6. Ho, T.K. Random Decision Forests. In Proceedings of the 2015 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; pp. 278–282.
7. Ho, T.K. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844. [CrossRef]
8. Freund, Y.; Schapire, R.E. A decision-theoretic [sic] generalization of on-line learning and an application to boosting. *Lect. Notes Comput. Sci.* **1995**, *904*, 23–37. [CrossRef]
9. Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; Al-Sabahi, K. Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection. *IEEE Access* **2018**, *6*, 52843–52856. [CrossRef]
10. Gu, J.; Zhu, M.; Zhou, Z.; Zhang, F.; Lin, Z.; Zhang, Q.; Breternitz, M. Implementation and evaluation of deep neural networks (DNN) on mainstream heterogeneous systems. In Proceedings of the 2014 5th Asia-Pacific Workshop on Systems (APSys '14), Beijing, China, 25–26 June 2014; ACM: New York, NY, USA, 2014; pp. 1–7. [CrossRef]
11. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [CrossRef] [PubMed]
12. van der Maaten, L.J.P.; Hinton, G.E. Visualizing Data Using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
13. Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.A.; Tsunoda, T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **2019**, *9*, 11399. [CrossRef] [PubMed]
14. Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Math. Comput. Simul.* **2020**, *177*, 232–243. [CrossRef]
15. Schölkopf, B.; Smola, A.; Müller, K.-R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comput.* **1998**, *10*, 1299–1319. [CrossRef]
16. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2018**. [CrossRef]
17. Broomhead, D.H.; Lowe, D. Multivariable Functional Interpolation and Adaptive Networks. *Complex Syst.* **1998**, *2*, 321–355.
18. Sun, F.X. Danger Theory Based Risk Evaluation Model for Smurf Attacks. *Key Eng. Mater.* **2011**, *467–469*, 515–521. [CrossRef]
19. Erikson, J. *Hacking the Art of Exploitation*, 2nd ed.; NoStarch Press: San Francisco, CA, USA, 1997; p. 264.

20. Dharmapurikarg, S.; Krishnamurthy, P.; Sproull, T.; Lockwood, J. Deep packet inspection using parallel bloom filters. In Proceedings of the 2003 11th Symposium on High Performance Interconnects, Stanford, CA, USA, 20–22 August 2003.
21. Symantec Internet Security Threat Report: Trends for July–December 2007. Symantec Corporation. Available online: https://docs.broadcom.com/doc/istr-08-april-exec-sum-en (accessed on 17 February 2023).
22. Aho, A.V.; Corasick, M.J. Efficient string matching: An aid to bibliographic search. *Commun. ACM* **1975**, *18*, 333–340. [CrossRef]
23. Malek, Z.S.; Trivedi, B.; Shah, A. User behavior Pattern -Signature based Intrusion Detection. In Proceedings of the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 27–28 July 2020; pp. 549–552. [CrossRef]
24. Li, L.; Yu, Y.; Bai, S.; Hou, Y.; Chen, X. An Effective Two-Step Intrusion Detection Approach Based on Binary Classification and *k*-NN. *IEEE Access* **2017**, *6*, 12060–12073. [CrossRef]
25. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009. [CrossRef]
26. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ali, A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
27. Description of Kyoto University Benchmark Data. Available online: https://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf (accessed on 13 January 2023).
28. Quinlan, R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
29. Quinlan, R. Data Mining Tools See5 and c5. Available online: https://www.researchgate.net/publication/307174962_Data_Mining_Tools_See5_and_C50 (accessed on 17 February 2023).
30. Lu, C. Research on the technical application of artificial intelligence in network intrusion detection system. In Proceedings of the 2022 International Conference on Electronics and Devices, Computational Science (ICEDCS), Marseille, France, 22–24 September 2022. [CrossRef]
31. Wang, J.D.; He, A.; Castiglione, B.; Gupta, B.; Karuppiah, M.; Wu, L. PCNNCEC: Efficient and Privacy-Preserving Convolutional Neural Network Inference Based on Cloud-Edge-Client Collaboration. *IEEE Trans. Netw. Sci. Eng.* **2022**. [CrossRef]
32. Shi, Z.; Chehade, A. A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2021**, *205*, 107257. [CrossRef]
33. Li, X.Y.; Tang, R.; Song, W. Intrusion Detection System Using Improved Convolution Neural Network. In Proceedings of the 2022 11th International Conference of Information and Communication Technology (ICTech), Wuhan, China, 4–6 February 2022. [CrossRef]
34. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2017**, *6*, 1792–1806. [CrossRef]
35. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
36. Khan, A.; Cotton, C. Detecting Attacks on IoT Devices using Featureless 1D-CNN. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 27–29 July 2021. [CrossRef]
37. Navya, V.K.; Adithi, J.; Rudrawal, D.; Tailor, H.; James, N. Intrusion Detection System using Deep Neural Networks (DNN). In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–6. [CrossRef]
38. Soheily-Khah, S.; Marteau, P.; Béchet, N. Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset. In Proceedings of the 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 8–10 April 2018. [CrossRef]