electronics

MDPI

Article

Anomaly Detection Model of Network Dataflow Based on an Improved Grey Wolf Algorithm and CNN

Liting Wang, Qinghua Chen * and Chao Song

Naval Aviation University, Yantai 264001, China; litingwang_start@163.com (L.W.); schhh_1983@126.com (C.S.)
* Correspondence: 201813678@sdtbu.edu.cn; Tel.: +86-13963879380

Abstract: With the popularization of the network and the expansion of its application scope, the problem of abnormal network traffic caused by network attacks, malicious software, traffic peaks, or network device failures is becoming increasingly prominent. This problem not only leads to a decline in network performance and service quality but also may pose a serious threat to network security. This paper proposes a hybrid data processing model based on deep learning for network anomaly detection to improve anomaly detection performance. First, the Grey Wolf optimization algorithm is improved to select high-quality data features, which are then converted to RGB images and input into an anomaly detection model. An anomaly detection model of network dataflow based on a convolutional neural network is designed to recognize network anomalies, including DoS (Denial of Service), R2L (Remote to Local), U2R (User to Root), and Probe (Probing). To verify the effectiveness of the improved Grey Wolf algorithm and the anomaly detection model, we conducted experiments on the KDD99 and UNSW-NB15 datasets. The proposed method achieves an average detection rate of 0.986, which is much higher than all the counterparts. Experimental results show that the accuracy and the detection rates of our method were improved, while the false alarm rate has been reduced, proving the effectiveness of our approach in network anomaly classification tasks.

Keywords: anomaly detection; feature selection; Grey Wolf algorithm; convolutional neural network

check for updates

Citation: Wang, L.; Chen, Q.; Song, C. Anomaly Detection Model of Network Dataflow Based on an Improved Grey Wolf Algorithm and CNN. *Electronics* **2023**, *12*, 3787. https://doi.org/10.3390/electronics12183787

Academic Editors: Wenguan Wang, Tianfei Zhou and Xiankai Lu

Received: 31 July 2023
Revised: 3 September 2023
Accepted: 5 September 2023
Published: 7 September 2023

1. Introduction

Network abnormal traffic refers to data streams in a communication network that deviate from normal traffic behavior patterns. It is usually caused by reasons such as network attacks, malicious software, traffic surges, or network equipment failures. Network abnormal traffic not only leads to decreased network performance and service quality but also poses a serious threat to network security. For example, network attackers may utilize abnormal traffic for denial of service attacks, DDoS attacks, intrusion attempts, or data breaches. Additionally, abnormal network traffic can result in network interruptions, data loss, system crashes, and significant damage to critical infrastructure. The detection and analysis of abnormal traffic to identify anomalous activities has become a hot topic in current network security research. This research has widespread applications in various fields, such as smart logistics and intelligent warehouse management. It is also important for the autonomous vehicle's data transmission as they are an important step for valid data preprocessing [1–3]. Research on autonomous driving security has attracted attention recently [4–6]. For example, Xia et al. [4] proposed a data acquisition and analytics platform for automated driving systems (ADS) to realize connected automated vehicle (CAV) cooperative perception.

In the field of network security research, because deep learning algorithms can learn the deep features of large-scale network data, they have been widely used in network space security. Kim et al. [7] proposed a denial of service (DoS) attack network intrusion detection system using a convolutional neural network (CNN). They developed a deep learning (DL) model specifically designed for DoS attacks. The experiments were conducted on the KDD dataset. Leveraging an ample number of samples from the KDD dataset, the model was trained using a convolutional neural network to recognize various types of attacks. To effectively extract and learn spatio-temporal features, Kanna et al. [8] proposed a high-precision IDS model using a unified model of optimized CNN and hierarchical multi-scale to effectively extract and learn spatio-temporal features. Thakkar et al. [9] propose a novel feature selection technique focused on enhancing the performance of DNN-based IDS, which selects features by incorporating statistical importance using standard deviation and mean and median differences. Shahin et al. [10] proposed a deep hybrid learning model to improve network intrusion detection systems and successfully applied it in the Industrial Internet of Things (IoT).

Anomaly detection in high-dimensional data is a key research problem with serious implications for real-world problems [11,12]. Li et al. [13] proposed an intrusion detection method based on deep learning, which transforms network traffic data into images that can be processed by CNNs. Subsequently, deep learning models are utilized for anomaly detection. Wang et al. [14] introduced a method for classifying malicious software traffic using convolutional neural networks and representation learning. Experimental results demonstrate that this method exhibits high performance and effectiveness in the classification of malicious software traffic. To enhance the capabilities of anomaly detection models, Garg et al. [15] proposed a hybrid data processing model for network anomaly detection using Gray Wolf Optimization (GWO) and Convolutional Neural Networks (CNN). This method improves the exploration, exploitation, initial population generation capabilities, and dropout function. The effectiveness of the proposed model is validated on benchmarks (DARPA'98 and KDD'99) and synthetic datasets, outperforming other state-of-the-art models (for network anomaly detection). In order to enhance the reliability of SDN, Garg et al. [16] proposed a hybrid deep learning-based anomaly detection scheme for suspicious flow detection in the context of social multimedia. Muneer et al. [17] proposed a hybrid model based on a five-layer deep autoencoder neural network (DANN) to reduce the difference between input and output. During the experiments, the class imbalance and poor performance of the dataset were investigated and resolved to improve the results.

The existing methods [13–17] for real-time detection of network anomalies using large-scale network log data suffer from low efficiency, computational complexity, and high false positive rates. By leveraging evolutionary computation methods to improve feature selection, an optimal trade-off between the two objectives can be achieved, resulting in reduced error rates and minimized feature sets. The anomaly detection method based on feature selection holds significant significance in enhancing anomaly detection efficiency and reducing false positive rates. Commonly used evolutionary computation methods include genetic algorithms, particle swarm optimization algorithms, ant colony optimization algorithms, and artificial immune system algorithms. Mirjalili et al. [18] proposed the Grey Wolf Optimizer (GWO), which mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves, namely alpha, beta, delta, and omega, are utilized to simulate the leadership hierarchy. Additionally, three main steps are performed: searching for prey, encircling prey, and attacking prey. The effectiveness of this algorithm has been validated through comparative studies with Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Differential Evolution (DE), Evolutionary Programming (EP), and Evolution Strategies (ES). The results demonstrate that the algorithm is well-suited for challenging problems with unknown search spaces. Mirjalili et al. also provided a reviewer of its recent variants and applications in the document [19]. Meidani et al. [20] proposed an improved version of the Grey Wolf Optimizer, which adapted the tuning of the parameters based on the fitness of the candidate solutions during the optimization. It can automatically converge to a sufficiently good optimum in the shortest time. Wang et al. [21] proposed an improved Grey Wolf Optimizer (IGWO) with evolutionary and elimination mechanisms to achieve a proper trade-off between exploration and exploitation, further enhancing the convergence speed and optimization accuracy of GWO. Experimental results demonstrate that the IGWO algorithm exhibits faster convergence speed and higher optimization accuracy.

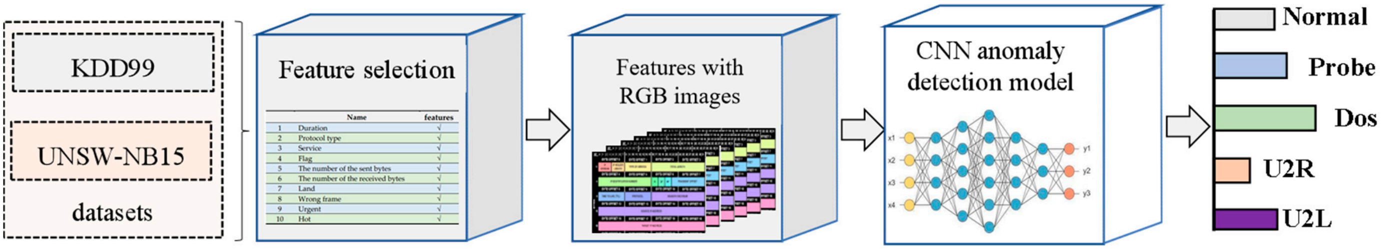
Currently, research on network anomaly traffic detection has achieved significant advancements. Methods and algorithms based on statistical analysis, machine learning, data mining, and deep learning techniques have been widely applied in network anomaly traffic detection [22,23]. However, research on network anomaly traffic detection faces a series of challenges. First, network anomaly traffic exhibits diversity and complexity, potentially having different features and behavioral patterns, thus necessitating the design of flexible and robust detection methods. Second, network data typically possess high dimensionality, high velocity, and large-scale characteristics, requiring efficient algorithms and technologies to achieve real-time anomaly traffic detection. Additionally, network anomaly traffic detection must consider the trade-off between false positive and false negative rates on normal traffic to ensure accuracy and reliability.

Due to the heterogeneous and diverse nature of cloud environments, existing techniques may not be applicable to handle the challenges induced by the existence of virtualized environments and the underlying security risks. Thus, it is meaningful to introduce hybrid data processing to involve both historical and real-time data streams. This paper aims to investigate methods and technologies for network anomaly traffic detection and propose an efficient, accurate, and reliable model for network anomaly detection. Such a model will be of great significance in the research of real-time video security analysis technology [24,25], intelligent warehouse management, autonomous driving security, and other fields.

The remainder of this paper is organized as follows. Section 2 introduces the proposed method. In Section 3, experimental results are presented. Section 4 discusses our results, and Section 5 makes some concluding remarks.

2. Materials and Methods

The anomaly detection model of network dataflow based on an improved Grey Wolf Algorithm and CNN is shown in Figure 1.



	Name	features
1	Duration	√
2	Protocol type	√
3	Service	√
4	Flag	√
5	The number of the sent bytes	√
6	The number of the received bytes	√
7	Land	√
8	Wrong frame	√
9	Urgent	√
10	Hot	√

Figure 1. Anomaly detection model of network dataflow based on an improved Grey Wolf algorithm and CNN.

2.1. Feature Selection Based on the Gray Wolf Optimization Algorithm

The Grey Wolf Optimization (GWO) algorithm draws inspiration from the social behaviors exhibited by a pack of grey wolves. This simulation mode effectively explores the search space, enhancing the likelihood of discovering the global optimal solution. GWO, which does not necessitate the gradient information of the objective function, is particularly suited for non-smooth and non-convex problems, rendering it well-performing for intricate scenarios. The algorithm maintains individual diversity during the search process, diminishing the susceptibility entrapment in local optima. Relative to other evolutionary algorithms, GWO boasts simpler parameter configurations. It typically achieves commendable solutions within a relatively limited number of iterations, thereby accelerating convergence rates when seeking global optima.

2.1.1. Principles of Grey Wolf Optimization Algorithm

In the Grey Wolf Algorithm, a hierarchical structure is formed among grey wolves, including leaders and followers. Leaders typically have better fitness and advantageous positions, while followers improve their fitness by learning from and imitating the leaders. This hierarchical structure and collaborative behavior can be applied to solving optimization problems. The Grey Wolf algorithm exhibits excellent global search and local optimization capabilities, making it suitable for various optimization problems, including function optimization, machine learning, and image processing. It is simple to implement and does not rely on gradient information. In the context of network anomaly classification, the grey wolf optimization algorithm can be applied to feature selection, model parameter optimization, and other aspects to enhance the performance and effectiveness of the classifier. The basic idea of GWO is shown in Figure 2.

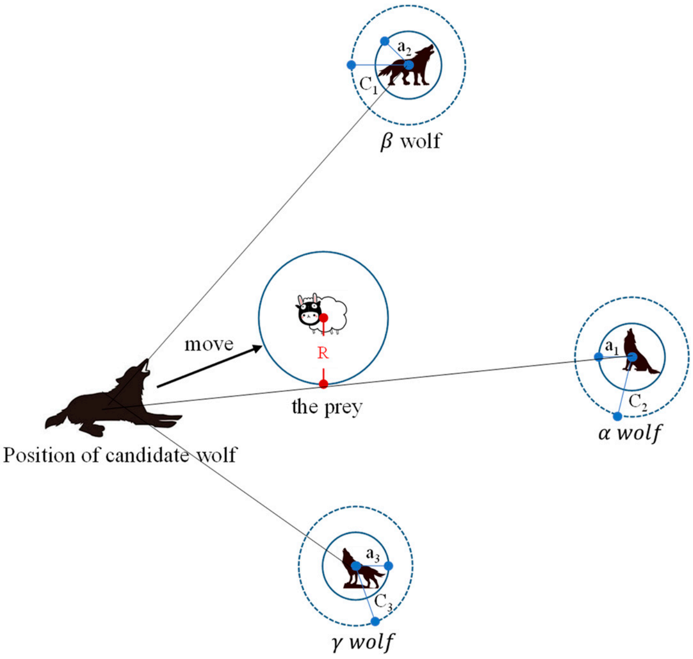


Figure 2. Principle of Grey Wolf Algorithm.

The search process of the Grey Wolf Algorithm is as follows:

(1) Initialize the population: In the initial population, randomly select a group of grey wolves as the population. Each grey wolf represents a potential solution.

(2) Evaluate fitness: Calculate the fitness value of each grey wolf based on a specific fitness function for the problem. This value is used to assess the quality of the solutions.

(3) Determine the leader: Select the grey wolf with the best fitness value as the leader. The position of the leader represents the current optimal solution.

(4) Update grey wolf positions: Update the position of each grey wolf based on their distances and fitness values. The position update is influenced by the leader, as the better solutions guide the search direction of other solutions.

(5) Handle boundaries: When updating positions, ensure that the grey wolves' positions do not exceed the defined boundaries of the problem.

(6) Iterative search: Repeat steps three to five until reaching the predetermined number of iterations or meeting the stopping criteria.

(7) Output the result: After completing the iterative search, output the found optimal solution as the solution to the optimization problem.

2.1.2. Basic Grey Wolf Optimization Algorithm

(1) A random initial population is generated by creating a certain number of wolves. Each wolf is represented by a matrix of size *i* rows and *n* columns, where *i* is the number of wolves and *n* is the number of features. Each column of the matrix is randomly assigned a value of 0 or 1, where 1 represents the selection of that feature, and 0 represents the non-selection of that feature. The fitness of each wolf is calculated using a fitness function. The wolves are then sorted based on their fitness, with lower fitness indicating closer proximity to the prey. The top three wolves are designated as α, β, and δ wolves, while the remaining wolves belong to the omega category. The fitness is calculated according to the following definition:

$$\mathrm{Fit}_1 = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{Fit}_2 = \gamma \times \frac{\mathrm{F}'}{\mathrm{F}} + (1-\gamma) \times \frac{\varepsilon_{\mathrm{c}}^{\mathrm{F}'}}{\varepsilon_{\mathrm{c}}^{\mathrm{F}}} \tag{2}$$

where FP, FN, TP, and TN represent false positives, false negatives, true positives, and true negatives, respectively. γ is a random number between [0, 1]. F' represents the number of selected features, and F represents the total number of features. $\varepsilon_{\mathrm{c}}^{\mathrm{F}'}$ the error rate with the selected features, and $\varepsilon_{\mathrm{c}}^{\mathrm{F}}$ is the error rate with all features. Equation (1) aims to minimize the error rate, while Equation (2) ensures the minimum number of features.

(2) Initialize the system vector using the following formula:

$$\mathrm{A} = 2\mathrm{a} \times \mathrm{r}_1 - \mathrm{a} \tag{3}$$

$$\mathrm{C} = 2 \times \mathrm{r}_2 \tag{4}$$

where a is the convergence factor, which linearly decreases from 2 to 0 with the number of iterations. r_1 and r_2 are two random vectors between 0 and 1.

(3) The grey wolves begin their hunting by calculating the distances between each omega wolf and the α, β, and δ wolves. Based on the positions of the α, β, and δ wolves, each ε wolf updates its own position to encircle the prey. Each population's position update represents an iteration. After each position update, the fitness of each wolf is recalculated, and new α, β, and δ wolves are selected for the next iteration. This process is repeated iteratively. The distance formula and position update formula are as follows:

$$\mathrm{D}_\alpha = |\mathrm{C}_1 \times \mathrm{X}_\alpha - \mathrm{X}| \tag{5}$$

$$\mathrm{D}_\beta = |\mathrm{C}_2 \times \mathrm{X}_\beta - \mathrm{X}| \tag{6}$$

$$\mathrm{D}_\delta = |\mathrm{C}_3 \times \mathrm{X}_\delta - \mathrm{X}| \tag{7}$$

$$\mathrm{X}_1 = \mathrm{X}_\alpha - \mathrm{A}_1 \times (\mathrm{D}_\alpha) \tag{8}$$

$$\mathrm{X}_2 = \mathrm{X}_\beta - \mathrm{A}_2 \times (\mathrm{D}_\beta) \tag{9}$$

$$\mathrm{X}_3 = \mathrm{X}_\delta - \mathrm{A}_3 \times (\mathrm{D}_\delta) \tag{10}$$

$$\mathrm{X}_{\mathrm{t}+1} = \frac{\mathrm{x}_1 + \mathrm{x}_2 + \mathrm{x}_3}{3} \tag{11}$$

where D_α, D_β, and D_δ represent the distances between the α wolf, β wolf, δ wolf, and the current ω wolf, respectively. Here, α, β, and *δ* refer to different wolf populations. X_α, X_β, and X_δ indicate the current positions of the α wolf, β wolf, and *δ* wolf, respectively, while X represents the current position of the ω wolf. $\mathrm{X}_{\mathrm{t}+1}$ denotes the updated position after the update.

2.1.3. The Improved Grey Wolf Optimization Algorithm

The traditional Grey Wolf Algorithm utilizes vectors A and C for exploration, which may result in a rapid convergence of the pseudo-Pareto front during the iteration process. The pseudo-Pareto front refers to an approximate Pareto front obtained through an optimization algorithm in multi-objective optimization but is not the actual Pareto front.

In multi-objective optimization, the Pareto front is a set of non-dominated solutions representing the solutions that cannot be further improved in one objective without sacrificing the others. Each solution in the Pareto front is the best trade-off solution, and no solution can outperform others on all objectives. However, optimization algorithms may find a pseudo-Pareto front, which includes some non-dominated solutions but is not the true Pareto front. The Pseudo-Pareto front may contain redundant or local optimal solutions, or it may lack certain true Pareto optimal solutions.

The conventional optimization process may get stuck in local optima, wasting considerable time. Therefore, a mutation function is introduced. Through mutation operations, individuals can randomly jump around the current solution, hoping to escape local optima and search for the global optimum. To address this issue, a mutation method is employed to enhance the exploration capability of the Grey Wolf Algorithm. The function P_m is introduced, and the formula is as follows:

$$P_m = 0.5e^{-10*t/T} + 0.01 \tag{12}$$

where t represents the current iteration number, and T denotes the total number of iterations, making it an exponentially decreasing function. At this point, we compare P_m with a random number between 0 and 1; if P_m is greater than the random number, a mutation operation is performed. The range of mutation is recorded with N, and the mutation formula is as follows:

$$N = \max\left\{1, \quad \left\lceil D - \left(\frac{t}{T}\right)^{\gamma} \times P_m \right\rceil \right\} \tag{13}$$

where D represents the total number of features, and γ is a random number between 0 and 1. The result obtained is the number of features to be subjected to mutation. The wolf to be mutated is initialized, and its first feature to the Nth feature is reassigned to achieve mutation.

2.2. Anomaly Detection Model of Network Dataflow Based on VGG16

Representing network flow data as RGB images and leveraging the image classification capabilities of convolutional neural networks, we perform network anomaly classification. This paper builds a classification model based on VGG16 [26], as illustrated in Figure 3.

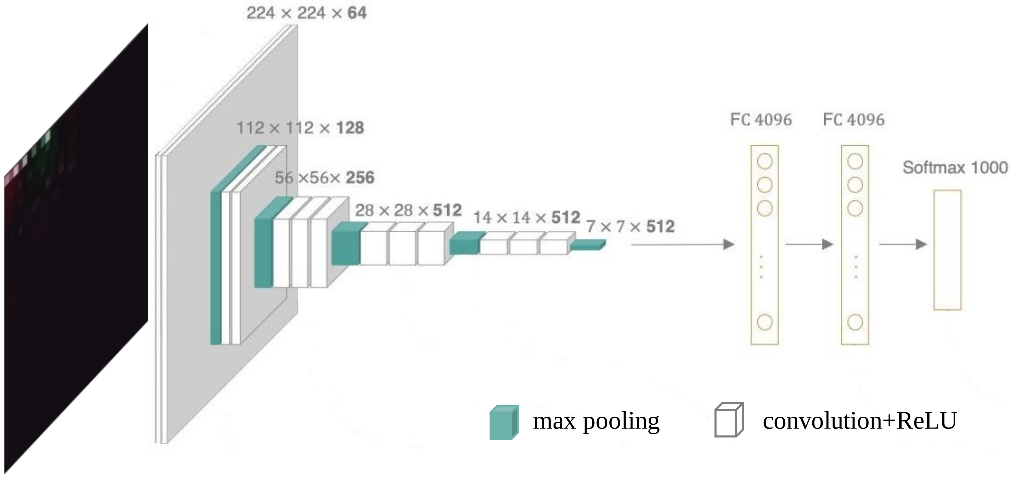


Figure 3. The architecture of the network anomaly detection model based on VGG16.

(1) Input layer: Receives the pixel values of the input image.
(2) Convolutional layers: The VGG16 model consists of 13 convolutional layers, where each convolutional layer utilizes a 3 × 3-sized convolutional kernel and employs

the ReLU activation function for non-linear transformation. The purpose of these convolutional layers is to extract features from the input image.

(3) Pooling layers: After each convolutional layer, the VGG16 model uses 2 × 2 max-pooling layers to perform downsampling operations, reducing the spatial dimensions of the feature maps while retaining the most prominent features.

(4) Fully connected layers: The VGG16 model contains three fully connected layers, each comprising 4096 neurons. The role of these fully connected layers is to convert the feature maps into specific class probabilities.

(5) Softmax layer: Following the last fully connected layer is the Softmax layer, which is used to map the output of the network to a probability distribution over classes.

(6) Output layer: The output layer provides the final classification result.

3. Results

3.1. Datasets

3.1.1. KDD99

KDD99 [27] (Knowledge Discovery in Databases 1999) is a commonly used network intrusion detection dataset, which was provided in the 1999 International Data Mining and Knowledge Discovery Competition (KDD Cup). This dataset is widely used to evaluate and study the performance and algorithms of network intrusion detection systems. The KDD99 data set is a network traffic data set based on the TCP/IP protocol, which contains a large number of network connection records, including both normal network connections and various types of network attacks. These attack types include DoS (denial of service), R2L (remote to local), U2R (user to root), Probing (detection), etc.

The KDD99 dataset encompasses approximately 494,020 network connection samples and comprises 41 distinct features. These features encompass various attributes associated with network connections, such as source IP address, destination IP address, source port, destination port, connection duration, packet count, and others. These features are utilized to establish models and algorithms aimed at discerning whether network connections are benign or indicative of malicious activity. The original data of the KDD99 dataset is stored in a textual format, where each line represents network traffic data with values separated by commas and features separated by periods. Therefore, the KDD99 dataset can be viewed as a matrix with a dimension of 494,020 rows and 42 columns. The number 494,020 represents the total of 494,020 network data entries, with the first 41 columns being the network data features and the forty-second column representing the type label. The text-formatted data has been converted into an RGB image format using the Pandas library and PIL library in Python, as illustrated in Figure 4. To handle the class imbalance in the dataset during training, we employed data augmentation methods [28] to randomly crop and flip the training data, generating more training samples to increase data diversity. To speak concretely, given the image I(x, y), the random crop operation can be implemented by sub-image cropping. The sub-image is included in the original image, and its center position is randomly selected. Using this method, our random crop augmentation crops the original image into a given size sub-image randomly. Then, the cropped images are scaled up to the original image size by an upsampling technique.

Given the image, the horizontal or vertical flip augmentation flips the input image along its vertical or horizontal axis randomly with a given probability. The obtained augmented image I(x', y') is formulated as follows (left: horizontal, right: vertical):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{14}$$

Related sentences have been added in the updated version.

When employing the KDD99 dataset, customary preprocessing steps are often undertaken, including data sampling, feature selection, feature value normalization, and class balance adjustment. The dataset has been divided into a training set and a test set in a 7:3 ratio, where 70% of the data serves as the training set for model training. The remaining 30% is allocated as the test set for evaluation.

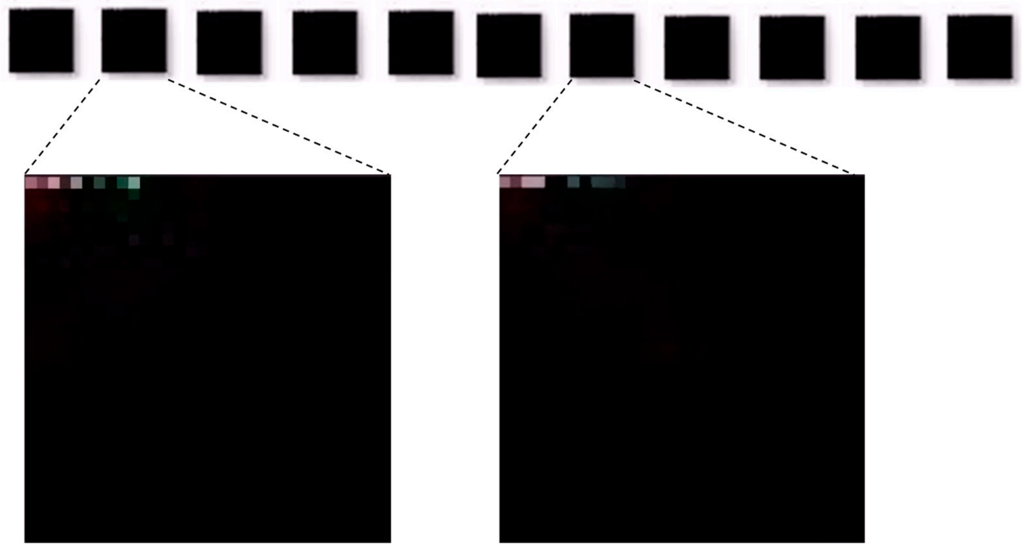


Figure 4. RGB images of network data.

3.1.2. UNSW-NB15

UNSW-NB15 dataset is an amalgamation of real traffic and generated attack data from the University of New South Wales (UNSW) [29]. It is a dataset extracted from 100 GB of normal and modern attack traffic by researchers at the Australian Centre for Cyber Security (ACCS) using the IXIA tool. It contains nine different attacks, including DoS, worms, Backdoors, and Fuzzers. The dataset contains raw network packets. The number of records in the training set is 175,341 records, and the testing set is 82,332 records from the different types, attack and normal.

3.2. Experiment Settings

The experiment was conducted using PyTorch 1.11.0, Python 3.8 (Ubuntu 20.04), and Cuda 11.3. The GPU selected for the experiment was RTX 4090 (24 GB), and the CPU chosen was 15 vCPU Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz. The system had 80GB of memory (RAM). We clarify that the hyperparameters of the VGG16 convolution network are finetuned by using the Adam Optimization Algorithm with a steady learning rate of 0.03. The batch size is set to 16.

3.3. Evaluation Metrics and Methods

Cross-entropy loss [30] was used as train loss. To assess the model performance, train loss, accuracy, detection rate (DR), false positive rate (FPR), precision, and F-score were used, which were defined as follows:

$$\text{train loss} = \frac{\text{running loss}}{\text{train steps}} \tag{15}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{16}$$

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

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

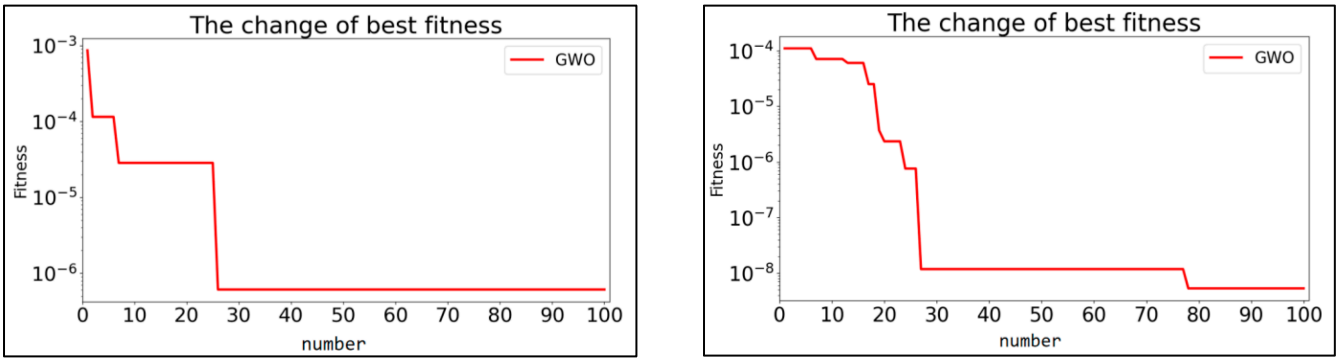
$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{F} - \text{score} = 2 \times \frac{\text{Precision} \times \text{DR}}{\text{Precision} + \text{DR}} \tag{20}$$

where running loss refers to the accumulated training loss during the training process, train steps represent the number of training steps performed during the training phase, and TP, TN, FP, and FN are True Positive, True Negative, False Positive, and False Negative, respectively. TP refers to the situation where normal network traffic is predicted by the model to be normal. FP refers to situations where other abnormal categories are predicted as normal. TN refers to the situation where normal categories are misclassified into other abnormal categories, while FN refers to the situations where abnormal categories are predicted as abnormal categories. Notably, False Positive Rate and False Negative Rate are important criteria that show how many standard data are predicted as anomalous and how many anomalies are, on average, missed by the detector, respectively.

3.4. Experimental Results of Optimizing the Grey Wolf Algorithm

The results of feature selection using the traditional Grey Wolf Algorithm and the improved Grey Wolf Algorithm are shown in Figure 5.



(**a**) The fitness of the traditional Grey Wolf Algorithm (**b**) The fitness of the improved Grey Wolf Algorithm

Figure 5. The fitness of the traditional Grey Wolf Algorithm and the improved Grey Wolf Algorithm. The horizontal axis represents the number of iterations, while the vertical axis represents the fitness. Lower fitness values indicate better optimization performance.

The fitness of the traditional Grey Wolf Algorithm reached the level of 10^{-6} after around 26 iterations, whereas the improved algorithm achieved this level around the 19th iteration. As a result, the improved algorithm required fewer iterations, accelerating the feature selection process of the Grey Wolf Algorithm. The fitness of the improved algorithm after 25 iterations reached the level of 1.0^{-8}, In comparison the fitness of the traditional algorithm after 100 iterations remained at the level of 1.0^{-6}, which indicates an enhanced exploration space in the improved Grey Wolf Algorithm, allowing it to escape local optima and search for global optima.

The selected features are shown in Table 1. The selected features are reduced from 39 to 36 when using the improved Grey Wolf Algorithm.

Table 1. The selected features. √ denotes that the corresponding feature is selected in the method, while - denotes that the corresponding feature is not selected in the method.

	Name	**Description**	**The Selected Features**	
			Traditional Algorithm	**Improved Algorithm**
1	Duration	Connection duration	√	√
2	Protocol type	Protocol types for links: TCP, UDP, ICMP	√	√
3	Service	Network service types: HTTP, FTP, SMTP, etc.	√	√
4	Flag	Status flags for connections	√	√
5	The number of the sent bytes	Number of bytes sent by a link	√	-
6	The number of received bytes	The number of bytes accepted by a link	√	-
7	Land	Is the source IP, port consistent with the target IP, and port?	√	√

Table 1. *Cont.*

	Name	Description	The Selected Features	
			Traditional Algorithm	**Improved Algorithm**
8	Wrong frame	Number of segments with invalid checksums in the connection	√	√
9	Urgent	Number of emergency segments in the connection	√	√
10	Hot	Number of hot metrics related to the current connection	√	√
11	The number of the wrong land	Incorrect login count in a link	√	√
12	Logged_in	Successfully logged in	√	√
13	Num_compromised	The total number of errors not found in a link	√	√
14	Root_shell	The root is getting the shell	√	-
15	Su_attempted	Whether to try to authenticate as Superuser	√	√
16	Num_root	Number of users with root privileges in a link	√	√
17	Num_file_creation	Number of files created in a link	√	√
18	Num_shells	Number of normal user logins	√	√
19	Num_access_file	Number of operation control files in a file	√	√
20	Num_outbound_cmds	Number of outbound commands in FTP sessions	-	√
21	Is_hot_login	Is the user accessing as root or administrator?	√	√
22	Is_guest_login	Is it a guest login?	√	√
23	count	Number of links to the same destination IP	√	√
24	Srv_count	Number of links to the same destination port	√	√
25	Serror_rate	The ratio of incorrect links	√	√
26	Srv_serror_rate	The rate of incorrect links related to the current service	√	√
27	Rerror_rate	The rate of rejecting connections	√	√
28	Srv_error_rate	The rate of rejected links related to the current service	√	√
29	Sane_srv_rate	The ratio of links that are the same as the current service	√	√
30	Diff_srv_rate	The ratio of links different from the current service	√	√
31	Srv_diff_host_rate	The ratio of links from different hosts that are the same as the current service	√	√
32	Dst_host_count	The same number of links as the target host	√	√
33	Dst_host_srv_count	Number of connections to the same port	√	-
34	Dst_host_same_srv_rate	The ratio of links to the same service as the target host	√	√
35	Dst_host_diff_srv_rate	The ratio of links to different services from the target host	√	-
36	Dst_host_same_src_port_rate	Link ratio with the same source port as the target host	√	√
37	Dst_host_srv_diff_host_rate	The ratio of links from different hosts with the same service as the target host	-	√
38	Dst_host_serror_rate	The rate of incorrect links related to the target host	√	√
39	Dst_host_srv_serror_rate	The rate of incorrect links related to the target host service	√	√
40	Dst_host_rerror_rate	The rate of rejected links related to the target host	√	√
41	Dst_host_srv_error_rate	The rate of rejected links related to the service of the target host	√	√

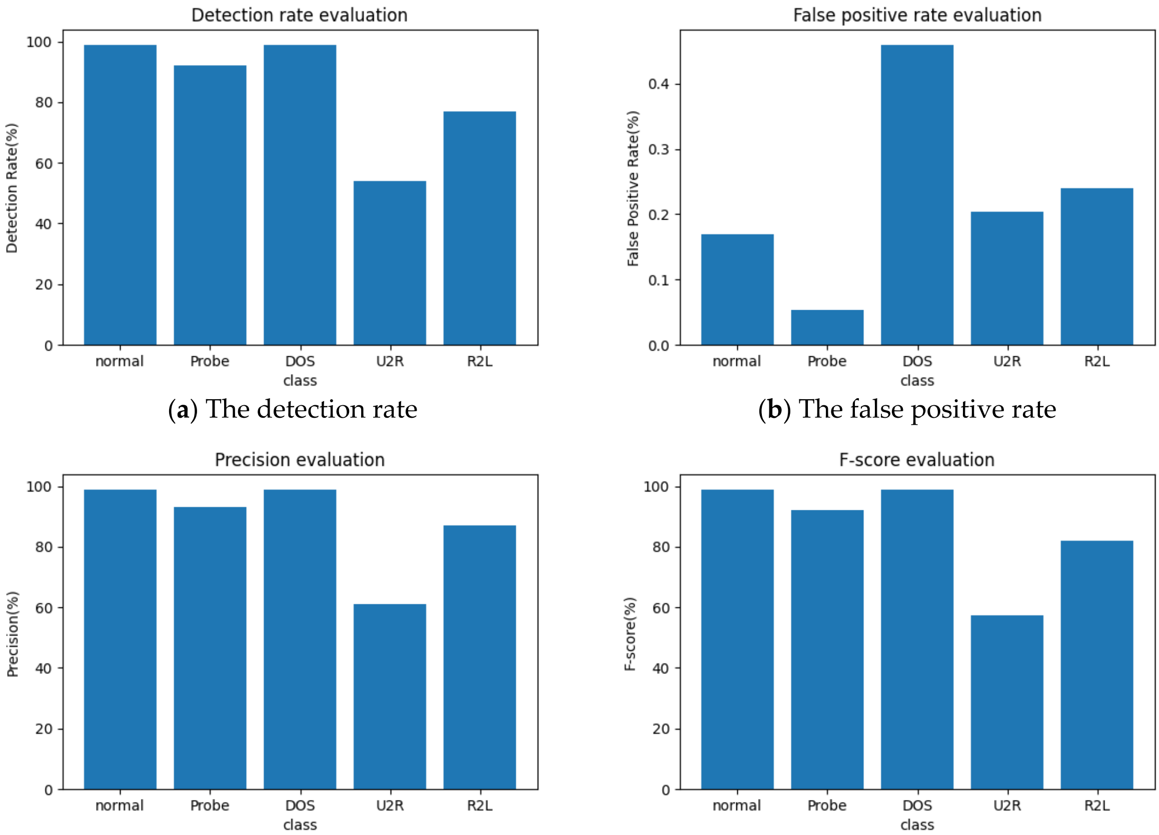
√ denotes that the corresponding feature is selected in the method, while - denotes that the corresponding feature is not selected in the method.

3.5. Experimental Results of Anomaly Detection on the KDD99 Dataset

Figure 6 shows the performance of anomaly detection of our method.

The average detection rate is about 0.986, as seen in Figure 6a. The detection rate represents the probability that a certain type of data can be detected, which can well reflect the sensitivity of the model to that type of data. From Figure 6, it can be seen that the model has a high detection rate for data of normal type, Probe type, and DOS type, all of which are above 90%. The detection rate for R2L data is also close to 80%. Due to the small number of original data samples, all performances of U2R are relatively low. The false positive rate represents the probability of detecting negative samples as positive samples, which reflects the probability of errors in the detection method. Figure 6b shows that the model has a low false positive rate for normal, Probe, U2R, and R2L types while a relatively high false positive rate for DOS types, all of which are below 0.5%. Figure 6c shows that the model's accuracy for normal, Probe, DOS, and R2L types exceeds 90%, while U2R is relatively low at about 60%. The F-score represents the fusion of detection rate and accuracy. As shown in Figure 6d, the model has an overall higher F-score for normal, Probe, DOS, and R2L types, while U2R is relatively lower.

Comparison results with other methods on the KDD99 dataset are shown in Table 2.



(**a**) The detection rate

(**b**) The false positive rate

(**c**) Precision

(**d**) F-score

Figure 6. The performance of anomaly detection.

Table 2. Comparison results with other methods on the KDD99 dataset.

Methods	Detection Rate	False Positive Rate	Precision	F-Score
Sharma et al. [31]	93.41	0.275	99.05	93
Pandeeshwari et al. [32]	98	3.05	-	83.20
Guo et al. [33]	91.86	0.78	93.29	-
Proposed Model	98.6	0.278	94.86	92.24

We can see from Table 2 that the proposed method achieves an average detection rate of 0.986, which is higher than all the counterparts. The false positive rate, precision, and F-score also achieved good results.

3.6. Experimental Results of Anomaly Detection on the UNSW-NB15 Dataset

We also conducted experiments on UNSW-NB15 and reported the performance of our proposed method with other similar previous works, as shown in Tables 3 and 4.

Table 3. Results about classifiers.

Attack Class	Accuracy	Precision	Recall	F-Score
Exploits	0.921	0.763	0.596	0.669
Generic	0.990	0.990	0.962	0.981
Reconnaissance	0.969	0.812	0.665	0.733
Analysis	0.893	0.899	0.951	0.925
Shellcode	0.794	0.817	0.987	0.890
DoS	0.847	0.799	0.943	0.867
Worms	0.851	0.782	0.891	0.843
Fuzzers	0.789	0.761	0.994	0.862
Backdoors	0.723	0.672	0.855	0.751

Table 4. Comparison results with other methods on the UNSW-NB15 dataset.

Methods	Accuracy (%)
Kasongo and Sun [34]	77.51
Roy and Singh [35]	84.1
Kasongo and Sun [36]	77.16
Eunice et al. [37]	82.1
Moustafa and Slay [38]	81.34
Proposed method	84.5

The confusion matrix is shown in Figure 7. We can see that our method can handle the different classes of anomaly detection in the UNSW-NB15 dataset, especially for Fuzzers and Recon cases, and our solution can address the data imbalance effectively.

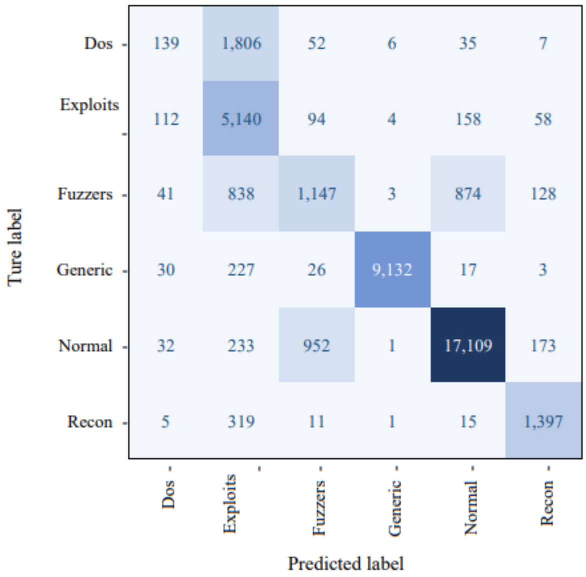


Figure 7. The confusion matrix.

4. Discussion

This paper first optimizes the GWO algorithm for extracting network traffic features and achieves network anomaly recognition by visualizing features and establishing a classification model. This method is different from the existing text feature-based recognition methods. The experiments were conducted on the KDD99 dataset, and the experimental results showed that the model achieved high scores in terms of detection rate, accuracy, and F-score for normal, Probe, and Dose. Table 2 shows that the proposed method achieves an average detection rate of 0.986, which is higher than all the counterparts. The false positive rate, precision, and F-score also achieved good results.

Due to the limited samples of U2R and R2L in the dataset, the detection rate, accuracy, and F-score of U2R and R2L are relatively low. To this end, this paper focuses on the problem of category imbalance in the KDD99 dataset. It adopts data augmentation methods to randomly crop and flip the training data, generating more training samples to increase data diversity. In addition, Dropout is used to randomly inactivate a portion of the data to prevent overfitting, thereby breaking the dependency relationship between the data and reducing the situation of collaborative adaptation. In this paper, dropout is set to 0.5, randomly discarding half of the output. More experiments will be conducted in future work to demonstrate the robustness of our method.

More experiments on the UNSW-NB15 dataset also demonstrate the effectiveness of the proposed method. Comparing results with other methods, our method achieves an accuracy of 84.5%, which is higher than the results of other methods.

5. Conclusions

This paper proposes an improved Grey Wolf Optimization algorithm for the feature selection of data sets. The data are then represented by images, and the Convolutional Neural Network algorithm is used to train the classification model. Experiments are conducted on the KDD99 and UNSW-NB15 datasets to evaluate the effectiveness of the model by recording indicators such as accuracy, detection rate, and false alarm rate. The experimental results show that this method performs well in network anomaly classification tasks. The improved Gray Wolf Optimization algorithm is used for feature selection, which makes the selected features have high accuracy and reduces the number of features, thus improving the accuracy and efficiency of the model. Our approach can improve the performance by enhancing anomaly detection efficiency and improving the detection rate.

In future research, we plan to use multiple optimization algorithms to further improve the performance of feature selection and introduce more complex deep learning models, such as recurrent neural networks, attention mechanisms, etc., to further improve the performance and generalization ability of the model. In addition, it is possible to consider using larger real network datasets to comprehensively evaluate the robustness and scalability of the model.

Author Contributions: Conceptualization, L.W.; methodology, L.W. and Q.C.; software, L.W.; validation, Q.C.; formal analysis, L.W. and Q.C.; investigation, Q.C. and C.S.; resources, Q.C. and C.S.; data curation, L.W.; writing-original draft preparation, L.W. and Q.C.; writing-review and editing, L.W.; visualization, L.W.; supervision, L.W. and C.S.; L.W. and Q.C. designed the study; L.W. analyzed and interpreted the data; L.W. conducted the experiments; L.W. and Q.C. provided the technical and material support. All authors contributed to the writing of the manuscript and final approval. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China under grant no. 61971432.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors thank the anonymous reviewers for their valuable comments and suggestions.

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

References

1. Xia, X.; Bhatt, N.P.; Khajepour, A.; Hashemi, E. Integrated Inertial-LiDAR-Based Map Matching Localization for Varying Environments. *IEEE Trans. Intell. Veh.* **2023**, 1–12. [CrossRef]
2. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [CrossRef]
3. Meng, Z.; Xia, X.; Xu, R.; Liu, W.; Ma, J. HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR. *IEEE Trans. Intell. Veh.* **2023**, 1–13. [CrossRef]
4. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An automated driving systems data acquisition and analytics platform. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104120. [CrossRef]
5. Gao, C.; Wang, G.; Shi, W.; Wang, Z.; Chen, Y. Autonomous Driving Security: State of the Art and Challenges. *IEEE Internet Things J.* **2021**, *9*, 7572–7595. [CrossRef]
6. Bogdoll, D.; Nitsche, M.; Zöllner, J.M. Anomaly detection in autonomous driving: A survey. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–22 June 2022; pp. 4488–4499.
7. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. CNN-based network intrusion detection against denial-of-service attacks. *Electronics* **2020**, *9*, 916. [CrossRef]
8. Kanna, P.R.; Santhi, P. Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features. *Knowl.-Based Syst.* **2021**, *226*, 107132. [CrossRef]
9. Thakkar, A.; Lohiya, R. Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System. *Inf. Fusion* **2023**, *90*, 353–363. [CrossRef]

10. Shahin, M.; Chen, F.F.; Hosseinzadeh, A.; Bouzary, H.; Rashidifar, R. A deep hybrid learning model for detection of cyber attacks in industrial IoT devices. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1973–1983. [CrossRef]
11. Tian, Z.; Luo, C.; Qiu, J.; Du, X.; Guizani, M. A distributed deep learning system for web attack detection on edge devices. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1963–1971. [CrossRef]
12. Yu, X.; Yang, X.; Tan, Q.; Shan, C.; Lv, Z. An edge computing based anomaly detection method in IoT industrial sustainability. *Appl. Soft Comput.* **2022**, *128*, 109486. [CrossRef]
13. Li, Z.; Qin, Z.; Huang, K.; Yang, X.; Ye, S. Intrusion detection using convolutional neural networks for representation learning. In *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, 14–18 November 2017, Proceedings, Part V*; Springer International Publishing: Cham, Switzerland, 2017; pp. 858–866.
14. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), IEEE, Da Nang, Vietnam, 11–13 January 2017; pp. 712–717.
15. Garg, S.; Kaur, K.; Kumar, N.; Kaddoum, G.; Zomaya, A.Y.; Ranjan, R. A hybrid deep learning-based model for anomaly detection in cloud data center networks. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 924–935. [CrossRef]
16. Garg, S.; Kaur, K.; Kumar, N.; Rodrigues, J.J.P.C. Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective. *IEEE Trans. Multimed.* **2019**, *21*, 566–578. [CrossRef]
17. Muneer, A.; Taib, S.M.; Fati, S.M.; Balogun, A.O.; Aziz, I.A. A Hybrid Deep Learning-Based Unsupervised Anomaly Detection in High Dimensional Data. *Comput. Mater. Contin.* **2022**, *70*. [CrossRef]
18. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
19. Faris, H.; Aljarah, I.; Al-Betar, M.A.; Mirjalili, S. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **2017**, *30*, 413–435. [CrossRef]
20. Meidani, K.; Hemmasian, A.; Mirjalili, S.; Barati Farimani, A. Adaptive grey wolf optimizer. *Neural Comput. Appl.* **2022**, *34*, 7711–7731. [CrossRef]
21. Wang, J.S.; Li, S.X. An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Sci. Rep.* **2019**, *9*, 7181. [CrossRef] [PubMed]
22. Yidan, L.; Yanli, C.; Runze, C.; Lan, Y.; Fangming, R. An Encryption Traffic Classification Method Based on ResNeXt. In Proceedings of the 2021 IEEE 15th International Conference on Anti-counterfeiting, Security, and Identification (ASID), IEEE, Xiamen, China, 29–31 October 2021; pp. 47–52.
23. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
24. Qin, Z.; Lu, X.; Nie, X.; Liu, D.; Yin, Y.; Wang, W. Coarse-to-Fine Video Instance Segmentation with Factorized Conditional Appearance Flows. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1192–1208. [CrossRef]
25. Lu, X.; Wang, W.; Shen, J.; Crandall, D.; Luo, J. Zero-Shot Video Object Segmentation with Co-Attention Siamese Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2228–2242. [CrossRef]
26. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
27. Chaudhuri, S.; Madigan, D.; Fayyad, U. KDD-99: The fifth ACM SIGKDD international conference on knowledge discovery and data mining. *ACM SIGKDD Explor. Newsl.* **2000**, *1*, 49–51. [CrossRef]
28. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
29. Khamis, R.A.; Matrawy, A. Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs. In Proceedings of the IEEE ISNCC 2020: 2020 IEEE International Symposium on Networks, Computers and Communications, Montreal, QC, Canada, 20–22 October 2020; pp. 1–6.
30. Gneiting, T.; E Raftery, A. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [CrossRef]
31. Sharma, N.; Mukherjee, S. A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS. *Procedia Technol.* **2012**, *6*, 913–921. [CrossRef]
32. Pandeeswari, N.; Kumar, G. Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN. *Mob. Netw. Appl.* **2016**, *21*, 494–505. [CrossRef]
33. Guo, C.; Ping, Y.; Liu, N.; Luo, S.-S. A two-level hybrid approach for intrusion detection. *Neurocomputing* **2016**, *214*, 391–400. [CrossRef]
34. Kasongo, S.M.; Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *J. Big Data* **2020**, *7*, 1–20. [CrossRef]
35. Roy, A.; Singh, K.J. Multi-classification of UNSW-NB15 Dataset for Network Anomaly Detection System. In *Proceedings of International Conference on Communication and Computational Technologies, Algorithms for Intelligent Systems*; Purohit, S., Singh Jat, D., Poonia, R., Kumar, S., Hiranwal, S., Eds.; Springer: Singapore, 2020; pp. 429–451. [CrossRef]
36. Kasongo, S.M.; Sun, Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput. Secur.* **2020**, *92*, 101752. [CrossRef]

37. Eunice, A.D.; Gao, Q.; Zhu, M.-Y.; Chen, Z.; Na, L. Network anomaly detection technology based on deep learning. In Proceedings of the 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), IEEE, Greenville, SC, USA, 12–14 November 2021; pp. 6–9.
38. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. A Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.