information

*Article*

# Image-Based Approach to Intrusion Detection in Cyber-Physical Objects

**Sergey Golubev, Evgenia Novikova * and Elena Fedorchenko ***

Saint Petersburg Institute for Informatics and Automation, Federal Research Center of the Russian Academy of Sciences, 199178 Saint Petersburg, Russia
* Correspondence: novikova@comsec.spb.ru (E.N.); doynikova@comsec.spb.ru (E.F.)

**Abstract:** Recently, approaches based on the transformation of tabular data into images have gained a lot of scientific attention. This is explained by the fact that convolutional neural networks (CNNs) have shown good results in computer vision and other image-based classification tasks. Transformation of features without spatial relations to images allows the application of deep neural networks to a wide range of analysis tasks. This paper analyzes existing approaches to feature transformation based on the conversion of the features of network traffic into images and discusses their advantages and disadvantages. The authors also propose an approach to the transformation of raw network packets into images and analyze its efficiency in the task of network attack detection in a cyber-physical object, including its robustness to novel and unseen attacks.

**Keywords:** intrusion detection; network traffic; image-based features; grayscale image; convolutional neural network

check for updates

**Citation:** Golubev, S.; Novikova, E.; Fedorchenko, E. Image-Based Approach to Intrusion Detection in Cyber-Physical Objects. *Information* **2022**, *13*, 553. https://doi.org/10.3390/info13120553

Academic Editor: Beng Soo Ong, Tianchong Wang and Eric C. K. Cheng

Received: 3 October 2022
Accepted: 14 November 2022
Published: 25 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



## 1. Introduction

The wide application of deep neural networks is explained by their ability to reveal hidden non-linear relations between the analyzed attributes. They have turned out to be extremely efficient in such tasks as computer vision, and speech recognition, i.e., tasks where parameters have explicit spatial relations. Currently, there are many studies devoted to the problem of transformation of tabular data without such relations into images in order to apply deep neural models and take advantage of pre-trained models [1–3].

The input data in network intrusion detection tasks are usually represented by a vector of numerical values. There are only a few approaches that use image-based features to detect network attacks [4–6], and all of them use the transformation of tabular data, such as statistics on network flows extracted from PCAP (Packet Capture) files. To the best of the authors' knowledge, there are no approaches that are based on direct conversion of raw binary packets into images [7], though when used as a data preprocessing step, this transformation has proved its efficiency in cyber security tasks, such as malware detection and authorship attribution [8–11]. Moreover, Alrabaee et al. [8] and Kaur et al. [9] demonstrated that analysis models trained on such images are able to detect the authors of a code, even if different techniques that could mask authorship, such as code obfuscation or compiler optimization settings are applied. Rong et al. [10] used the conversion of raw session network traffic into fixed-size RGB images as a preprocessing step, for training a deep transfer learning model on the basis of ResNet-50 [12] in order to detect unseen malware samples.

These results motivated the authors to analyze the conversion of raw network packets into images as a data preprocessing step in the process of network attack detection, and to evaluate the performance of deep neural networks that are trained on this type of dataset. The results of the research were first presented at the 15th International Symposium on Intelligent Distributed Computing (IDC 2022), where the authors introduced the approach

and primary experimental results [13]. This paper provides an extended analysis of related research and a detailed description of the approach, including the key challenges that should be addressed when constructing images from raw network packets. It also presents an evaluation methodology for assessing the robustness of the trained classifier to unseen attacks.

Thus, the authors' *contributions* are as follows:

- An approach to the preprocessing of the network data in PCAP format into images;
- An evaluation of the impacts of different settings of the image generation procedure on the efficiency of the analysis models, namely, ResNet34 and MobileNetV3-small, that are used in the feature extraction mode, as well as the CNN constructed by the authors.

The *novelty* of this research consists in a novel approach to data preprocessing that is based on a transformation of raw packets into images, which allows the achievement of high accuracy in attack detection.

This paper is structured as follows. Section 2 reviews approaches to intrusion detection, including ones that use the transformation of data into images. Section 3 describes the proposed approach to attack detection with a particular focus on the image generation step. Section 4 details the methodology of the experiments and discusses the obtained results. Section 5 summarizes the obtained results and defines the direction of future work.

## 2. Related Work

Cyber intrusion detection is a highly relevant area today, and it continues to develop. Over recent years, researchers have studied and proposed various approaches to the detection of known types of cyber attacks; these approaches have ranged from knowledge-based methods (signature-based methods, scenario description languages, finite-state machines, Petri nets, expert systems, and model checking [14]) to machine-learning-based methods (decision trees [15], support vector machines [16], Bayesian networks [17], Bayesian methods [18], multivariate adaptive regression (MAR) splines [19], clustering algorithms [20], and regression algorithms [21]) and other advanced computational intelligence methods (neural networks [22,23], genetic algorithms [24], fuzzy logic [25], immune systems [26], and swarm intelligence [7]). For the detection of anomalies (unknown types of cyber intrusions), researchers have also proposed numerous techniques based on wavelet analysis [27], statistical analysis [28], entropy analysis [29], spectral analysis, fractal analysis [30], and cluster analysis [7,31].

The proposed methods have demonstrated good results for the detection of known types of cyber intrusions, but are still limited in the detection of unknown types of attacks (anomalies). Anomaly detection methods use input data, such as network traffic or event logs, for feature extraction, and they construct the normal behavior profile on that basis. This profile is then compared with new activity profiles to detect anomalies. New approaches to feature extraction have been proposed as some of possible ways to overcome limitations in the detection of unknown types of intrusions. Thus, the transformation of features into images as a feature extraction procedure has demonstrated promising results in various object detection tasks in other areas, such as genetics.

In [3], the authors proposed an algorithm called DeepInsight, which converts various non-image data into images, and they used this to train a CNN to differentiate among phenotypes or categories. They tested different kinds of non-image datasets, including RNA-seq, vowel, text, and artificial datasets and they obtained promising results.

In [32], the authors tested their approach in drug sensitivity prediction scenarios by using synthetic and pharmacological datasets. They researched the limitations of CNNs in relation to predictive modeling. The authors proposed a novel feature representation approach called REFINED (representation of features as images with neighborhood dependencies). The idea consisted of the transformation of high-dimensional vectors into images, which were then used for CNN deep learning. The peculiarity of this approach was its use of embedded feature extraction. The authors generated a concise feature map in the form of a two-dimensional image using a Bayesian metric multi-dimensional scaling

approach in order to minimize the pairwise distance values and, thus, to consider the similarities between features. The experiments demonstrated that obtained REFINED CNN outperforms such commonly used approaches as artificial neural networks, random forests, support vector machines, elastic nets, and linear regressions, and such state-of-the-art methods as Deep–Resp–Forest [33] and heterogeneous graph networks [34] on a synthetic dataset, NCI60 drug response dataset [35] , and heterogeneous Genomics of Drug Sensitivity in Cancer (GDSC) dataset [36] in terms of predictive accuracy, statistical significance, and robustness.

In [37], the authors transform tabular data into images to predict anti-cancer drug response. Their algorithm, image generator for tabular data (IGTD), assigns features to pixel positions to locate similar features close to each other in the image minimizing the distance between the pixels that correspond to the features with minimum distance between them. The authors used the Cancer Therapeutics Response Portal v2 (CTRP) [38] and the Genomics of Drug Sensitivity in Cancer (GDSC) (https://www.cancerrxgene.org/, accessed on 1 October 2022) datasets to demonstrate on the experiments that CNNs trained using the output images of their algorithm outperform CNNs trained using another image representations and prediction models trained using the original tabular data in predicting anti-cancer drug response.

Thus, the transformation of the network data (PCAP packets) to image as a feature extraction procedure looks promising. It usually incorporates the following stages:

1. Extraction of the numerical and nominal features from the PCAP packets.
2. Transformation of the numerical and nominal features to image.

There are several research papers that implement this procedure to generate image-based features and further use obtained images to learn convolutional neural networks (CNN) for intrusion detection tasks [4–6,39–41].

Thus, in [4] authors use a convolutional neural network pre-trained model VGG-16 [42]. They transform 41 network features from NSL-KDD data [43] as follows: (1) normalize features; (2) extend their number from 41 to 121 to generate a grayscale image with size $11 \times 11$; (3) duplicate the single color channel of the generated image for each color channel and reshape to $224 \times 224 \times 3$ because VGG-16 uses RGB image as input. The authors obtained anomaly detection accuracy of 89.30% for KDDTest+ and 81.77% for KDDTest-21 in case of binary classification task.

In [44], the authors also use Visual Geometry Group pre-trained model (VGG-19). After that they use a hybrid deep neural network based on CNN and long short-term memory (LSTM) to extract features from network traffic. The final VGG-19 + Hybrid CNN-LSTM model allows obtaining an accuracy of 98.86% while classifying attacks within the network intrusion benchmark dataset.

In [5], the authors use the MobileNetV2 convolutional neural network model [45] to detect attacks in binary and multi-class modes. They transform features extracted from the UNSW-NB15 dataset [46], containing labeled PCAP packets, into a $16 \times 16$ grayscale image as follows: each pixel corresponds to some feature value. For example pixel with coordinates (3, 13) is set to 255 if the packet uses HTTP protocol. The authors obtained the trained model's accuracy 97%. The best accuracy was obtained for Generic, Fuzzers types of attacks and normal traffic. In [6], the authors trained their own CNN model on the NSW-NB15 dataset first and then used the pre-trained model to detect attacks in the NSL-KDD dataset. They also converted network features into images. The authors obtained up to 99.82% of detection rate on the KDDTest-21 dataset.

The research [47] deserves attention as the authors applied another approach to image generation—they transform texts (HTTP messages) to images on the character level: each character was represented by a pixel. After that, they trained a convolutional auto-encoder to detect anomalies in HTTP messages. Through the experiments, the authors demonstrated that the suggested approach outperforms traditional unsupervised methods, such as isolation forest and one-class support vector machine, in the anomaly detection task.

The analysis of the related works showed that transformation of non-image data into images with further training of a CNN has the following essential advantages: finding hidden relationships in attributes and their values; robustness in unseen objects detection (it is essential to detect unknown attacks); extracting basic knowledge from limited datasets [13].

Though researchers from different subject domains have adopted the transformation of various data types into images as the feature extraction procedure, in cyber security and security incidents detection in particular, the researchers work mainly with numerical and nominal features extracted from the PCAP packets and rarely transform them into images.

This research investigates a direct transformation of the raw PCAP packets into images. This approach has demonstrated high performance in the malware analysis tasks where the researchers analyze, usually, raw binaries [8,9,48]. The most interesting capabilities of such transformation were demonstrated in [8,9]. The authors showed that the features extracted from the malware images allowed detection of the malicious code compiled with different compiler settings, and obfuscation techniques.

## 3. A Proposed Approach to Image-Based Feature Extraction

The key idea of a proposed approach is a transformation of raw (binary) network packets into grayscale images that serve as an input to a classification module. This idea relies on the hypothesis that the model trained on images is protocol independent and robust to different even unseen types of network attacks because it is not required to calculate protocol-specific features, and CNNs are able to reveal hidden non-linear relations between attributes. Thus, the proposed approach to network intrusion detection consists of the following steps:

1. An extraction of the raw (binary) packets from the PCAP files.
2. A transformation of each packet into a grayscale image.
3. An attack detection based on the analysis of the images.

The schema of the approach is given in Figure 1.

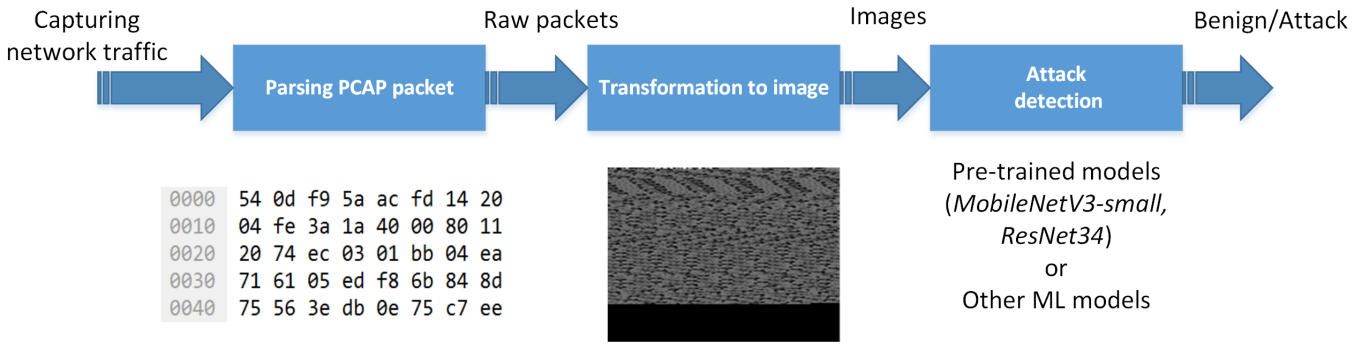


**Figure 1.** Generic schema of the intrusion detection using image-based features.

The grayscale images have only one channel that carries information about luminous intensity only, thus the pixels encode the amount of light only that varies from 0 to 255. In this case, the procedure of the network packet transformation is a quite straight-forward process. It does not require any extraction of such attributes as protocol type, service type, and others. Each network packet is treated as a binary sequence, which can be split into a sequence of bytes. Then, each byte is mapped into a grayscale level according to the following rule:

$$\begin{aligned} 0 \times 0 &\longrightarrow 0(\text{black color}) \\ &\cdots \\ 0 \times FF &\longrightarrow 255(\text{white color}) \end{aligned} \tag{1}$$

Figure 2 shows the process of byte conversion to a grayscale image.

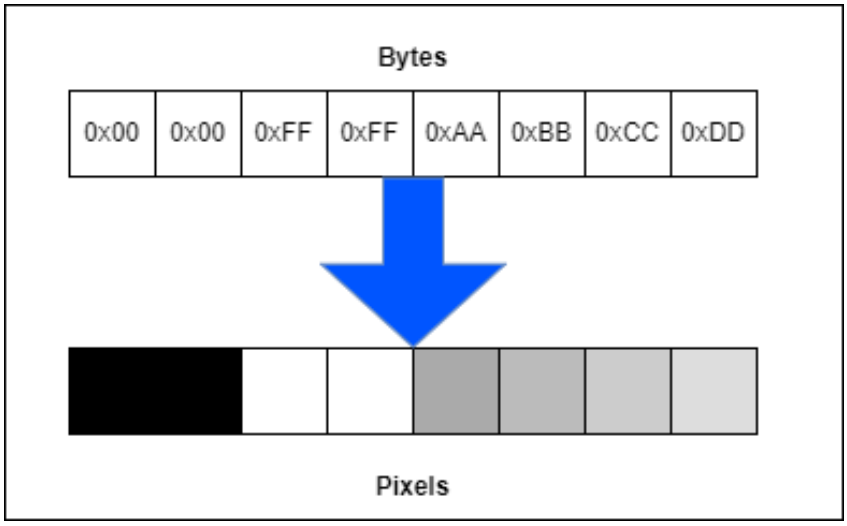


**Figure 2.** Process of a grayscale image generation from a sequence of bytes.

Each grayscale image can be considered as a two-dimensional matrix, and it is necessary to solve two tasks when constructing it:

1. To define a width and a height of the matrix, i.e., size of an image to be generated.
2. To select a way how pixels are filled during the image generation.

The task of the image size definition is not trivial as soon as the length of network packets varies significantly. In malware analysis, the most common approach to construct the image from the binary code is to set a constant image width and a variable height that depends on the length of the malware code. In major cases, the generated image requires implementation of a resizing operation to fit the input data format of an analysis model. For example, the format of the input data of the majority of the pre-trained deep neural networks is a square image (N $\times$ N). Thus, if a particular pre-trained neural network is used to detect malicious packets then the size of the image is defined by the input format of this neural network. The authors suggest using another option. It consists of an analysis of the packet length distribution in the test dataset, and a selection of the image size based on a calculated statistical measure such as a mean packet size, maximum or minimum packet size, median, etc. It is proposed to use Formula (2) to determine the image size $S_{image}$, where $P_{stat}$ stands for a statistical parameter characterizing packet lengths' distribution.

$$S_{image} = ceil(e^{ln(P_{stat})/2}) \quad (2)$$

In all cases except one, when the maximum packet length is selected to determine the image size, the packets are cropped when the image is generated. In order to preserve all information about the network packets, it is recommended to use the maximum possible size of the packet as a measure to determine the image size. If the packets have a smaller size, the extra bytes are filled up with $0 \times 00$.

There are several approaches to laying out the pixels within the image.

- *Linear layout*. The pixel matrix is filled up row by row. This is the most widely used approach.
- *Spiral layout*. The filling of the pixel matrix is started from the center and continues in a spiral form. This technique is suitable for square matrices.
- *Zig–zag layout*. This type of image construction is suggested in [49]. The pixels are filled up in the zigzag form. The underlying idea of such a layout relates to the fact that when the image size is greater than the length of the packet, the majority of pixels will be padded with zeros, however, the zig–zag filling of the pixel matrix with further discrete cosine transform can preserve and enforce existing patterns in low frequency part of the frequency domain.

Figure 3 shows pixel filling schemes for each approach and shows the corresponding examples of the generated images.

The attack detection could be performed by either pre-trained models such as VGG-16, or ResNet, or by a specially trained neural network. The authors studied both options, and the experiments were performed with two pre-trained models (ResNet34 [12] and MobileNetV3-small [50]) and one own CNN model.

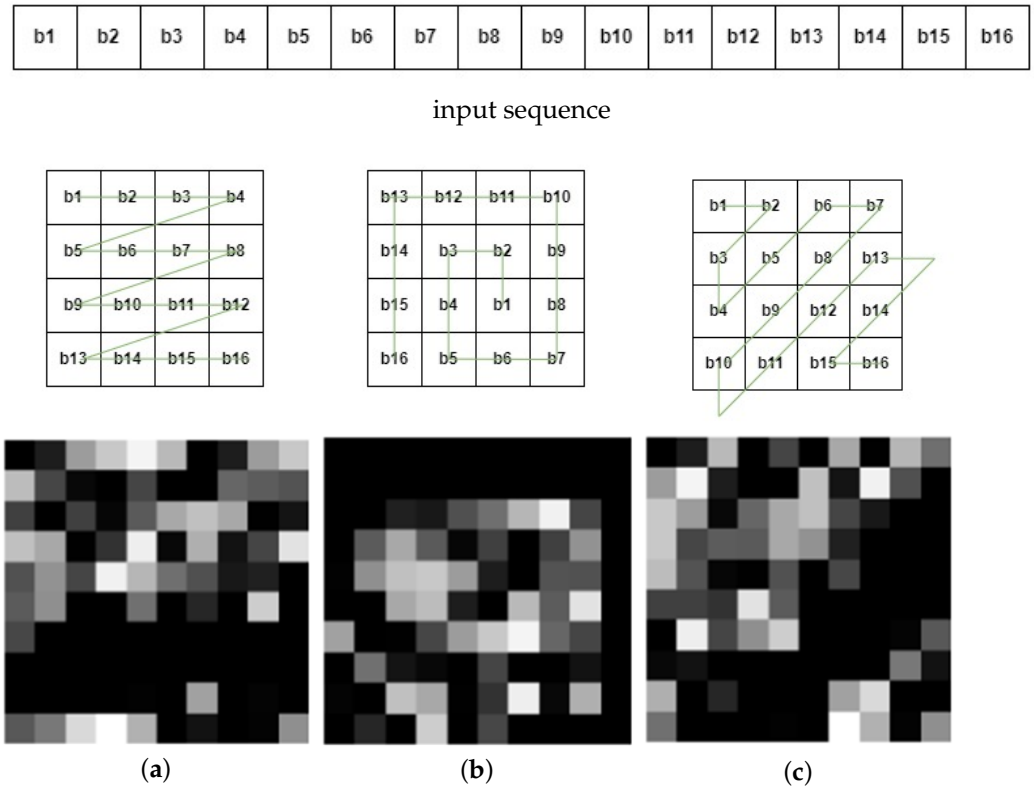


**Figure 3.** Different schemes for pixel layout: linear (**a**), spiral (**b**), and zig–zag (**c**).

## 4. Experiments and Discussion

To evaluate the approach, Secure Water Treatment (SWaT) dataset [51] was selected. This dataset is generated using a test bed that models a large water treatment facility with a six-staged technological process. The communication part of the SWAT test bed is represented by a layered communication network, programmable logical controllers, Supervisory Control and Data Acquisition (SCADA) server and a workstation, and a repository with historic data. The architecture of the test bed allows remote connection to the facility infrastructure by the operational personnel. The dataset has several versions that vary in types of collected data, attacks performed, and overall duration of the test bed functioning. To perform experiments, the SWaT.A6_Dec 2019 version of the dataset was selected. It consists of several PCAP files with network traffic and historical data from sensors in a .csv file. This dataset describes 3 h of normal functioning and 1 h during which 9 attacks of two different types were implemented. These attacks are targeted to extract historical data or disrupt the functioning of the sensors and actuators. Figure 4 shows the timeline of the attacks in the dataset with their type specification.

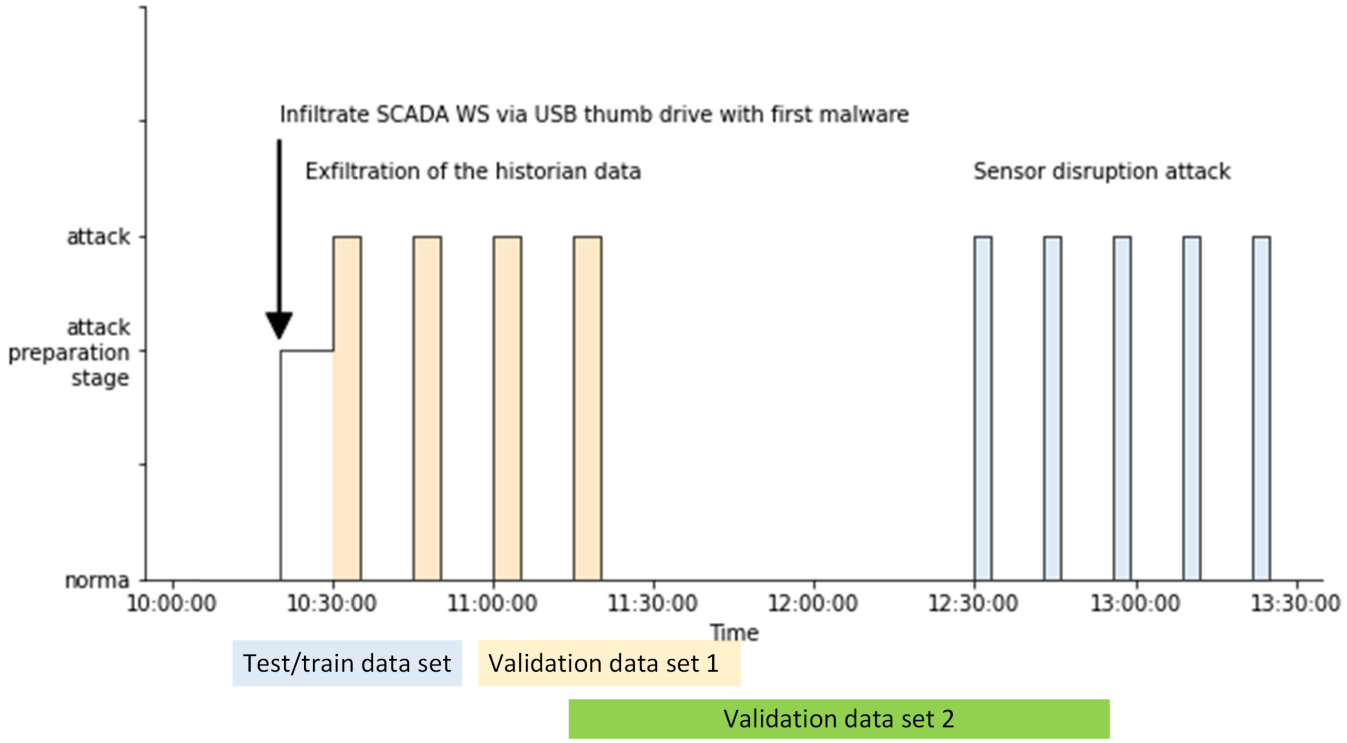


**Figure 4.** Timeline of the attacks in the SWaT dataset.

To evaluate the ability of the approach to detect attacks, the following experiment scenario was developed. The first stage included the analysis of the package length distribution and the choice of possible image size. In the second stage, the performance of the pre-trained models, as well as the impact of the different pixel layout schemes on their performance was evaluated. In the third stage, the experiments with the own CNN were performed. They included an analysis of the impact of image size, as well as the size of the training set on the model performance. The authors evaluated also its capability to detect unseen attacks. In order to implement this task, three different datasets were formed on the basis of the initial dataset. They are schematically shown in Figure 4: two datasets, namely (*test/train dataset*) and *validation dataset* 1 were formed from the first part of the SWAT dataset, and contained one attack type only, while the third dataset (*validation dataset* 2) was formed on the basis of the second part of the SWaT dataset and contained both types of attacks. The *test/train dataset* was used to train and assess the efficiency of all analysis models included in the experiments.

To assess the efficiency of the attack detection, the *accuracy*, *F1-measure*, *recall*, and *precision* metrics were used.

**Choice of image size.** The analysis of the network packet length distribution revealed that there is almost no difference in statistical characteristics for normal and abnormal packets—the maximum length of the malicious packets is slightly less than a normal one. Table 1 summarizes the obtained results, and Figure 5 shows the distribution of the packet lengths for normal and abnormal traffic. Using Formula (2), the maximum image size is determined as $138 \times 138$.

**Table 1.** Statistics on packet length distribution.

| Packet Type | Mode | Median | 80 Percentile | 99 Percentile | Min | Max |
|---|---|---|---|---|---|---|
| Normal | 64 | 90 | 128 | 633 | 60 | 19,034 |
| Attack | 64 | 86 | 128 | 633 | 60 | 14,888 |

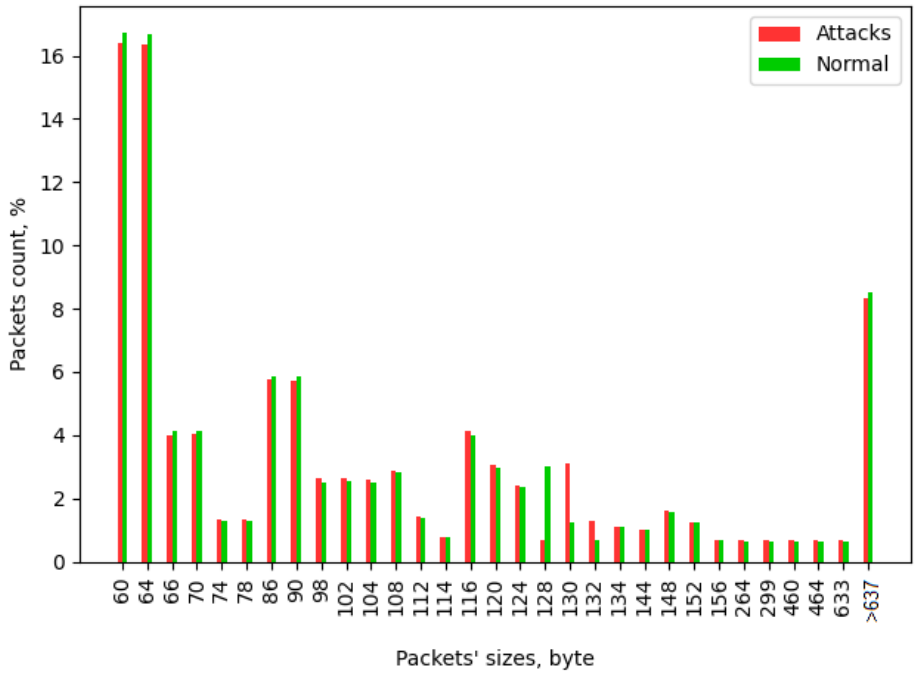


**Figure 5.** Distribution of the packet lengths for normal and abnormal traffic.

**Performance evaluation of the pre-trained deep neural networks.** In this scenario, a pre-trained model is used in the feature extraction task, and the weights of the final layer only are updated in order to make predictions. Two pre-trained neural networks, namely ResNet34 and MobileNetV3-small were selected. Both of these networks were trained on the ImageNet dataset, which contains images and is usually used in the object detection task. Their key characteristics such as the classification accuracy and the number of FLOPs are given in Table 2. It is obvious that the accuracy of the ResNet34 is higher than one of the MobileNetV3-small, however, the MobileNetV3-small is a lightweight neural network designed for mobile CPUs with low computational resources. It should be noted that the classification accuracy of these neural networks is higher than the ones that are used in [5,39].

**Table 2.** Characteristics of the selected pre-trained models.

| Pre-Trained Model | Input Image Size | Top-1 Accuracy | Top-5 Accuracy | FLOPs (Millions) | Num. of Trained Parameters |
|---|---|---|---|---|---|
| ResNet34 [12] | 224 × 224 | 73.31 | 91.42 | 21.8 | 1026 |
| MobileNetv3-small [50] | 224 × 224 | 67.67 | 87.40 | 2.5 | 2050 |

The experiments were performed with models from PyTorch model hub [52], and the following parameters were used in the experiments to train the model:

- Number of epochs: 7;
- Optimizer: SGD (Stochastic gradient descent);
- Loss function: CrossEntropyLoss;
- Learning rate: 0.001;
- Batch size: 16.

It should be noted that an attack with a duration of 15 min is described by 22 million network packets. To train the model, the dataset was reduced to 1 million packets, and balanced to keep the ratio of normal packets (70%) and attack packets (30%). The obtained results are given in Table 3. It is obvious that the performance of the classifiers is low and

comparable to the accuracy of a random classifier. The calculated metrics are slightly higher for the ResNet34 classifier with zig–zag pixel layout, but still not acceptable.

The authors assume that there are few reasons for such results. First of all, in the experiments, two pre-trained models were used in feature selection mode, i.e., all layers were "freezed", and the weights of the last layer only were updated during the training. The number of the training epochs was not large and, perhaps, not enough to train the model. Another possible reason is a significant difference between the train dataset and Image net dataset that was used to train MobileNetV3-small and ResNet3 models. These models are trained to detect real-world objects, while the images generated from network traffic are totally different, in major cases, they are mostly black images. Moreover, the size of the initially generated image is smaller than that is required for neural network input, and it is re-scaled using *nearest* interpolation operation [53]. Thus, the usage of the pre-trained models in feature extraction mode is not efficient, they require significant fine-tuning, i.e., updating all neural network weights. This process could be resource-exhaustive, especially for large models such as ResNet34. That is why authors believe that a possible way of transfer learning application could be as presented in [6]. The model is trained on some public labeled dataset with network attacks, and then fine-tuned on some private or other dataset. Additionally, in this case feature selection mode of the pre-trained models would be efficient as the subject domain of the datasets used to train and fine-tune the model is the same.

**Table 3.** Experimental results with pre-trained models and different pixel layout schemes.

| Pre-Trained Model | Pixel Layout Scheme | Accuracy | Precision | Recall | F1-Measure |
|---|---|---|---|---|---|
| ResNet34 [40] | linear | 0.56 | 0.51 | 0.51 | 0.49 |
| | spiral | 0.59 | 0.53 | 0.52 | 0.51 |
| | zig-zag | 0.64 | 0.55 | 0.57 | 0.57 |
| MobileNetv3-small [41] | linear | 0.54 | 0.49 | 0.47 | 0.48 |
| | spiral | 0.52 | 0.49 | 0.5 | 0.48 |
| | zig-zag | 0.56 | 0.5 | 0.52 | 0.44 |

**Experiments with CNN designed for the task.** To test the approach to feature extraction, the authors also developed their own CNN with the following architecture:

- 1 input layer;
- 3 convolutional layers;
- 3 max pooling layers;
- 1 flatten layer;
- 1 dense layer;
- 1 output layer.

This network was trained for 10 epochs and demonstrated significantly higher results. The experiments also showed that they do not depend on image size and the size of the training dataset. Table 4 gives brief results of the CNN training with different initial parameters, and Figure 6 provides detailed information on the training process for the case when the CNN was trained on the 10 millions of images. The best performance of the model is achieved on the 5–6 epoch training, then the precision and the recall metrics slightly deteriorate. The recall parameter is high, it equals 0.98, however, the precision is low—it is slightly higher than 0.7. The precision value is more important in the attack detection task, as it considers false positive cases, and the low value of precision indicates the high rate of false positive cases.

**Table 4.** Accuracy of the CNN trained with different initial parameters of the training dataset.

| Size of the Test Dataset | Image Size | Metric Used to Define Image Size | Accuracy |
|---|---|---|---|
| 1 million | 10 × 10 | Median (and mode) | 0.99 |
| | 26 × 26 | 99 Percentile | 0.99 |
| | 138 × 138 | Maximum size | 0.99 |
| 10 millions | 138 × 138 | Maximum size | 0.99 |

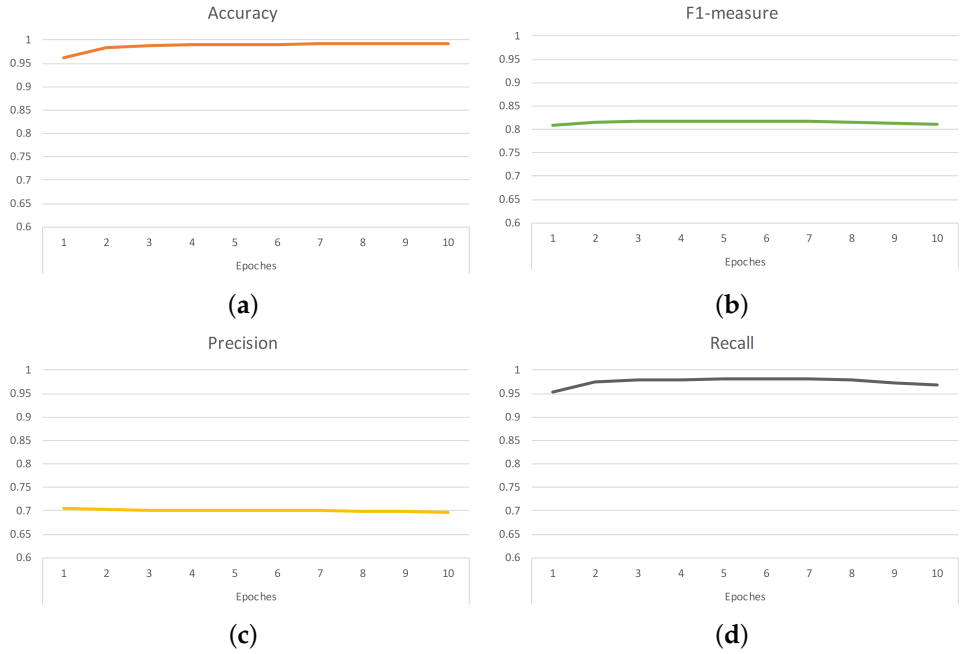


**Figure 6.** Parameters of the CNN training on the dataset consisting of 10 millions of images.

The last series of experiments was devoted to the evaluation of the CNN ability to detect novel and unseen attacks. In this experiment two datasets, *validation dataset 1* and *validation dataset 2*, were used. In the case of the first dataset the authors tried to evaluate the robustness of the model to detect similar attacks that could vary in such attributes as packet timestamps and checksums. The goal of the second dataset application is to assess the model robustness to novel and unseen attacks. It should be noted that the considered attacks are different in their origin, the target of the first attack is to retrieve data from the repository with archive data, while the target of the second attack is to disrupt the functioning of the sensors and actuators. The obtained results are shown in Table 5. Interestingly, they are almost similar for both validation datasets. The precision metric is slightly lower on the *test/train dataset*, but it is almost the same for both types of attacks, however, the recall metric that characterizes the true positive rate is high indicating that the classifier detected almost all samples with the attack. The latter looks very promising and stimulating to continue further research on the suggested approach to network traffic pre-processing. The main problem consists in decreasing the false positive rate while maintaining the high recall rate. The possible solution of this task is changing the parameters of the image construction procedure, for example, selecting the mode of network packet lengths as a base parameter to define image size. We also compared the performance of the CNN model to the performance of the Random Forest (RF) model that is often used in attack detection tasks as it usually demonstrates high performance and is characterized by low requirements to computational resources [54,55]. It was trained on *test/train dataset*, and then evaluated on the *validation dataset 2*. The performance of the RF on the dataset with a known attack was high, reaching 99% of the accuracy and F1-measure, however, when

applied to dataset with an unseen attack the accuracy decreased to 63%, thus indicating about low generality of the model.

Thus, a certain bottleneck of the suggested approach consists in selecting the image size that is used to generate an image. Though the performed experiments showed that it does not impact the accuracy much, it is required to have a priori information on statistical characteristics of package length distribution in order to define image size. The authors also consider that it is necessary to evaluate the impact of image size more thoroughly, for example, by performing experiments on different datasets that include different types of attacks, and by evaluating how the most important pixels (features) are located within the image. Secondly, though experiments showed that the pre-trained models in feature extraction mode do not demonstrate high efficiency, in a fine-tuning mode they could have high accuracy, and in this case the selection of the image size would depend also on the input format of the selected model.

**Table 5.** Performance of the CNN on different validation datasets.

| Validation Dataset | F1-Measure | Precision | Recall |
|---|---|---|---|
| Validation dataset 1 | 0.78 | 0.64 | 0.98 |
| Validation dataset 2 (with novel attack type) | 0.75 | 0.62 | 0.95 |

## 5. Conclusions

Approaches for feature extraction based on the transformation of the raw data into images have attracted recently a lot of research attention. This paper investigated the approach to attack detection that uses the transformation of network traffic into grayscale images.

The analysis of the related works showed that currently there are only a few research papers devoted to this problem, and they use a feature vector that is extracted from the PCAP files to generate an image. This paper suggests creating grayscale images directly from raw network packets that are extracted from PCAP files. The main motivation for such a solution is that the model trained on images could be robust to novel and unseen types of network attacks.

To evaluate the efficiency of the approach, a series of experiments were performed. They included performance assessment of the pre-trained models, such as MobileNetV3-small and ResNet34, training and testing our own CNN on different datasets with different types of attacks.

The SWaT dataset was used as a test dataset, which describes the functioning of a smart water treatment facility and contains such attacks as historical data extraction and sensor disruption.

The implemented experiments showed that the performance of the pre-trained models when they are used in feature extraction mode is low and comparable to the performance of the random classifier. The CNN which was created by authors demonstrated quite a high positive detection rate even when detecting unseen attacks. In future research, we plan to hyper-tune the CNN to enhance the results.

The identified drawbacks of the approach, including the high false positive rate of the classifier and the image size selection, defined the direction of future research work. They include evaluation of the pixel's importance in the context of their layout within the image in order to determine optimal image size. Another direction of the future research relates to the investigation of another technique for image construction, which assumes the generation of one image for a series of packets.

**Author Contributions:** Conceptualization, E.N.; methodology, E.N. and S.G.; validation, E.F. and S.G.; investigation, S.G., E.N. and E.F.; writing—original draft preparation, E.F. and S.G.; writing—review and editing, E.F. and E.N.; visualization, E.N.; project administration, E.N.; funding acquisition, E.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is being supported by the grant of RSF #22-21-00724 in SPC RAS.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** This publication is supported by the SWAT dataset, which is available at the location [51].

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CNN | Convolutional Neural Network |
| FLOPs | Floating Point Operations |

## References


1. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
2. Debnath, B.; O'Brient, M.; Kumar, S.; Behera, A. Attention-Driven Body Pose Encoding for Human Activity Recognition. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5897–5904. [CrossRef]
3. Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.; 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]
4. Chollet, F. A Transfer Learning with Deep Neural Network Approach for Network Intrusion Detection. *Int. J. Intell. Comput. Res.* **2021**, *12*, 1087–1095.
5. Noever, D.A.; Noever, S.E.M. Image Classifiers for Network Intrusions. *arXiv* **2021**, arXiv:2103.07765.
6. Wu, P.; Guo, H.; Buckland, R. A Transfer Learning Approach for Network Intrusion Detection. In Proceedings of the 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA), Suzhou, China, 15–18 March 2019; pp. 281–285. [CrossRef]
7. Branitskiy, A.; Kotenko, I. Analysis and Classification of Methods for Network Attack Detection. *SPIIRAS Proc.* **2016**, *2*, 207. [CrossRef]
8. Alrabaee, S.; Karbab, E.B.; Wang, L.; Debbabi, M. BinEye: Towards Efficient Binary Authorship Characterization Using Deep Learning. In *Proceedings of the Computer Security—ESORICS 2019*; Sako, K., Schneider, S., Ryan, P.Y.A., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 47–67.
9. Kaur, R.; Ning, Y.; Gonzalez, H.; Stakhanova, N. Unmasking Android obfuscation tools using spatial analysis. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland, 28–30 August 2018; pp. 1–10.
10. Rong, C.; Gou, G.; Cui, M.; Xiong, G.; Li, Z.; Guo, L. TransNet: Unseen Malware Variants Detection Using Deep Transfer Learning. In *Proceedings of the Security and Privacy in Communication Networks*; Park, N., Sun, K., Foresti, S., Butler, K., Saxena, N., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 84–101.
11. Wang, F.; Chai, G.; Li, Q.; Wang, C. An Efficient Deep Unsupervised Domain Adaptation for Unknown Malware Detection. *Symmetry* **2022**, *14*, 296. [CrossRef]
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
13. Golubev, S.; Novikova, E. Image-based Intrusion Detection in Network Traffic. In *Proceedings of the Intelligent Distributed Computing XV*; Braubach, L., Jander, K., Bădic, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2022.
14. Zhu, W. On the model-checking-based IDS. *arXiv* **2018**, arXiv:1806.09337.
15. Kruegel, C.; Toth, T. Using Decision Trees to Improve Signature-Based Intrusion Detection. In *Proceedings of the Recent Advances in Intrusion Detection*; Vigna, G., Kruegel, C., Jonsson, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 173–191.
16. Chen, W.H.; Hsu, S.H.; Shen, H.P. Application of SVM and ANN for intrusion detection. *Comput. Oper. Res.* **2005**, *32*, 2617–2634. [CrossRef]
17. Heckerman, D. A Tutorial on Learning with Bayesian Networks. In *Innovations in Bayesian Networks: Theory and Applications*; Holmes, D.E., Jain, L.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 33–82. [CrossRef]
18. Barbará, D.; Wu, N.; Jajodia, S., Detecting Novel Network Intrusions Using Bayes Estimators. In Proceedings of the 2001 SIAM International Conference on Data Mining (SDM), Chicago, IL, USA, 5–7 April 2001; pp. 1–17.
19. Mukkamala, S.; Sung, A.H.; Abraham, A.; Ramos, V. Intrusion Detection Systems Using Adaptive Regression Spines. In *Proceedings of the Enterprise Information Systems VI*; Seruca, I., Cordeiro, J., Hammoudi, S., Filipe, J., Eds.; Springer: Dordrecht, The Netherlands, 2006; pp. 211–218.
20. Ranjan, R.; Sahoo, G. A New Clustering Approach for Anomaly Intrusion Detection. *Int. J. Data Min. Knowl. Manag. Process.* **2014**; *4*, 29–38. [CrossRef]
21. Wang, Y. A multinomial logistic regression modeling approach for anomaly intrusion detection. *Comput. Secur.* **2005**, *24*, 662–674. [CrossRef]

22. Sheth, H.; Shah, B.; Yagnik, S.B. A Survey on RBF Neural Network for Intrusion Detection System. *Int. J. Eng. Res. Appl.* **2014**, *4*, 17–22.
23. Sammany, M.; Sharawi, M.; El-beltagy, M.; Saroit, I. Artificial Neural Networks Architecture For Intrusion Detection Systems and Classification of Attacks. In Proceedings of the 5th International Conference INFO2007, Cairo University, Giza, Egypt, 24–26 March 2007.
24. Lu, W.; Traore, I. Detecting New Forms of Network Intrusion Using Genetic Programming. In Proceedings of the Congress on Evolutionary Computation, Canberra, Australia, 8–12 December 2003; IEEE-Press: Piscataway, NJ, USA, 2004; Volume 20, pp. 2165–2172. [CrossRef]
25. Mahendiran, A.; Appusamy, R. A Survey on Intrusion Detection System Using Fuzzy Logic. *Int. J. Control Theory Appl.* **2016**, *9*, 7517–7522.
26. Powers, S.T.; He, J. A hybrid artificial immune system and Self Organising Map for network intrusion detection. *Inf. Sci.* **2008**, *178*, 3024–3042. [CrossRef]
27. Barford, P.; Kline, J.; Plonka, D.; Ron, A. A signal analysis of network traffic anomalies. In Proceedings of the IMW'02, Marseille, France, 6–8 November 2002.
28. Denning, D. An Intrusion-Detection Model. *IEEE Trans. Softw. Eng.* **1987**, *SE-13*, 222–232. [CrossRef]
29. Gu, Y.; Mccallum, A.; Towsley, D. *Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation*; USENIX Association: Berkeley, CA, USA, 2005; pp. 345–350. [CrossRef]
30. Dymora, P.; Mazurek, M. Network Anomaly Detection Based on the Statistical Self-similarity Factor. *Lect. Notes Electr. Eng.* **2015**, *324*, 271–287. [CrossRef]
31. Lee, K.; Kim, J.; Kwon, K.H.; Han, Y.; Kim, S. DDoS attack detection method using cluster analysis. *Expert Syst. Appl.* **2008**, *34*, 1659–1665. [CrossRef]
32. Bazgir, O.; Zhang, R.; Dhruba, S.R.; Rahman, R.; Ghosh, S.; Pal, R. Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks. *Nat. Commun.* **2020**, *11*, 4391. [CrossRef]
33. Su, R.; Liu, X.; Wei, L.; Zou, Q. Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response. *Methods* **2019**, *166*, 91–102. [CrossRef] [PubMed]
34. Lim, J.; Ryu, S.; Park, K.; Choe, Y.J.; Ham, J.; Kim, W.Y. Predicting drug-target interaction using a novel graph neural network with 3D structure-embedded graph representation. *J. Chem. Inf. Model.* **2019**, *59*, 3981–3988. [CrossRef]
35. NCI60 Drug Response Data Set. Available online: https://dtp.cancer.gov/databases_tools/bulk_data.htm (accessed on 1 November 2022).
36. Drug Sensitivity in Cancer (GDSC) Data Set. Available online: https://www.cancerrxgene.org/downloads/bulk_download (accessed on 1 November 2022)
37. Zhu, Y.; Brettin, T.; Xia, F.; Partin, A.; Shukla, M.; Yoo, H.; Evrard, Y.; Doroshow, J.; Stevens, R. Converting tabular data into images for deep learning with convolutional neural networks. *Sci. Rep.* **2021**, *11*, 11325. [CrossRef] [PubMed]
38. Cancer Therapeutics Response Portal v2 (CTRP). Available online: https://portals.broadinstitute.org/ctrp.v2.1/ (accessed on 1 November 2022)
39. Masum, M.; Shahriar, H. TL-NID: Deep Neural Network with Transfer Learning for Network Intrusion Detection. In Proceedings of the 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 8–10 December 2020; pp. 1–7. [CrossRef]
40. Wang, W.; Wang, Z.; Zhou, Z.; Deng, H.; Zhao, W.; Wang, C.; Guo, Y. Anomaly detection of industrial control systems based on transfer learning. *Tsinghua Sci. Technol.* **2021**, *26*, 821–832. [CrossRef]
41. Zhao, J.; Shetty, S.; Pan, J.W.; Kamhoua, C.; Kwiat, K. Transfer learning for detecting unknown network attacks. *Int. J. Comput. Vision* **2019**, *2019*, 1. [CrossRef]
42. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
43. NSL-KDD Data Set. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 1 November 2022)
44. Manjula, P.; Sankaralingam, B.P. An effective network intrusion detection and classification system for securing WSN using VGG-19 and hybrid deep neural network techniques. *J. Intell. Fuzzy Syst.* **2022**, *43*, 6419–6432. [CrossRef]
45. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
46. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
47. Park, S.; Kim, M.; Lee, S. Anomaly Detection for HTTP Using Convolutional Autoencoders. *IEEE Access* **2018**, *6*, 70884–70901. [CrossRef]
48. Nataraj, L.; Karthikeyan, S.; Jacob, G.; Manjunath, B.S. Malware Images: Visualization and Automatic Classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security, Pittsburgh, PA, USA, 20 July 2011; Association for Computing Machinery: New York, NY, USA, 2011. [CrossRef]
49. Zhang, X.; Chen, J.; Zhou, Y.; Han, L.; Lin, J. A Multiple-Layer Representation Learning Model for Network-Based Attack Detection. *IEEE Access* **2019**, *7*, 91992–92008. [CrossRef]

50. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
51. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In Proceedings of the Critical Information Infrastructures Security, Lucca, Italy, 8–13 October 2017; Havarneanu, G., Setola, R., Nassopoulos, H., Wolthusen, S., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 88–99.
52. PyTorch Model Hub. Available online: https://pytorch.org/vision/stable/models.html (accessed on 1 November 2022)
53. Suresh, C.; Singh, S.; Saini, R.; Saini, A.K. A Comparative Analysis of Image Scaling Algorithms. *Int. J. Image Graph. Signal Process.* **2013**, *5*, 55–62. [CrossRef]
54. Chen, L.; Zhang, Y.; Zhao, Q.; Geng, G.; Yan, Z. Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark. *Procedia Comput. Sci.* **2018**, *134*, 310–315. [CrossRef]
55. Resende, P.A.A.; Drummond, A.C. A Survey of Random Forest Based Methods for Intrusion Detection Systems. *ACM Comput. Surv.* **2018**, *51*, 1–38. [CrossRef]