technologies

MDPI

Article

Network Intrusion Detection Integrating Feature Dimensionality Reduction and Transfer Learning

Hui Wang [1], Wei Jiang [1], Junjie Yang [2,*], Zitao Xu [1] and Boxin Zhi [1]

[1] College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai 201306, China; dxxy5045@mail.shiep.edu.cn (H.W.); shiepjw@shiep.edu.cn (W.J.); m18744806822_2@163.com (Z.X.); lxqioun@gmail.com (B.Z.)

[2] College of Electronics and Information Engineering, Shanghai Dianji University, Shanghai 201306, China

* Correspondence: yangjj@sdju.edu.cn; Tel.: +86-133-7189-6843

Abstract

In the Internet era, network malicious intrusion behaviors occur frequently and network intrusion detection is increasingly in demand. Addressing the challenges of high-dimensional data, nonlinearity and noisy network traffic data in network intrusion detection, a net-work intrusion detection model is proposed in this paper. Firstly, a hybrid multi-model feature selection and kernel-based dimensionality reduction algorithm is proposed to map high-dimensional features to low-dimensional space to achieve feature dimensionality reduction and enhance nonlinear differentiability. Then the semantic feature mapping is introduced to convert the low-dimensional features into color images which represent distinct data characteristic. For classifying these images, an integrated convolutional neural network is constructed. Moreover, sub-model fine-tuning is performed through transfer learning and weights are assigned to improve the performance of multi-classification detection. Experiments on the UNSW-NB15 and CICIDS 2017 datasets show that the proposed model achieves accuracies of 99.99% and 99.96%. The F1-scores of 99.98% and 99.91% are achieved respectively.

Keywords: intrusion detection; feature dimensionality reduction optimization; traffic visualization; transfer learning; integrated neural networks

Academic Editor: Arash Habibi Lashkari

Received: 4 August 2025
Revised: 2 September 2025
Accepted: 5 September 2025
Published: 10 September 2025

Citation: Wang, H.; Jiang, W.; Yang, J.; Xu, Z.; Zhi, B. Network Intrusion Detection Integrating Feature Dimensionality Reduction and Transfer Learning. *Technologies* **2025**, *13*, 409. https://doi.org/10.3390/technologies13090409

1. Introduction

In recent years, with the rapid development of the Internet and information technology, network security problems have become increasingly serious [1]. Network intrusion, defined as unauthorized access to a computer system or network, may lead to data leakage, system paralysis, or even serious economic losses. Intrusion detection techniques [2] are widely used to monitor and identify abnormal behaviors in real time to enhance network security. Traditional intrusion detection methods are mainly based on rules [3] and statistical analysis, such as expert-defined feature rules, signature detection [4], and anomaly detection. Traditional intrusion detection methods are ineffective because rule-based methods are difficult to update in a timely manner, and statistical-based methods suffer from high false alarm rates and low generalization capabilities.

With the development of deep learning, intrusion detection methods based on deep learning have received increasing attention [5]. In 2015, Wang et al. pioneered the reconstruction of raw network packets from byte streams into grayscale images [6], a groundbreaking work that revealed the potential of convolutional neural networks (CNNs) in

handling protocol field spatial correlation. Hindy et al. proposed a dual temporal–spatial convolutional architecture [7], which effectively solves the heterogeneous characterization problem of discrete log data and continuous flow data by transforming the temporal features recorded by NetFlow into a two-dimensional spectrogram. Mazid et al. developed a 3D-CNN framework by fusing three-channel inputs—data packet length, time interval, and protocol type [8]—which achieved a high recall rate for detecting abnormal behavior in IoT devices, verifying the unique advantage of CNNs in high-dimensional spatiotemporal feature extraction.

Currently, existing binary classification detection models lack fine-grained identification of specific attack types. Developing multi-classification schemes is crucial for formulating precise, attack-specific defense strategies, and effective data preprocessing plays a key role in enhancing model generalization and accuracy. Furthermore, existing models struggle with large-scale, heterogeneous traffic data. Specifically, reliance on a single CNN model often fails to meet detection requirements, and insufficient feature generalization impedes the balance between accuracy and robustness across diverse intrusion types.

To address these limitations, this paper designs an interpretable, high-precision network intrusion detection (HMK-ST) model integrating feature dimensionality reduction, semantic mapping, and transfer learning. To address the high dimensionality, nonlinearity, and low signal-to-noise ratio of network data, this paper proposes a hybrid multi-model feature selection and kernel dimensionality reduction algorithm, which maps the high-dimensional features to a low-dimensional space to achieve feature dimensionality reduction and enhance nonlinear separability. A network traffic visualization algorithm is proposed based on semantic feature mapping that encodes structured data into color semantic images. Furthermore, an ensemble convolutional neural network model is constructed to perform multiclass detection on these generated images, enabling highly accurate network intrusion detection.

In this paper, we validate the performance of the HMK-ST model on the UNSW-NB15 and CICIDS 2017 datasets. Performance analysis shows that on the UNSW-NB15 dataset, the HMK-ST model achieved an accuracy of 99.99% and an F1 score of 99.98%. On the CICIDS 2017 dataset, it achieved an accuracy of 99.96% and an F1 score of 99.91%.

The rest of this document is organized as follows: Section 2 examines the current literature for network intrusion detection methods and introduces the general framework of the model in this paper. Section 3 describes the web intrusion detection (HMK-ST) model with feature dimensionality reduction, semantic mapping, and transfer learning. Section 4 validates the performance of the HMK-ST model proposed in this paper using a dataset. Section 5 summarizes the experimental results of the model in this paper and suggests future research directions for the model.

2. Related Work

Due to the increasing complexity of network environments and the diversification of attack methods, building an efficient, accurate, and generalizable network intrusion detection system (IDS) has become a research hotspot. In recent years, researchers have proposed a large number of intrusion detection methods based on traditional machine learning, deep learning, and multi-model fusion, and have made significant progress.

In order to improve the model's ability to express complex attack patterns, many studies have made innovative designs of the neural network structure. Geng Zhiqiang et al. [9] proposed an incremental intrusion detection model, incorporating a sparse self-attention mechanism, which significantly reduces computational complexity and maintains a good ability to memorize the old classes in a dynamic environment. Lu Haotian et al. [10] designed a continuous detection method based on a two-layer structure and indicator

distribution, which effectively improved the model's ability to recognize unknown classes. In terms of feature extraction, Tikhe et al. [11] constructed the FACS2ANet model by combining feature attention and a convolutional sparse self-encoder, which is suitable for intrusion detection in NFV virtual network environments. Jawad et al. [12], on the other hand, proposed the XGRU-IDS model, which fuses Extra Trees classifiers with GRU networks, and introduced the SHAP interpretation mechanism, thereby improving the interpretability of the model in industrial IoT environments. Deep reinforcement learning techniques are widely used in intrusion detection tasks with high adaptability and dynamically changing samples. Hu Yutao et al. [13] constructed an IoT-ONDDQN model based on the CIC-IoT dataset, introduced the Noisynet strategy to replace the traditional ε-greedy algorithm, realized a more exploratory training process, and improved the detection effect of malicious traffic in IoT. Jue Bo et al. [14], on the other hand, proposed an adaptive feature fusion mechanism and a metric learning strategy, which realized the effective identification of multiple classes of attacks under small samples. The automation of model structure design has also been investigated.

The automation of model architecture design has also attracted considerable research attention. Yang Yeming et al. [15] proposed the EMR-NID method to improve the model's generalization ability and adversarial robustness through multi-task robust architecture search, while Prabu et al. [16] combined the greedy sand cat swarm optimization algorithm with a dual attention graph convolutional network to construct an IDS architecture with strong expressive ability for complex network traffic features. Graph neural networks (GNNs) and sequence modeling techniques are widely used to capture structured and temporal information in network traffic. Specifically, GNNs excel at modeling the natural graph structure of network data, such as the connections between hosts, packets, or users, to detect sophisticated relational patterns that traditional methods might miss. Francis et al. [17] designed a feature delineation mechanism based on the firefly algorithm to enhance the robustness of attack detection in software-defined networks (SDNs). In the domain of GNNs, Sun et al. [18] proposed GNN-IDS, which incorporates both an attack graph and real-time measurements to represent complex computer networks. By learning network connectivity, their model can quantify the importance of neighboring nodes and features to make more reliable predictions and identify malicious actions causing anomalies. Lo et al. [19] presented E-GraphSAGE, a practical GNN approach that captures both edge features and topological information for intrusion detection in IoT networks using flow-based data. Their extensive evaluation showed that it outperforms state-of-the-art methods, demonstrating the strong potential of GNNs for NIDS. Additionally, Tran [20] addressed the challenge of preprocessing network data for GNN models by proposing an innovative method to extract relevant features from flow data to create nodes and edges, significantly enhancing IDS performance in detecting network attacks. Mohammed et al. [21] evaluated the performance of various recurrent neural network architectures and proposed introducing a DNN with an attention mechanism to enhance the model's recognition ability. Derui Guo et al. [22] constructed a TRBMA model that incorporates 1D-TCN, ResNet18, BiGRU, and a multi-head attention mechanism, achieving excellent detection results on the CIC-IDS2017 dataset. Congcong Li et al. [23], on the other hand, proposed the DSC-Inception-BiLSTM network, which uses a parallel structure to extract the spatial and temporal information of image and textual traffic features, respectively, to further improve the classification accuracy.

In resource-constrained IoT and industrial control systems, research focuses mainly on model lightweighting, accuracy, and multi-protocol compatibility. Mohamed et al. [24] constructed a CNN-GRU-LSTM three-layer model, which takes into account both spatial and temporal feature extraction in industrial CPS environments. Kalaivani et al. [25]

combined CNN with a restricted Boltzmann machine (RBM) to effectively improve the modeling capability of complex features, while Alabbadi et al. [26] proposed a cross-domain residual LSTM structure, incorporating an attention mechanism to adapt to the task of intrusion detection in heterogeneous IoT environments. Liu Liwei et al. [27] proposed an MRID model using a multi-scale residual temporal convolutional network with a traffic attention mechanism to achieve efficient real-time intrusion detection in fog computing architecture. Song et al. [28] constructed a TGA model that fuses TCN, BiGRU, and an attention mechanism to effectively capture temporal feature variations. Ling et al. [29] proposed a two-phase EGAT structure that introduces a graph attention mechanism to further enhance the edge information modeling capability. Mao et al. [30] constructed the FCN-Transformer architecture, which realizes the extraction of multi-level features from raw traffic.

In feature engineering, Qingyuan Peng et al. [31] used the gray wolf algorithm to optimize DBN-SVM, which significantly improves the classification performance on NSL-KDD and UNSW-NB15 datasets. Bouzaachane et al. [32] fused integrated learning and deep neural networks to construct a multilevel IDS model adapted to the modern complex network environment. Govindarajan et al. [33] proposed a cloud-based intrusion detection framework fusing graph features and contrast learning, which effectively improves the accuracy of the model in heterogeneous environments. Jessie Rani et al. [34] combined fuzzy integration and genetic optimization to maintain a high accuracy rate under small sample conditions. Farouqui et al. [35] proposed the SafetyMed system for IoMT, which combines CNN and LSTM to simultaneously cope with image and sequence intrusion data. Du et al. [36] constructed an MBConv-ViT model based on the fusion of Vision Transformer and MobileNet to effectively improve the IoT attack detection capability. Sun et al. [37] proposed an adversarial sample generation method based on the analysis of model simulation to improve the model robustness through adversarial training. Du Xinying et al. [38] proposed a multi-encoder feature fusion mechanism for the structural characteristics of the CAN bus to achieve accurate identification of injection attacks. Luo et al. [39] designed the MCLDM, a multi-channel contrastive learning network that combines supervised and unsupervised feature reconstruction strategies, which is suitable for feature modeling and classification of high-dimensional nonlinear traffic data. Neeraj Kumar et al. [40] proposed a multi-channel deep learning network for feature modeling and classification of high-dimensional nonlinear traffic data by introducing an improved deep learning structure and feature selection strategy, which demonstrated superior performance on multiple publicly available datasets.

Notably, with the development of distributed learning paradigms such as federated learning [41–43], intrusion detection research is expanding into privacy-preserving directions. Federated learning enables multiple devices to jointly train shared models without sharing raw data, thereby protecting data privacy [41]. However, this distributed learning paradigm is vulnerable to adversarial attacks, particularly data poisoning [41] and model poisoning attacks [42]. Research indicates that among data poisoning attacks like label flipping, feature poisoning, and VagueGAN, feature poisoning subtly degrades model performance by modifying high-impact features identified by random forest techniques [41]. Regarding model poisoning attacks, studies have found that federated networks exhibit strong robustness even when servers randomly adopt aggregation rules like Krum and Trimmed Mean in each federated learning round [42]. To enhance the security of federated learning systems, researchers have proposed various defense mechanisms. Randomized deep feature selection effectively mitigates the impact of feature poisoning attacks by randomizing server features of varying sizes [41]. Additionally, robust federated aggregation (RFA) methods, based on geometric median aggregation updates, demonstrate greater

resilience against potential poisoning of local data or model parameters. Variants include one-step robust aggregation and device-side personalized schemes [43]. Meanwhile, while large language models show promise in threat intelligence analysis, their computational overhead in real-time detection remains a consideration.

In summary, current intrusion detection technology exhibits trends of diversified model structures, fused feature modeling, and lightweight deployment methods. The research focus gradually shifts from traditional static identification toward dynamic adaptation, multi-source data fusion, and enhanced cross-domain generalization capabilities. In this paper, we use hybrid multi-model feature selection and kernel-based dimension reduction (HMK) to map the high-dimensional features of network traffic data into a low-dimensional space, achieving feature dimensionality reduction. Subsequently, the structured traffic data resulting from dimensionality reduction is transformed into visualized images using semantic features. The structured traffic data after dimensionality reduction is transformed into visualized images using semantic features. Then, we construct a detection model based on transfer learning and an ensemble convolutional neural network to perform network intrusion detection. The overall framework of the HMK-ST model is shown in Figure 1, which is mainly composed of four modules, namely, data preprocessing, feature selection and dimensionality reduction, traffic data visualization, and intrusion type detection. The data preprocessing module cleans, encodes, and normalizes the data; the feature selection and dimensionality reduction module maps the high-dimensional features into a low-dimensional space, enhances nonlinear separability, and balances accuracy and complexity; the traffic data visualization module transforms the structured traffic data into visual semantic images; the intrusion type detection module constructs an ensemble network model composed of ResNet50, VGG16, and Inception_v3. The sub-model fine-tuning is performed through transfer learning, and the static fusion strategy is utilized to assign the sub-model weights to perform multi-classification detection of network intrusions.

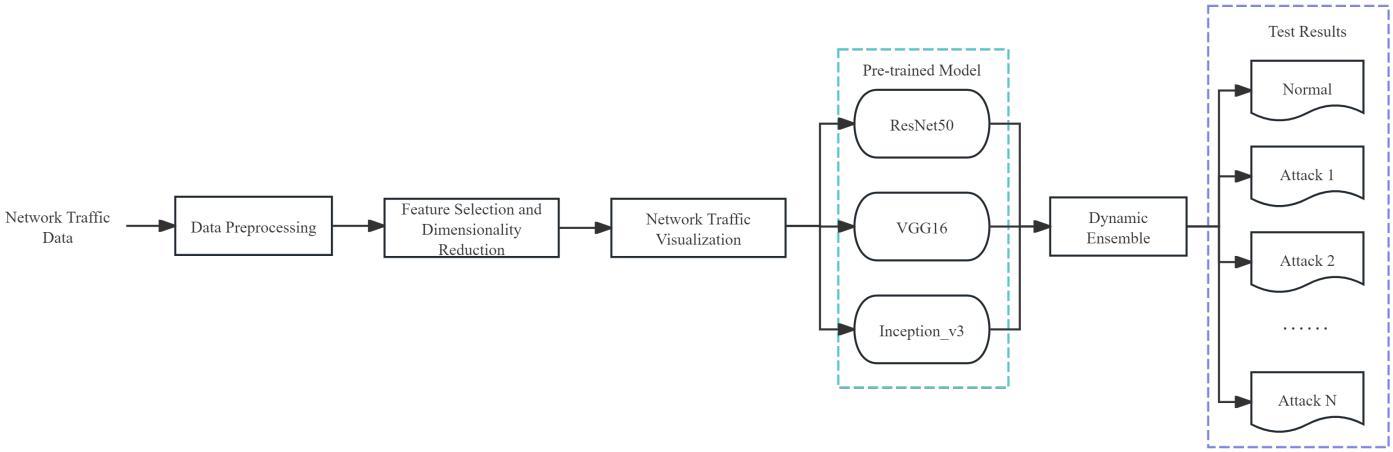

Figure 1. General framework of the HMK-ST model.

3. Methods and Techniques

This section details the proposed web intrusion detection (HMK-ST) model with feature dimensionality reduction, semantic mapping, and transfer learning.

3.1. Data Preprocessing

The data preprocessing module aims to remove redundant noise and irrelevant data from the dataset, unify feature representations, and align data distributions in order to avoid biases that can lead to degraded model performance. This module includes removing redundant data, feature coding, normalization, and feature space alignment processing.

In this paper, we use the UNSW-NB15 dataset and the CICIDS 2017 dataset. The UNSW-NB15 dataset has 82,332 test samples and 175,341 training samples, and each sample has 49 features [44]. It contains normal samples and eight types of attack samples—a total of nine types. The eight attack types are Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worm attacks. The CICIDS 2017 dataset contains several benign and the latest common cyber attacks. Note that the original dataset contains more than 2.8 million samples covering different time periods and multiple attack types, with a highly unbalanced sample distribution in which the percentage of normal traffic significantly exceeds that of attack traffic, resulting in a serious class imbalance problem. If left untreated, this imbalance can cause the model to become biased toward predicting normal samples, compromising attack detection performance. Therefore, to address this imbalance while maintaining representativeness and considering computational resources, we employ a sampling strategy that randomly selects 72,962 samples, including benign, denial-of-service (DoS), portscan, bruteforce, Webattack, Bot, and DDoS attacks, to ensure coverage of each attack category. The sampling strategy can improve the model's generalization and the model's performance. This sampling strategy aims to improve the model's generalization ability and performance by avoiding the interference of redundant data and mitigating the class imbalance during training.

Redundant data removal: Useless data is removed from the dataset. For example, the original fields "id", "time", and "lime" in the UNSW-NB15 dataset are record numbers or timestamps. These fields lack discriminative feature meaning, can introduce noise into the classification model, and affect the model's ability to learn effective patterns. Therefore, these fields are deleted. For the CICIDS 2017 dataset, an adaptive column detection mechanism is used to automatically identify the label column, which is then skipped during subsequent numerical transformation of the numerical features. Let the original dataset D_{raw} contain n samples and m features, and perform the following transformations:

$$D_{proc} = \phi(D_{raw}) = [X \mid y], X \in \mathbb{R}^{n \times m}, y \in \mathbb{R}^{n}, D_{raw} \in \mathbb{R}^{n \times (m+1)}, \tag{1}$$

where $\phi(\cdot)$ is the cleaning function of the data, identifying and labeling $\{x \mid x \in \{0, -\infty, \text{NaN}, ?\}\}$ as outliers for detection, implementing $f : \mathbb{R}^{\pi} \to \mathbb{R}^{\infty}$ mapping of the feature matrix for forced conversion of the type to ensure the consistency of numerical data. The HMK-ST model uses a list deletion method for missing values; in this method, the dataset D_{proc} removes all samples that contain missing values.

Categorical feature coding: The cleaned data are uniquely one-hot encoded to generate $k-1$-dimensional binary vectors (k is the number of categories) to avoid the introduction of multicollinearity and effectively retain key categorization information such as protocol type and service type. The FTP command count feature in the UNSW-NB15 dataset is encoded by thresholding it into a binary feature, i.e., setting it to 1 when its value is greater than 0 and setting it to 0 otherwise. This encoding method captures the activation state of the FTP session and eliminates the influence of the long-tailed distribution on the model.

The UNSW-NB15 dataset and the CICIDS 2017 dataset have different sampling sources, feature distributions, and coding methods, which, if not uniformly processed, will result in the misalignment of the input feature space in the training of the model, affecting the adaptability of the migration learning module. Therefore, the HMK-ST model uses the Z-score for the normalization of features:

$$z = \frac{x - \mu}{\sigma}, \tag{2}$$

where μ is the feature mean and σ is the standard deviation; the standardized data obeys the standard normal distribution to enhance the convergence speed of the model. In order

to ensure the consistency of the feature space between the training set $\mathbb{C}_{\text{train}}$ and the test set $\mathbb{C}_{\text{test}}$, the feature space is aligned as follows:

$$\mathbb{C}_{\text{common}} = \mathbb{C}_{\text{train}} \cap \mathbb{C}_{\text{test}} \tag{3}$$

3.2. Feature Selection and Dimension Reduction

With the increase in the dimensionality of network intrusion detection datasets, the spatial distribution of data becomes increasingly sparse and nonlinear, which makes it difficult for the model to capture effective patterns. At the same time, high-dimensional data increase the training time of the model and require a large amount of memory, so it is necessary to eliminate redundant features through feature selection and dimensionality reduction to extract key discriminative information from the high-dimensional data.

Based on this, this paper proposes a hybrid multi-model feature selection and kernel dimensionality reduction (HMK) algorithm, as shown in Figure 2, the flow chart. The HMK algorithm firstly constructs a hybrid multi-model feature importance function, which generates feature importance evaluation indexes by weighting the results of scoring the importance of features from different models to enhance the discriminative ability of the feature subset; then it dynamically selects the feature subset; finally, it constructs a kernel matrix and generalized feature decomposition to map the high-dimensional features to the low-dimensional space to achieve feature dimensionality reduction. Finally, the construction of the kernel matrix and generalized feature decomposition are carried out to map the high-dimensional features to the low-dimensional space to achieve feature dimensionality reduction.

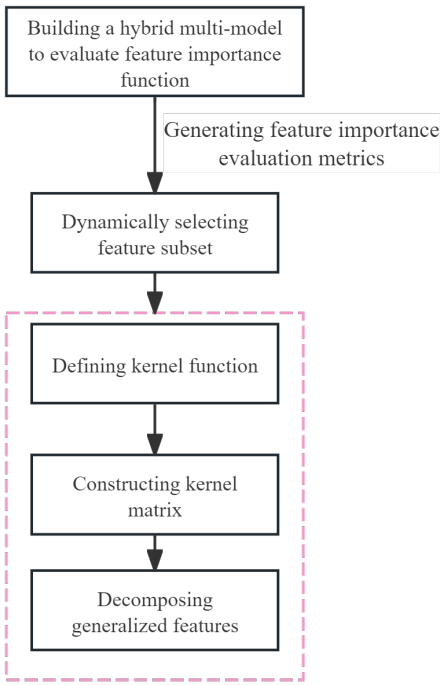

Figure 2. Flowchart of the hybrid multi-model feature selection and kernel dimension reduction-based algorithm.

3.2.1. Hybrid Multi-Model-Based Feature Importance Function

The feature importance function measures features that frequently split nodes or reduce classification impurity, features that contribute most to loss function optimization, and global nonlinear features that have statistical dependence on the attack labels, using the Gini importance of random forest (RF), the information gain importance of gradient

boosting tree (XGBoost, XGB), and the nonlinear correlation of mutual information (MI). Thus, the input training data matrix $X_{\text{train}} \in \mathbb{R}^{n\times m}$ is associated with the label vector $Y_{\text{train}} \in \mathbb{R}^{n}$, and the output feature importance function is

$$S_{\text{h}}(j) = \alpha\hat{S}_{\text{R}}(j) + \beta\hat{S}_{\text{X}}(j) + \gamma\hat{S}_{\text{M}}(j), \tag{4}$$

where the weight coefficients $\alpha + \beta + \gamma = 1$; j denotes features; $\hat{S}_{\text{R}}(j)$, $\hat{S}_{\text{X}}(j)$, and $\hat{S}_{\text{M}}(j)$ are the normalized values of the importance of Gini, denoted by $S_{\text{R}}(j)$, the importance of information gain, denoted by $S_{\text{X}}(j)$, and the nonlinear correlation of mutual information, denoted by $S_{\text{M}}(j)$. $S_{\text{R}}(j)$ and $S_{\text{X}}(j)$ are generated by random forest (RF) and XGBoost (XGB), respectively, for evaluating local feature interactions, and $S_{\text{M}}(j)$ is generated by the nonlinear correlation of mutual information for discovering global nonlinear features that are statistically dependent on the attack labels. RF and XGB models pay more attention to local feature interactions, and MI captures global nonlinear correlations, so the weight ratio is 4:4:2 in this paper. Due to the inconsistent dynamic range of the model metrics, min–max normalization is required before calculating the multi-model scores, which are normalized to the interval [0,1]:

$$\hat{S}_i(j) = \frac{S_i(j) - \min(S_i)}{\max(S_i) - \min(S_i)}, i \in \{\text{R}, \text{X}, \text{M}\} \tag{5}$$

1. Gini importance $S_{\text{R}}(j)$
   Random forest (RF) calculates the importance of feature j based on the weighting of the Gini impurity drop when the decision tree splits:

$$S_{\text{R}}(j) = \frac{1}{N_{\text{trees}}}\sum_{T}\sum_{t\in T}\left(\Delta Gini(t\,, j)\cdot\frac{n_t}{n}\right), \tag{6}$$

   where $\Delta Gini(t, j)$ denotes the Gini impurity reduction of feature j when node t splits, n_t is the number of node samples, and N_{trees} is the number of decision trees in the random forest.
2. Information gain importance $S_{\text{X}}(j)$
   In XGBoost (XGB), the importance of feature j is calculated by accumulating the information gain of each split point in the gradient boosted tree:

$$S_X(j) = \sum_{k}\sum_{s\in s_k(j)}\text{Gain}(s), \tag{7}$$

   where $s_k(j)$ denotes all the split points of feature j in the first k tree, and $\text{Gain}(s)$ is the cumulative value of the information gain, i.e., the reduced loss function value, of the split point, s.
3. Nonlinear correlation of mutual information $S_{\text{M}}(j)$
   Nonlinear correlation assessment using mutual information (MI) is computed by quantifying the mutual information between feature j and label y:

$$S_M(j) = \sum_{y\in\mathcal{Y}}\sum_{x_j\in\mathcal{X}_j} p(x_j, y)\log\frac{p(x_j, y)}{p(x_j)p(y)}, \tag{8}$$

   where $p(x_j, y)$ is the joint probability distribution of feature j and label y, $p(x_j)$ is the edge probability of feature j, and $p(y)$ is the edge probability of label y.

3.2.2. Dynamic Selection of Feature Subsets

The constructed hybrid multi-model evaluates the feature importance function and generates the feature importance evaluation index to rank the features. Then, dynamic

feature subset selection is performed—the objective function K is minimized via cross-validation to determine the optimal number of features k, and the objective function K is as follows:

$$K = \arg\max_k(\mathrm{Acc}_{\mathrm{CV}}(k) - \lambda \cdot k), \tag{9}$$

where λ is the complexity penalty factor, set to 0.01. Finally, the first k high-scoring features are selected to form the feature subset $X_{\text{selected}} \in \mathbb{R}^{n\times k}$, which balances the classification accuracy and the number of features to avoid overfitting.

3.2.3. Kernel Matrix-Based Feature Dimension Reduction

In order to further reduce the number of features, this paper combines feature similarity and label similarity to design the kernel matrix. Through the construction of the kernel matrix and the decomposition of generalized features, the high-dimensional features are mapped to the low-dimensional space to maximize the category differentiation and improve the nonlinear separability. Assuming the input feature subset X_{selected} with label Y_{train}, the output is a low-dimensional embedding $Z \in \mathbb{R}^{n\times d}$. First, the design of the kernel function is carried out, and the label information is introduced to guide the dimensionality reduction, where the label-weighted kernel function κ_s is defined:

$$\kappa_S(x_i, x_j) = \kappa(x_i, x_j) \cdot \phi(y_i, y_j), \tag{10}$$

Among them, x_i and x_j belong to the feature subset X_{selected}; y_i and y_j belong to the label set Y_{train}; uses a radial basis function (RBF) kernel to capture the nonlinear structure of the feature space, mapping the original features to the high-dimensional kernel space and thereby solving the linear indivisibility problem [45]:

$$\kappa(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right), \tag{11}$$

Label similarity function $\phi(y_i, y_j)$ for enhancing the similarity of like samples and suppressing the similarity of dissimilar samples:

$$\phi(y_i, y_j) = \begin{cases} 1 + \exp\left(-\frac{\|y_i - y_j\|^2}{2\tau^2}\right) & \text{if} y_i = y_j \\ \exp\left(-\frac{\|y_i - y_j\|^2}{2\tau^2}\right) & \text{otherwise} \end{cases} \tag{12}$$

The parameter τ controls the decay rate of the similarity of the samples within the class, which strengthens the aggregation of the similar samples in the kernel space and weakens the correlation of the dissimilar samples through the labeling information.

Next, the kernel matrix is constructed, and the kernel matrix $K_S \in \mathbb{R}^{n\times n}$ is computed, and its elements are as follows:

$$K_s(i, j) = \kappa_s(x_i, x_j), \tag{13}$$

Since $\phi(y_i, y_i)$ is non-negative symmetric for the similarity function (15), and according to the Schur product theorem, the Hadamard product of two positive definite matrices remains positive definite, the kernel matrix K_s is also positive definite.

A generalized eigen-decomposition is performed using K_s, and the following generalized eigen-equation is solved to obtain the projection matrix V:

$$K_s V = \Lambda D V, \tag{14}$$

where D is the degree matrix, the diagonal elements are $D_{ii} = \sum_j K_s(i,j)$, Λ is the eigenvalue diagonal matrix, and the eigenvectors corresponding to the first k eigen subsets are selected to form the projection matrix $V \in \mathbb{R}^{K \times k}$. Finally, the low-dimensional space mapping is performed to project the original data to the low-dimensional space:

$$Z = X_{\text{selected}} V, \tag{15}$$

The complexity of the feature selection and dimensionality reduction module mainly comes from the computation of the hybrid multi-model feature-importance evaluation function and the dimensionality reduction of the kernel matrix. The time complexity of evaluating the feature importance function is given by

$$O(m \cdot (n_{\text{trees}} \cdot n \log n + n_{\text{boost}} \cdot n)), \tag{16}$$

where m is the number of features, n is the number of samples, n_{trees} is the number of trees in the random forest, and n_{boost} is the number of boosting rounds in XGBoost. The time complexity of the kernel dimensionality reduction stage is given by the following:

$$O(n^3 + Kn^2), \tag{17}$$

The main sources are the construction of the kernel matrix $O(n^2K)$ and the decomposition of generalized features $O(n^3)$, where k is the number of selected features. The term $O(n^2K)$ accounts for the construction of the label-guided kernel matrix $K_S \in \mathbb{R}^{n \times n}$, which requires computing pairwise similarities for all samples using a kernel function over k-dimensional features. The term $O(n^3)$ corresponds to the dense generalized eigen-decomposition of the kernel matrix, which is typically solved via standard numerical methods such as the QR algorithm or Lanczos iteration for large matrices.

3.2.4. Feature Selection Results and Final Feature Set

As described in Section 3.2.2, the optimal number of features k was determined by minimizing the objective function K (Equation (9)) through 5-fold cross-validation. The value of the penalty factor λ was empirically set to 0.01 after a sensitivity analysis showed model performance was robust within a range of [0.005, 0.02].

For the UNSW-NB15 dataset, the cross-validation process determined that $k = 28$ features provided the optimal balance between accuracy and complexity. The top 10 most important features selected, along with their normalized hybrid importance scores, are listed in Table 1.

For the CICIDS 2017 dataset, the dynamic selection process identified $k = 35$ features as optimal. The top 10 features are listed in Table 2.

The results align well with known network attack characteristics. For instance, for both datasets, features related to flow duration, packet timing (e.g., sload, dload), and protocol-specific counts (e.g., ct_srv_src) consistently ranked highly across all three individual metrics (RF, XGB, MI) and, thus, received high composite $S_{\text{h}}(j)$ scores. This consensus among different selection methods reinforces that these features are critically discriminative.

The final selected feature subsets X_{selected} for each dataset, comprising 28 and 35 features, respectively, were then passed to the subsequent kernel-based dimensionality reduction module.

Table 1. Top 10 features selected for the UNSW-NB15 dataset by hybrid importance score $S_{\mathrm{h}}(j)$.

Rank	Feature Name	Normalized $S_{\mathbf{h}}(j)$
1	spkts	1.00
2	dpkts	0.98
3	sbytes	0.97
4	dbytes	0.96
5	dur	0.94
6	sload	0.93
7	dload	0.92
8	ct_srv_src	0.89
9	ct_state_ttl	0.87
10	sjit	0.85

Table 2. Top 10 features selected for the CICIDS 2017 dataset by hybrid importance score $S_{\mathrm{h}}(j)$.

Rank	Feature Name	Normalized $S_{\mathbf{h}}(j)$
1	Flow Duration	1.00
2	Total Fwd Packets	0.99
3	Total Length of Fwd Packets	0.98
4	Fwd Packet Length Max	0.96
5	Flow Bytes/s	0.95
6	Flow IAT Mean	0.93
7	Fwd IAT Total	0.90
8	Destination Port	0.88
9	ACK Flag Count	0.86
10	Subflow Fwd Bytes	0.84

3.3. Network Traffic Visualization Based on Semantic Feature Mapping

In this paper, we propose a structured data visualization method based on semantic feature mapping. By encoding the dimensionality-reduced features, the spatial and temporal characteristics of cyberattacks (burstiness, periodicity, covertness) are transformed into interpretable visual patterns (stripes, noise, etc.) to establish a "feature–image–semantic" mapping bridge. The structured network features are visualized as 32 × 32-pixel color images using multiple channels of color images to carry different features, such as the R channel to carry traffic intensity, the G channel to carry the distribution of protocol types, and the B channel to carry time-series characteristics.

The features are first locally normalized using quartiles to avoid the effect of outliers, and the feature vector $f \in \mathbb{R}^d$ after dimensionality reduction by the HMK algorithm is adaptively normalized:

$$f' = \frac{f - Q_1(f)}{Q_3(f) - Q_1(f) + \epsilon'}, \tag{18}$$

where Q_1 and Q_3 are the $\frac{1}{3}$th quartiles of the feature values, and $\epsilon = 10^{-8}$ prevents division by zero. This method retains the intra-class distribution property compared to global normalization. Secondly, we dynamically select the image generation strategy according to the attack type labels, establish a mapping relationship between feature statistics and visual patterns, and design differentiated coding strategies for several types of attack behaviors in the dataset. For normal traffic, we use a uniform green field, and the brightness of the green channel is positively correlated with the smoothness of the traffic.

$$I_{i,j,1} = \left\lfloor 255 \cdot \frac{1}{n} \sum_{k=1}^{n} f_k \right\rfloor, \tag{19}$$

where I represents the brightness value of the green channel, and f_k is the normalized eigenvalue. Based on the standard deviation distribution of normal traffic and attack traffic in this paper's dataset, when the traffic rate $\sigma_t < 5$, it is judged as normal traffic. The high brightness of the generated image corresponds to low-traffic fluctuations, whereas low brightness indicates large fluctuations. When $\sigma_\iota \geq 5$, it is classified as a DoS attack; for a DoS attack, the red channel intensity I is used to encode the traffic burst.

$$I_{i,j,0} = \left\lfloor 255 \cdot \sqrt{\frac{1}{n}\sum_{k=1}^{n}(f_k - \overline{f})^2} \right\rfloor, \tag{20}$$

For port scanning attacks, a horizontal blue gradient is constructed at I to characterize scanning persistence:

$$I_{i,j,2} = \left\lfloor 255 \cdot \frac{\sum_{k=1}^{j} f_k}{\sum_{k=1}^{W} f_k} \right\rfloor, \tag{21}$$

The cumulative effect of the gradient reflects the scanning process, and the horizontal position j is linearly correlated with the time dimension. At the same time, based on the spatial and temporal characteristics of the attack behavior, we define the rule for generating geometric patterns—for Exploit attacks, the attack stripe spacing Δx is inversely proportional to the frequency of the attack f_{req}.

$$\Delta x = \left\lfloor \frac{4}{f_{req} \,/\, f_{max}} \right\rfloor, \tag{22}$$

where $f_{\max} = 128$ Hz is the maximum observed frequency in the dataset; a high-frequency attack with $f_{\text{req}} > 64$ Hz generates dense stripes ($\Delta x = 2$); and the Worm attack generates Archimedean spirals in the polar coordinate system.

$$r = a + b\theta, \tag{23}$$

where $a = 5$ and $b = 0.3$, the spiral density is positively correlated with the propagation speed, and the difference between neighboring pitches $\Delta r = 2\pi b$ is set to $b = 0.3$ to match the typical worm propagation pattern.

Finally, the texture enhancement method is generated by a noise injection strategy combined with a frequency-domain enhancement technique. To enhance the robustness of the model to feature perturbations, adaptive noise is introduced—the noise variance is correlated with the feature dispersion in the Fuzzers attack:

$$\sigma_{\text{noise}} = 25 \cdot \frac{\sigma(f)}{\sigma_{\max}}, \tag{24}$$

where σ_{max} is the maximum standard deviation of the training set, and the noise obeys the $\mathcal{N}(128f, \sigma^2_{\text{noise}})$ distribution; in the Backdoor attack, the speckle density reflects the strength of covert communication

$$\lambda = 50 \cdot \frac{\sum f_k}{\sum f_{k,max}}, \tag{25}$$

Meanwhile, the generation process follows the Poisson point process:

$$P(N(B) = k) = \frac{e^{-\lambda|B|}(\lambda \mid B \mid)^k}{k!}, \tag{26}$$

where B is the image sub-region and $|B| = 8 \times 8$ pixels. In the frequency-domain enhancement technique, the attack features are embedded in the frequency domain by the Fourier transform. A motion-fuzzy kernel is designed to simulate traffic flooding in the DDoS attack:

$$K = \frac{1}{k}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}_{k \times k}, \quad (27)$$

The kernel size k and the attack duration T satisfy $k = \lfloor T/5 \rfloor$, with a maximum limit of $k_{\max} = 15$. Web attacks preserve the mid-frequency component to highlight SQL injection features:

$$H(u,v) = \begin{cases} 1 & \text{if } D_0 - W/2 \le D(u,v) \le D_0 + W/2 \\ 0 & \text{otherwise} \end{cases}, \quad (28)$$

The center frequency $D_0 = 0.3N$, and the bandwidth $W = 0.2N$, where N is the image width. Based on the above algorithm, the obtained partial 32 × 32 pixel color images are shown in Figure 3. From the figure, it can be seen that the sample types generate different color images.

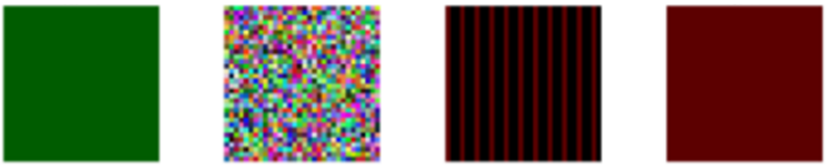

Figure 3. A visualization of network traffic characteristics by type.

3.4. Network Intrusion Detection Model Based on Migration Learning and Integrated Convolutional Neural Networks

The network intrusion detection model based on migration learning and an integrated convolutional neural network is shown in Figure 4. First, 32 × 32-pixel images are upsampled to 128 × 128 pixels by bilinear interpolation, and by fusing the deep features of three pre-trained models—ResNet50, VGG16, and Inception_v3—and adopting the dynamic weighted integration strategy, the efficient feature extraction and multi-classification detection of network traffic data are realized.

The dataset pre-trained ResNet50, VGG16, and Inception_v3 models are used as the base feature extractors in the feature extraction module, and their convolutional layer parameters are retained to capture generalized image features. The ResNet50 subnetwork has a 50-layer residual network, which achieves feature multiplexing through skip connections and is good at capturing deep features; the VGG16 subnetwork adopts a consecutive 3 × 3 small convolutional kernel cascade strategy, whose receptive field is equivalent to a 7 × 7 large kernel but with 44% fewer parameters; the Inception_v3 subnetwork introduces a parallel multi-branch structure to achieve multi-scale feature fusion within a single layer and enhance spatial perception.

A static weight fusion strategy is proposed in the dynamic integration module to balance the predictive contributions of each subnetwork, and the optimal weight allocation is determined through experimental validation:

$$y_{ensemble} = 0.4 \cdot y_{ResNet} + 0.3 \cdot y_{VGG} + 0.3 \cdot y_{Inception}, \quad (29)$$

ResNet50 performs optimally in deep feature extraction due to the residual structure's mitigation of the gradient vanishing problem and is given the highest weight. VGG16

is prone to overfitting in scenarios with small amounts of data due to high parameter redundancy, and Inception_v3 input interpolation introduces noise and has a lower weight.

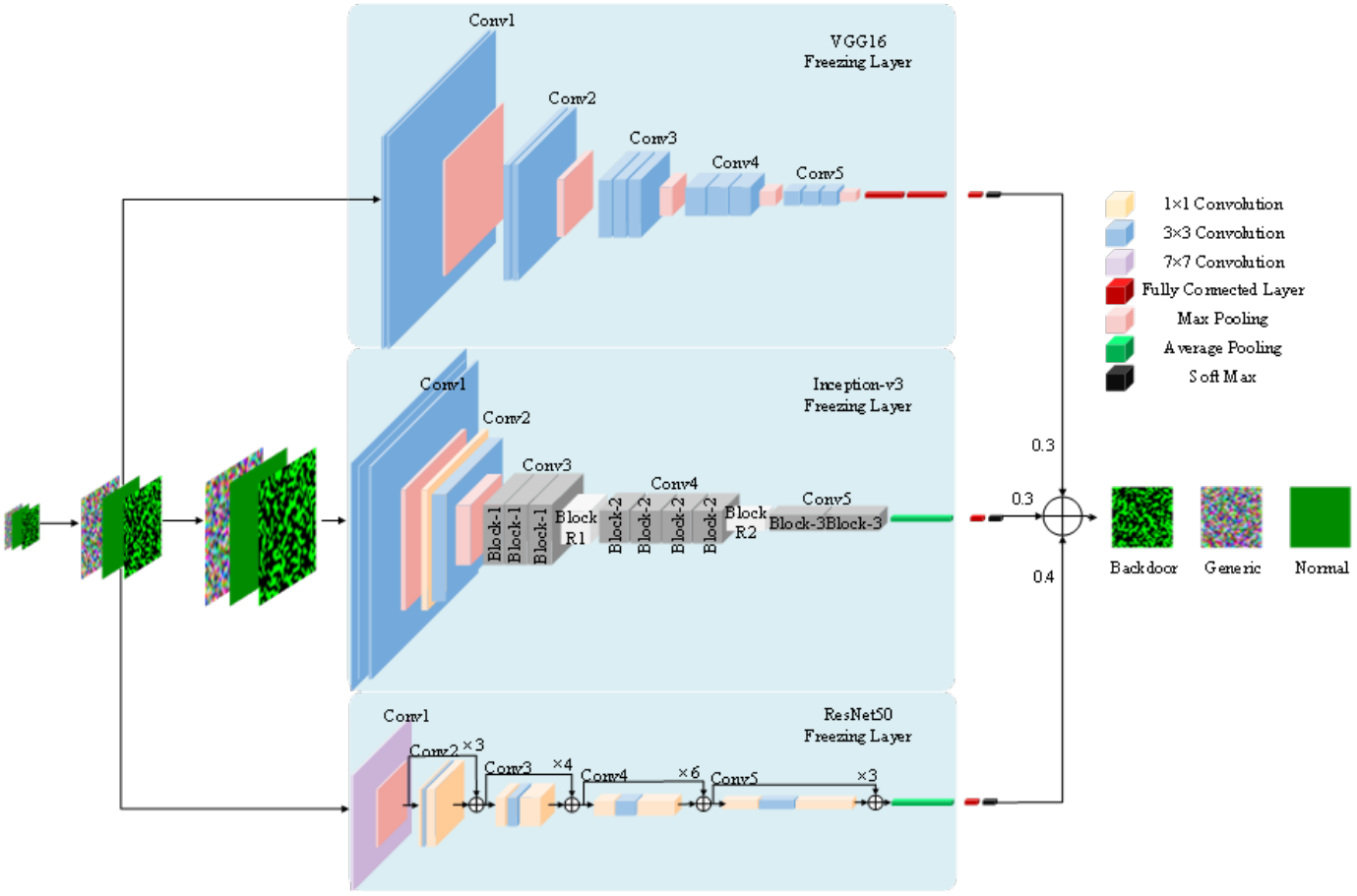

Figure 4. Network intrusion detection model based on transfer learning and integrated convolutional neural networks.

In the model training phase, the feature reuse strategy for the ResNet50 subnetwork is to freeze all the convolutional layer parameters from conv1 to conv5 (23.5M in total) in the model, preserving the generalized edge–texture–semantic feature extraction capability learned on the dataset. The feature enhancement mechanism for the VGG16 subnetwork extracts multi-scale features progressively through five convolutional blocks (2–2–3–3–3 structure), freezing the convolutional blocks and two fully connected layer parameters (14.7M), and retaining its high-resolution, fine-grained feature capture capability. The original input of the Inception_v3 subnetwork requires a resolution of 299 × 299, and the present design upsamples the 128 × 128 inputs to the target size through bilinear interpolation. Aimed at the domain characteristics of the network intrusion detection task, a parameter unfreezing strategy is designed. The strategy freezes the parameters of the feedforward layer of each model, and only unfreezes and fine-tunes the fully connected layer, whose dimension is adjusted to N, the dimensional classification space

$$f_{fc}^{(i)}(x) = W^{(i)}x + b^{(i)}, i \in \{ResNet, VGG, Inception\} \quad (30)$$

4. Experimental Analysis

In this paper, we validate the performance of the HMK-ST model using the UNSW-NB15 dataset and the CICIDS 2017 dataset. PyTorch 1.9.1 GPU version serves as the core framework, integrating pre-trained models and customized datasets for model loading and training. At the same time, the dataset is divided into a training set, a test set, and a validation set. Moreover, 70% of the dataset is used to train the model, 15% is used to test the model, while the remaining 15% is used to validate the performance of the HMK-ST model.

4.1. Experimental Environment and Related Parameters

PyCharm version 2024.2.4. was used to write the testing program, and the computer parameters—an Intel sixteen-core processor (i7-14650HX 2.2 GHz), a 64-bit operating system, and Windows 11. The HMK-ST model is an integrated convolutional neural network, weighted and fused from three pre-trained models—ResNet50, VGG16, and Inception_v3—with fine-tuning of the fully connected layer of each model adapted to different classification tasks ($N = 10$ for the UNSW-NB15 dataset and $N = 16$ for the CICIDS 2017 dataset). The input layer receives 128 × 128-pixel images transformed from traffic statistics features, with weight coefficients in the weighted fusion strategy of 0.4:0.3:0.3. The training parameters are as follows: Use the Adam optimizer, set the initial learning rate $\eta_0 = 0.001$ and a learning rate update strategy for ReduceLROnPlateau; monitor the validation set accuracy and dynamic learning rate adjustment. When the indicator stagnates for 2 epochs, the learning rate is decayed to 0.1 times the original value. Set the early stopping mechanism—terminate training when the validation loss does not decrease for 5 consecutive epochs to prevent overfitting. The batch size is 16.

4.2. Performance Indicators

Accuracy, precision, recall, and F1 score are used as evaluation metrics [46]. TP denotes true positive, which indicates the number of samples that are actual and predicted to be normal; TN denotes true negative, which indicates the number of samples that are actual and predicted to be attacks; FP denotes a false positive, which indicates the number of samples that are actually attacks and predicted to be normal; FN denotes a false negative, which indicates the number of samples that are actually normal and predicted to be attacks. Accuracy indicates the ratio of the number of correctly categorized samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{31}$$

Unlike the accuracy rate, the precision rate indicates the ratio of samples correctly predicted as normal to all samples predicted as normal:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{32}$$

Recall is dedicated to assessing the ability of the detection model to classify samples as normal and is equal to the ratio of samples correctly predicted as normal to all samples that are actually normal:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{33}$$

Considering only the precision and recall metrics is somewhat one-sided, and the F1 score metric is also chosen. The F1 score fuses precision and recall, and is defined as shown in the equation. The larger the value of the F1 score, the better the comprehensive performance of the model's classification.

$$\text{F1-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{34}$$

4.3. Network Traffic Visualization Results

The HMK-ST model converts the dimensionality-reduced structured traffic data into 32 × 32-pixel color images with significant differentiation through the HMK algorithm, as shown in Figure 5. It can be seen that due to the differences in spatial and temporal characteristics of different attack types, the generated images are strongly discriminative in terms of the color channels, geometrical patterns, and texture features. The HMK algorithm converts the suddenness, persistence, and covertness of the attacks into visual patterns discernible by the human eye, laying an interpretable visual foundation for the subsequent intrusion detection model. The HMK algorithm transforms the sudden, persistent, and stealthy attacks into visual patterns that are recognizable to the human eye, laying an interpretable visual foundation for the subsequent intrusion detection models. The images generated by the HMK algorithm maximize the differences between classes, such as the uniform green color of the Normal and the red patch of the DoS attack, and maintain consistency within the classes, such as the stability of the curvature parameter of the Worm helix.

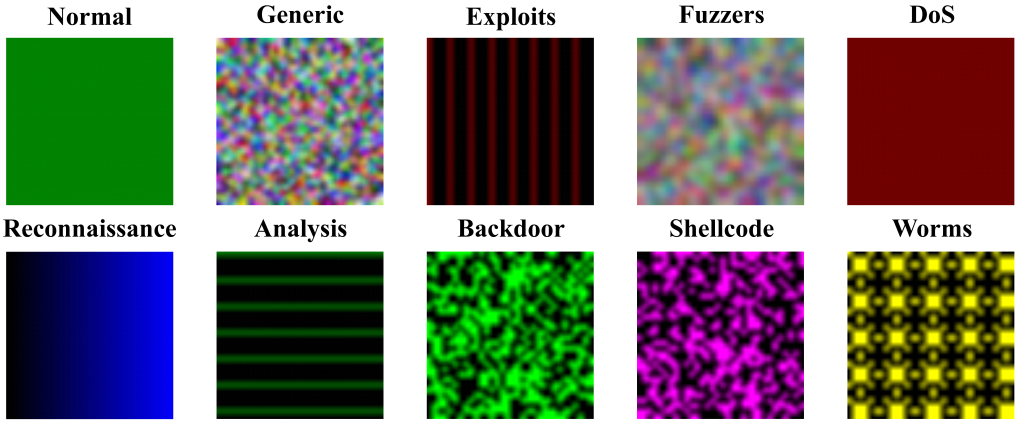

Figure 5. Network traffic characterization visualization.

4.4. Performance Analysis Based on the UNSW-NB15 Dataset

This section evaluates HMK-ST performance on the UNSW-NB15 dataset, comparing it with models from [47–49], selected for comparison.

Tables 3 and 4 show the prediction performance of HMK-ST and comparative models on UNSW-NB15. The results show that the HMK-ST model performs superiorly in the global classification task, with an overall accuracy of 99.99%, which is about 1.3% higher compared to 98.68% in the ResNet50 network model, significantly enhancing the global classification ability; moreover, the recall and F1 score of the HMK-ST model are both close to 100%, which indicates that the model in the attack detection process rarely has leakage detection and false alarms, especially in high-risk scenarios with high reliability.

The HMK-ST model effectively solves the underutilization of nonlinear structures and label information in high-dimensional data by the weighted fusion of global and local feature importance extracted by RF, XGB, and MI methods (4). Meanwhile, the label similarity constraint is introduced with the help of the kernel matrix construction (15) to enhance intra-class aggregation, so that the discriminative properties of low-dimensional

features are significantly better than those of traditional dimensionality reduction methods. Meanwhile, HMK-ST also converts structured network traffic into an image form for processing. For example, DoS attacks show red burst patches, Exploit attacks form vertical stripes, and port scans show a continuous pattern of blue gradients. In this way, intrusion detection not only relies on raw data features but also captures spatiotemporal patterns that are difficult to express using traditional statistical methods, thereby effectively improving detection accuracy.

This experiment also verifies the performance of the integrated network. From Table 4, we can see that there are limitations in the feature extraction capability of a single model, and the F1 score of 92.68% using only ResNet50 is much lower than the 99.98% of HMK-ST, which indicates that it is difficult for a single network architecture to comprehensively capture complex attack features. Despite the high precision of the VGG16 model, its relatively low recall rate reflects the lack of clear classification boundaries, which may lead to unstable identification of some attack categories. In addition, ResNet50 is not sensitive enough to temporal features and has a recall rate of 95.69% on this dataset, while VGG16 is prone to overfitting problems due to parameter redundancy and has an overall accuracy rate of only 94.94%.

Table 3. Predictive performance of the HMK-ST model on the UNSW-NB15 dataset.

Model	Accuracy	Precision	Recall	F1 Score
ENFS (2023) [47]	79.70	96.70	77.30	85.90
gmm-wgan-ids (2024) [48]	87.70	88.46	87.70	85.40
NAEF (2025) [49]	85.70	80.20	87.30	88.30
HMK-ST	**99.99**	**99.99**	**99.97**	**99.98**

Table 4. Performance Comparison Between the Single Model and the HMK-ST Integrated Model on the UNSW-NB15 Dataset.

Model	Accuracy	Precision	Recall	F1 Score
ResNet50	98.68	97.42	95.69	92.68
VGG16	94.94	98.89	95.74	97.29
Inception_v3	85.70	98.34	95.56	96.91
HMK-ST	**99.99**	**99.99**	**99.97**	**99.98**

The confusion matrix of the HMK-ST model in classifying 10 sample types from the UNSW-NB15 dataset is shown in Figure 6. All 37,000 samples of the Normal type are correctly classified with 100% recall and no missed detection, but three cases of the DoS attack type are misclassified as Normal. Three cases of the Fuzzers attack type are misclassified as Generic, reflecting the similarity of the attack features that lead to boundary blurring; three cases of the DoS attack type are misclassified as Normal, probably because the statistical characteristics of the attack traffic during a specific time period overlap with the normal traffic, which leads to confusion in the model classification; one case of the Backdoor attack type is misclassified as Generic, which indicates that the model needs to further enhance its ability to differentiate covert communication patterns to improve detection accuracy.

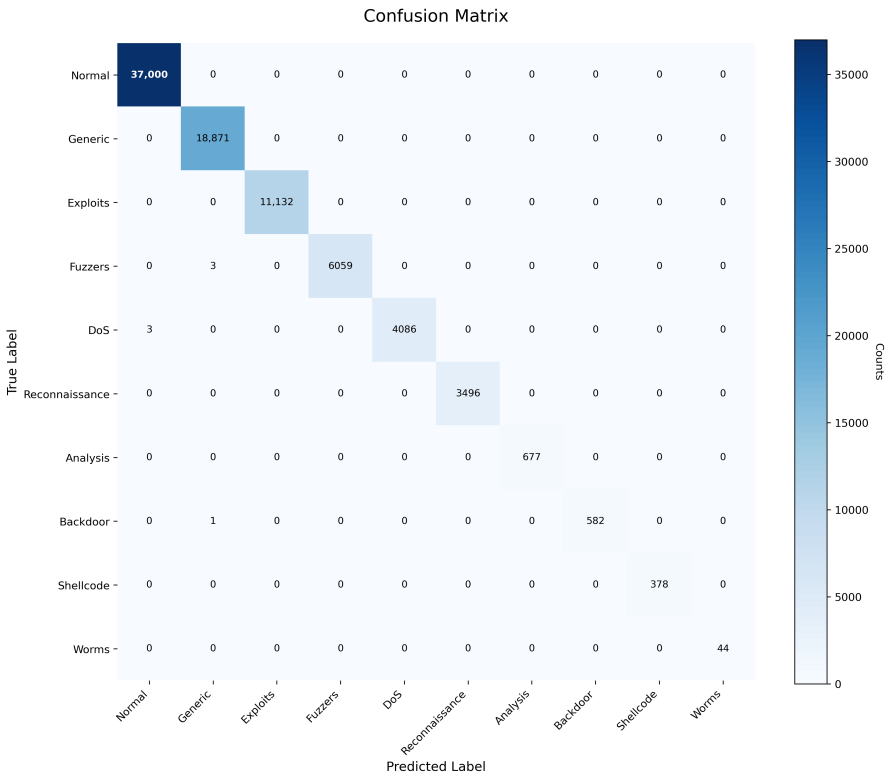

Figure 6. Confusion matrix for the HMK-ST model on the UNSW-NB15 dataset.

4.5. Performance Analysis Based on the CICIDS 2017 Dataset

In this section, the performance of HMK-ST is verified using the CICIDS 2017 dataset, which is selected for comparison with the intrusion detection models proposed in the literature [47,50,51].

The prediction performance of the HMK-ST model on the CICIDS 2017 dataset is shown in Tables 5 and 6, from which it can be seen that the HMK-ST model significantly outperforms the other models in terms of accuracy and F1 score, and the recall rate is as high as 99.85%, which suggests that there is almost no leakage of detection in this model. Compared with individual deep learning models, ResNet50 performs the worst, with an accuracy of only 72.22%, probably because its residual structure is not adapted to the temporal characteristics of network traffic data, resulting in insufficient feature extraction capability. Inception_v3 has a higher F1 score, but the accuracy and recall are still significantly lower than HMK-ST, indicating that it is difficult for a single architecture to comprehensively catch the complex attack patterns. HMK-ST's accuracy of 99.96% is much higher than ResNet50's 72.22% and VGG16's 88.96%, indicating that the model still maintains a stable classification in high-dimensional data; and the model's recall of 99.85% suggests that it misses very few detections, which is suitable for intrusion detection needs in high-risk scenarios (e.g., finance, healthcare); the F1 score of 99.91% suggests that Inception_v3 has a higher F1 score than HMK-ST, indicating that the single architecture is difficult to comprehensively capture complex attack patterns. A score of 99.91% indicates that the model achieves an efficient balance between precision and recall.

Table 5. Predictive performance of the HMK-ST model on the CICIDS 2017 dataset.

Model	Accuracy	Precision	Recall	F1 Score
ENFS (2023) [47]	92.90	96.90	95.81	96.35
SABiLSTM (2024) [50]	96.92	95.15	96.09	95.62
SVM + RBFN (2024) [51]	95.68	94.71	96.89	95.79
HMK-ST	**99.96**	**99.97**	**99.85**	**99.91**

Table 6. Performance Comparison Between the Single Model and the HMK-ST Integrated Model on the CICIDS 2017 dataset.

Model	Accuracy	Precision	Recall	F1 Score
ResNet50	72.22	66.34	80.72	69.66
VGG16	88.96	87.27	91.61	87.34
Inception_v3	94.37	93.38	92.84	96.17
HMK-ST	**99.96**	**99.97**	**99.85**	**99.91**

The confusion matrix of the HMK-ST model in classifying 16 sample types from the CICIDS 2017 dataset is shown in Figure 7. Twenty-eight samples of the DDoS attack type are misclassified as DoS Slowloris attacks, with a recall of 0.9768, because the traffic-bursting characteristic of DDoS and the low-speed, sustained attack of DoS Slowloris partially overlap in the statistical characteristics (such as the average traffic rate).

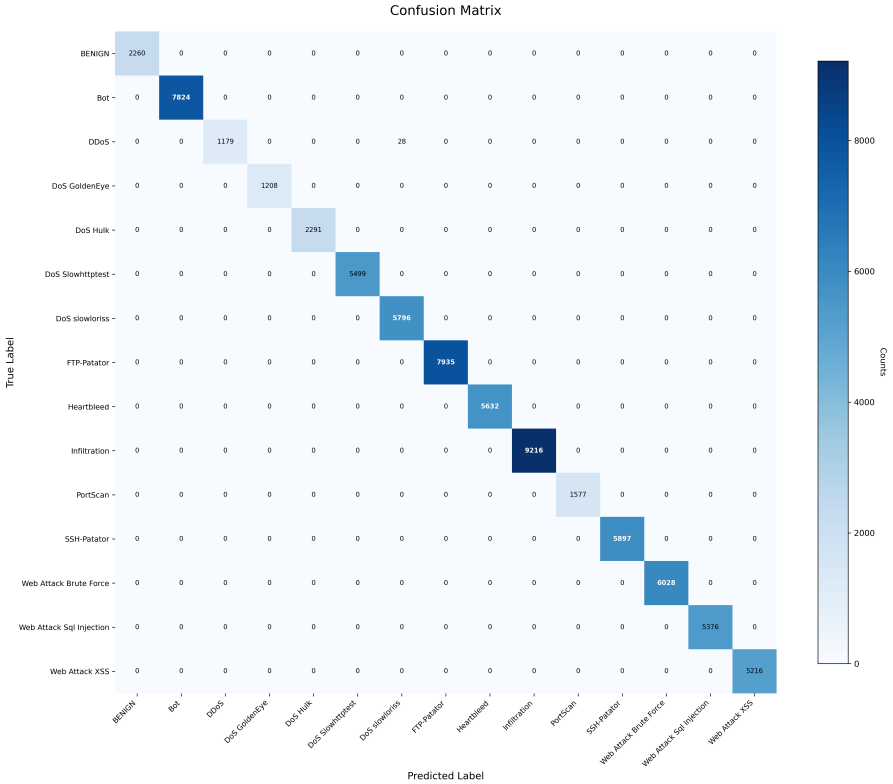

Figure 7. Confusion matrix for the HMK-ST model on the CICIDS 2017 dataset.

4.6. Analysis of the Weighted Fusion Strategy for the HMK-ST Model

In order to analyze the effect of weights on performance in a dynamic weighting strategy, the performance of the algorithm under different weights is tested in this paper. First, using the average weight as the baseline, four sets of weight combinations are generated by grid search and evaluated on the UNSW-NB15 dataset and the CICIDS

2017 dataset, respectively. The evaluation results are shown in Tables 7 and 8, where the weight combinations are denoted as $W(1)$:$W(2)$:$W(3)$, where $W(1)$ denotes the weights for ResNet50, $W(2)$ denotes the weights for VGG16, and $W(3)$ denotes the weights for Inception_v3. The experimental results show that when the weight combination is 0.3:0.4:0.3 in CICIDS 2017, its accuracy is 88.79%, which indicates that over-reliance on VGG16 may lead to a decrease in the model's adaptability to complex attack modes (e.g., DDoS, web attacks); when the weight of ResNet50 is high, the F1 score of UNSW-NB15 reaches 99.53%, but is slightly lower on CICIDS 2017, indicating that the feature extraction ability of ResNet50 varies across datasets. The combination of a high weight for ResNet50 and a medium weight for Inception_v3 can effectively capture the fine-grained features of protocol fields, which can improve the detection accuracy of complex attacks.

On the UNSW-NB15 and CICIDS 2017 datasets, the F1 score is improved by 0.58% and 0.51% when the ratio of the combined weights of ResNet50, VGG16, and Inception_v3 is 0.4:0.3:0.3, which is better than the average integration. This verifies that the dynamic weighting strategy can optimize inter-model feature complementarity through weight allocation and improve classification performance.

Table 7. Dynamic weighting strategy for ablation experiments on the UNSW-NB15 dataset.

Weighting Combination	Accuracy	Precision	Recall	F1 Score
Average Integration	99.99	99.99	98.86	99.40
0.3:0.3:0.4	98.86	98.24	97.96	98.89
0.5:0.3:0.2	99.99	99.99	99.07	99.53
0.3:0.4:0.3	98.32	98.58	98.04	98.29
0.4:0.3:0.3	99.99	99.99	99.97	99.98

Table 8. Dynamic weighting strategy for ablation experiments on the CICIDS 2017 dataset.

Weighting Combination	Accuracy	Precision	Recall	F1 Score
Average Integration	99.46	99.46	98.34	99.40
0.3:0.3:0.4	98.94	98.93	98.83	98.88
0.5:0.3:0.2	99.54	99.03	99.39	99.26
0.3:0.4:0.3	88.79	96.55	93.95	92.33
0.4:0.3:0.3	99.96	99.97	99.85	99.91

5. Conclusions

Driven by 5G and other ultra-high-speed network technologies, the complex environment of the Internet of Everything makes network attack methods increasingly diverse, and traditional detection methods face the severe challenges of high-dimensional data and nonlinear features. The HMK-ST model proposed in this paper not only achieves anomaly detection of network traffic, but also accurately classifies multiple types of attacks (e.g., DoS, port scanning) by integrating transfer learning and semantic visualization techniques. For high-dimensional noisy data, the model combines hybrid multi-evaluation feature selection with a supervised kernel dimensionality reduction algorithm (HMFS-SKDR) to effectively extract key features and enhance nonlinear differentiability; meanwhile, it converts structured traffic into discriminative images through semantic feature mapping, which significantly improves the pattern recognition capability of convolutional neural networks. On the UNSW-NB15 and CICIDS 2017 datasets, which are close to real scenarios, the accuracy of HMK-ST reaches 99.99% and 99.96%, and the F1 scores are 99.98% and 99.91%, respectively, which verifies its high efficiency and robustness in complex attack detection. However, the computational complexity of the model is high, which may affect

real-time performance, and the generalization ability to unknown attacks still needs to be strengthened. In future work, the model will focus on lightweight architectural design and incremental-learning strategies to further enhance its adaptive defense capability in a dynamic network environment.

Although the HMK-ST model demonstrates outstanding performance on two benchmark datasets, it still has certain limitations.

First, the computational complexity of the model poses a potential bottleneck for deployment on resource-constrained edge devices or in network environments with stringent real-time requirements. Future research will focus on model lightweighting and optimization. By adopting advanced techniques such as pruning, quantization, and knowledge distillation, the model's parameter count and computational overhead can be significantly reduced. This will enhance detection speed, lower energy consumption, and improve practicality.

Second, current models require strengthened defense capabilities against unknown zero-day attacks. While their core strength lies in fine-grained classification of known attack types, their potential to detect unknown attacks as anomalies is constrained by predefined visualization mapping rules. Therefore, a key future direction is developing adaptive or learning-based visualization mechanisms that can automatically discover optimal visual representations for novel attack patterns without human intervention. Advanced methods such as generative models or attention mechanisms may play a pivotal role in this endeavor. Simultaneously, we will incorporate incremental learning algorithms, enabling models to continuously learn from emerging attack data and update themselves without full retraining. This approach facilitates long-term, effective defense against dynamic network threat environments.

Furthermore, novel datasets like UWF-ZeekData24, featuring updated attack vectors and rich Zeek log context, provide an ideal testbed for evaluating model generalization in real-world network environments. We plan to immediately validate the HMK-ST model's performance on this dataset and further refine its feature selection and visualization strategies to ensure resilience against evolving cyber threats.

Finally, we will explore innovatively applying the core technological innovations of this research to a privacy-preserving federated learning framework. Specifically, we plan to deploy the HMK feature selection and dimensionality reduction modules across client nodes in a federated learning setup. This enables each client to extract standardized, low-dimensional, and highly discriminative features locally without sharing data, before uploading them to the server for aggregation and global model training. This approach holds promise for rigorously safeguarding data privacy while pooling collective efforts to enhance the robustness and accuracy of intrusion detection models. It also effectively mitigates potential issues such as poisoning attacks in federated learning environments.

Author Contributions: Conceptualization, H.W. and W.J.; methodology, H.W.; software, H.W.; validation, H.W. and W.J.; formal analysis, H.W.; investigation, H.W., Z.X., and B.Z.; resources, H.W.; data curation, H.W.; writing—original draft preparation, H.W.; writing review and editing, W.J. and J.Y.; visualization, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61401269.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset UNSW-NB15 used for this research is available at https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 4 September 2025).The dataset CICIDS 2017 used for this research is available at https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 4 September 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
IDS	intrusion detection system
HMK	hybrid multi-model feature selection and kernel-based dimension reduction

References

1. Xie, W. A Brief Introduction to Network Information Security Hazards and Application of Security Technology. *Netw. Secur. Technol. Appl.* **2021**, 66–67.
2. James P. Anderson Co. *Computer Security Threat Monitoring and Surveillance*; James P. Anderson Co.: Fort Washington, PA, USA, 1980.
3. Kshirsagar, D.; Shaikh, J.M. Intrusion Detection Using Rule-Based Machine Learning Algorithms. In Proceedings of the 2019 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 19–21 September 2019; pp. 1–4. [CrossRef]
4. Weon, I.Y.; Song, D.H.; Lee, C.H. Effective Intrusion Detection Model through the Combination of a Signature-based Intrusion Detection System and a Machine Learning-based Intrusion Detection System. *J. Inf. Sci. Eng.* **2006**, *22*, 1447–1464.
5. Zhang, H.; Zhang, X.; Zhang, Z.; Li, W. Summary of Intrusion Detection Models Based on Deep Learning. *Comput. Eng. Appl.* **2022**, *58*, 17–28.
6. Wang, Z.G.; Oates, T. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence 2015, Austin, Texas, USA, 25–30 January 2015.
7. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 104650–104675. [CrossRef]
8. Mazid, A.; Kirmani, S.; Abid, M. Enhanced intrusion detection framework for securing IoT network using principal component analysis and CNN. *Inf. Secur. J. Glob. Perspect.* **2024**, 1–21. [CrossRef]
9. Geng, Z.; Li, X.; Ma, B.; Han Y.M. Improved convolution neural network integrating attention based deep sparse auto encoder for network intrusion detection. *Appl. Intell.* **2025**, *55*, 141. [CrossRef]
10. Lu, H.T.; Dong, Y.N.; Quan, Y.X. A Method for Continuous Detection and Classification of Malicious Network Traffic Based on Double-Layer Model and Distribution of Indexes. *Acta Electron. Sin.* **2025**, *53*, 1–13. [CrossRef]
11. Tikhe, G.N.; Patheja, P.S. Feature attention assisted convolutional stacked sparse auto-encoder model for intrusion detection in network function virtualization environment. *Comput. Secur.* **2025**, *157*, 104595. [CrossRef]
12. Ahmad, J.; Latif, S.; Khan, I.U.; Alshehri, M.S.; Khan, M.S.; Alasbali, N.; Jiang, W. An interpretable deep learning framework for intrusion detection in industrial Internet of Things. *Internet Things* **2025**, *33*, 101681. [CrossRef]
13. Hu, Y.; Feng, Y.; Zhao, Y.; Mao, X. IoT-ONDDQN: A detection model based on deep reinforcement learning for IoT data security. *Comput. Commun.* **2025**, *241*, 108263. [CrossRef]
14. Bo, J.; Chen, K.; Li, S.; Gao, P. Boosting Few-Shot Network Intrusion Detection with Adaptive Feature Fusion Mechanism. *Electronics* **2024**, *13*, 4560. [CrossRef]
15. Yang, Y.; Liu, Z.; Wong, K.C.; Lin, Q.; Luo, J.; Li, J. Evolutionary multi-task robust architecture search for network intrusion detection. *Expert Syst. Appl.* **2026**, *296*, 128899. [CrossRef]
16. Prabu, M.; Sasikala, L.; Suresh, S.; Ramya, R. A novel intrusion detection system: Integrating greedy sand cat swarm optimization and dual attention graph convolutional networks. *Int. J. Syst. Assur. Eng. Manag.* **2025**, 1–21. [CrossRef]
17. Francis, G.T.; Souri, A.; İnanç, N. A hybrid firefly-based attribute selection and split-point mechanism for securing software-defined industrial Internet of Things. *J. High Speed Netw.* **2025**, *31*, 228–243. [CrossRef]
18. Sun, Z.; Teixeira, A.M.; Toor, S. GNN-IDS: Graph Neural Network based Intrusion Detection System. In Proceedings of the ARES '24: 19th International Conference on Availability, Reliability and Security, Vienna, Austria, 30 July–2 August 2024. [CrossRef]

19. Lo, W.W.; Layeghy, S.; Sarhan, M.; Gallagher, M.; Portmann, M. E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. In Proceedings of the NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; pp. 1–9. [CrossRef]
20. Tran, D.H.; Park, M. Graph Embedding for Graph Neural Network in Intrusion Detection System. In Proceedings of the 2024 International Conference on Information Networking (ICOIN), Ho Chi Minh City, Vietnam, 17–19 January 2024; pp. 395–397. [CrossRef]
21. Tayebi, M.; Kafhali, S.E. Performance analysis of recurrent neural networks for intrusion detection systems in Industrial-Internet of Things. *Frankl. Open* **2025**, *12*, 100310. [CrossRef]
22. Guo, D.; Xie, Y. Research on Network Intrusion Detection Model Based on Hybrid Sampling and Deep Learning. *Sensors* **2025**, *25*, 1578. [CrossRef]
23. Li, C.C.; Yuan, Z.L.; Tang, G.F. Research on Deep Learning-based Spatio-temporal Feature Fusion Network Intrusion Detection Model. *J. Inf. Secur. Res.* **2025**, *11*, 122–129.
24. Abbassy, M.M.; Kumar, A.S.S.; Ead, W.M.; Aboalndr, A.A.M.; Alsheref, F.K.; Abdalla, M.; Shivani, A. Hybrid deep learning framework for detection of anomalies in cyber-physical systems. *Int. J. Inf. Technol.* **2025**, 1–7. [CrossRef]
25. Kalaivani, A.; Pugazendi, R. CNN-RBM Integrated Deep Learning Design for Categorizing Attack in an Intrusion Detection System. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2025**, *33*, 613–622. [CrossRef]
26. Alabbadi, A.; Bajaber, F. X-FuseRLSTM: A Cross-Domain Explainable Intrusion Detection Framework in IoT Using the Attention-Guided Dual-Path Feature Fusion and Residual LSTM. *Sensors* **2025**, *25*, 3693. [CrossRef]
27. Liu, L.; Zhao, H.; Li, X.; Sun, B. Multiscale residual temporal convolutional networks-based intrusion detection model in Internet of things. *Telecommun. Sci.* **2025**, *41*, 164–175. [CrossRef]
28. Song, Y.; Luktarhan, N.; Shi, Z.; Wu, H. TGA: A Novel Network Intrusion Detection Method Based on TCN, BiGRU and Attention Mechanism. *Electronics* **2023**, *12*, 2849. [CrossRef]
29. Ling, J.; Zhang, L.; Liu, C.; Xia, G.; Zhang, Z. Machine Learning-Based Multilevel Intrusion Detection Approach. *Electronics* **2025**, *14*, 323. [CrossRef]
30. Mao, J.; Yang, X.; Hu, B.; Lu, Y.; Yin, G. Intrusion Detection System Based on Multi-Level Feature Extraction and Inductive Network. *Electronics* **2025**, *14*, 189. [CrossRef]
31. Peng, Q.; Wang, X.; Tang, A.; Wang, J.; Hua, Y.; He, F. Intrusion detection method based on DBN-SVM optimized by grey wolf algorithm. *J. Nanjing Univ. Sci.* **2025**, *61*, 270–282. [CrossRef]
32. Bouzaachane, K.; Guarmah, E.M.E.; Alnajim, A.M.; Khan, S. Addressing Modern Cybersecurity Challenges: A Hybrid Machine Learning and Deep Learning Approach for Network Intrusion Detection. *Comput. Mater. Contin.* **2025**, *84*, 2391–2410. [CrossRef]
33. Govindarajan, V.; Muzamal, J.H. Advanced cloud intrusion detection framework using graph based features transformers and contrastive learning. *Sci. Rep.* **2025**, *15*, 20511. [CrossRef] [PubMed]
34. Barve, A.; Malviya, A.; Ranjan, V.; Jeet, R.; Bhosle, N. Enhancing detection rates in intrusion detection systems using fuzzy integration and computational intelligence. *Comput. Secur.* **2025**, *157*, 104577. [CrossRef]
35. Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, A.; Alyami, S.A.; Liò, P.; Kabir, M.A.; Moni, M.A. SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization. *Electronics* **2023**, *12*, 3541. [CrossRef]
36. Du, C.; Guo, Y.; Zhang, Y. A Deep Learning-Based Intrusion Detection Model Integrating Convolutional Neural Network and Vision Transformer for Network Traffic Attack in the Internet of Things. *Electronics* **2024**, *13*, 2685. [CrossRef]
37. Sun, J.; Yang, S. Adversarial Sample Generation Based on Model Simulation Analysis in Intrusion Detection Systems. *Electronics* **2025**, *14*, 870. [CrossRef]
38. Du, X.; He, M.; Yu, X. A Feature Fusion-based Controller Area Network Intrusion Detection Method. *J. Beijing Univ. Posts Telecommun.* **2025**, *48*, 112–118. [CrossRef]
39. Luo, J.; Zhang, Y.; Wu, Y.; Xu, Y.; Guo, X.; Shang, B. A Multi-Channel Contrastive Learning Network Based Intrusion Detection Method. *Electronics* **2023**, *12*, 949. [CrossRef]
40. Kumar, N.; Sharma, S. A Hybrid Modified Deep Learning Architecture for Intrusion Detection System with Optimal Feature Selection. *Electronics* **2023**, *12*, 4050. [CrossRef]
41. Nowroozi, E.; Haider, I.; Taheri, R.; Conti, M. Federated Learning Under Attack: Exposing Vulnerabilities Through Data Poisoning Attacks in Computer Networks. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 822–831. [CrossRef]
42. Nabavirazavi, S.; Taheri, R.; Shojafar, M.; Iyengar, S.S. Impact of Aggregation Function Randomization against Model Poisoning in Federated Learning. In Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK, 1–3 November 2023; pp. 165–172. [CrossRef]
43. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [CrossRef]
44. Pan, Y. Bio-Syncretic Rehabilitation Mechanism Theory and Application. Master's Thesis, Yanshan University, Qinhuangdao, China, 2021. [CrossRef]

45. Ma, T.; Wu, C.; Qiao, W.; Wang, Y.; Sha, Y. Application of Kernel Function in Irregular Face Recognition. *Comput. Digit. Eng.* **2019**, *47*, 1338–1341. [CrossRef]
46. Wang, H.; Zhang, H. Moving Object Detection Method Based on Low-Rank and Sparse Decomposition in Dynamic Background. *J. Electron. Inf. Technol.* **2020**, *42*, 2788–2795. [CrossRef]
47. Das, S.; Saha, S.; Priyoti, A.T.; Roy, E.K.; Sheldon, F.T.; Haque, A.; Shiva, S. Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4821–4833. [CrossRef]
48. Cui, J.; Zong, L.; Xie, J.; Tans, M. A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data. *Appl. Intell.* **2023**, *53*, 272–288. [CrossRef]
49. Yu, X.; Lu, Y.; Jiang, F.; Hu, Q.; Du, J.; Gong, D. A Cross-Domain Intrusion Detection Method Based on Nonlinear Augmented Explicit Features. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 187–197. [CrossRef]
50. Attique, D.; Hao, W.; Ping, W.; Javeed, D.; Kumar, P. Explainable and Data-Efficient Deep Learning for Enhanced Attack Detection in IIoT Ecosystem. *IEEE Internet Things J.* **2024**, *11*, 38976–38986. [CrossRef]
51. Djama, A.; Maazouz, M.; Kheddar, H. Hybrid Machine Learning Approaches for Classification DDoS Attack. In Proceedings of the 2024 1st International Conference on Electrical, Computer, Telecommunication and Energy Technologies (ECTE-Tech), Oum El Bouaghi, Algeria, 17–18 December 2024; pp. 1–7. [CrossRef]