Hybrid Intrusion Detection System Based on Data Resampling and Deep Learning

Huan Chen[1], Gui-Rong You[2], Yeou-Ren Shiue[3]
College of Information Engineering, Fujian Business University, Fuzhou, China[1, 2]
Fujian Provincial Universities Engineering Research Center of Big Data Analytics for Business Intelligence,
Fujian Business University, Fuzhou, China[2]
Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan[3]

***Abstract*—The growth of the internet has advanced information-sharing capabilities and vastly increased the importance of global network security. However, because new and inconspicuous abnormal behaviors are nearly impossible to detect in massive network access environments, modern intrusion detection systems have identified a high rate of false-positive (FP) and false-negative (FN) attacks. To overcome this, this paper proposes a hybrid deep learning model that significantly mitigates the disadvantages of consistently imbalanced sample attack data. First, it resolves imbalanced data using random undersampling and synthetic minority oversampling techniques. Then, convolutional neural networks (CNNs) extract local and spatial features, and a transformer encoder extracts global and temporal features. The novelty of this combination increases recognition accuracy at the algorithm level, which is crucial to reducing FPs and FNs. The model was subjected to multiclassification testing on the NSL-KDD and CICIDS2017 benchmark datasets, and the results show that our model has higher classification accuracy and lower FP rates than state-of-the-art intrusion detection models. Moreover, it significantly improves the detection rate of low-frequency attacks.**

Keywords—Intrusion detection; deep learning; random undersampling; synthetic minority oversampling technique; convolutional neural network; transformer

I. Introduction

The ubiquity of mobile handheld internet devices allows people to access digital information quickly and effortlessly, just about anywhere. The associated transference and storage of vast volumes of data over computer networks have created new and evolving opportunities for cybercriminals [1]. From 2019 to 2022, the cost of repairing cyberattack damage increased by USD 6T, and the average detection time increased from 57.4 to 93.2 days [2]. Traditional cybersecurity methods (e.g., firewalls, user authentication, and data encryption) cannot handle the complex attacks that take place online. Intrusion detection systems (IDSs) are designed to detect a variety of anomalous patterns that serve as the attack signatures of new and known attacks [3], using advanced database systems with machine learning [4]. When an IDS reports potential malicious activities in an information system [5], it kicks off various analytical and alerting processes to confirm the nature of the attack and launch protection measures.

IDSs generally operate in three phases: information collection, data analysis, and response. The fact that most advanced cyberattacks utilize new and unusual network and system penetration methods makes it nearly impossible to train machine learning models to recognize discrete new and seemingly inconspicuous threats. On the other hand, the models must still be trained with legacy class types from past attacks. Hence, IDS training datasets grow heavily imbalanced over time [6]. Supposing the target class is rare (< 10%) in terms of its representation in the training dataset, critical new and unusual network behaviors can be easily overlooked (as with human perception).

Modern machine-learning methods that handle unbalanced classes typically consider both data- and algorithm-level remedies. That is, training data and classification algorithms are modified separately so that their combination will improve the detection and recognition accuracy of minority samples. Hence, advantageous tradeoffs can be gained. Unfortunately, even the most state-of-the-art IDS models continue to suffer high rates of false-positive (FP) and false-negative (FN) attack detection.

To contribute to the robustness of machine-learning IDS accuracy and recognition, this study makes the following contributions. (1) We apply a novel combination of data- and algorithm-level techniques to specifically reduce the FP rate while improving the model's recall rate. (2) We provide legitimate and reproducible results by applying our combined model to state-of-the-art NSL-KDD [7] and CICIDS2017 [8] benchmark intrusion detection datasets as our research objects. (3) To improve data-level class balancing, we provide an ingenious combination of random undersampling (RUS) and synthetic minority oversampling to adjust the data distribution structure and improve minority class detection. (4) To improve algorithm-level class balancing, we apply a hybrid convolutional neural network (CNN) and a Transformer model to adopt new detection performance efficiencies over contemporary models.

Our model's performance is compared with that of state-of-the-art IDS models, demonstrating that our innovations have clear advantages in terms of accuracy, FP rate, and recall.

To convincingly deliver this information, the remainder of this paper proceeds as follows. Section II covers the extant research that has led us to pursue our current motivations. Section III adequately describes the proposed model and the related techniques and technologies applied. Section IV describes our experiments and presents the comparison results,

Fund Project: Natural Science Foundation of Fujian Province, China [grant no. 2021J01332]

configurational impacts, and implications of our findings. Finally, Section V presents the conclusions.

II. Related Works

This study provides an IDS model that can more accurately identify malicious traffic and detect a wider variety of intrusion attacks than current models. First, our model resolves sample imbalance problems at the data and algorithm levels based on the lessons learned from current studies, described briefly in the following sections.

A. Data-Level Mitigation Efforts

In terms of the current data-level mitigation efforts used to overcome problems related to training models with imbalanced datasets, data reconstruction efforts prevail. Related strategies focus on preprocessing original datasets to provide appropriately weighted training sets for model learning and tailoring the model's feature classification methods to maximize learning and retention based on the task at hand.

A healthy number of intrepid researchers have applied oversampling [9-14], undersampling [15-19], and hybrid [20-23] preprocessing methods to restore balance to their training datasets. These methods are combined with feature classification methods to maximize benefits. For example, the synthetic minority oversampling technique (SMOTE) [24] is a widely used data reconstruction strategy that provides good data balancing and classification results while effectively avoiding overfitting. Noting that the SMOTE algorithm analyzes minority class samples and manually synthesizes new ones based on the needed additions, Dablain et al. [25] provided a deep learning-based SMOTE method that applies a novel oversampling method to counter class imbalances and train new skew-insensitive classifiers. Joloudari et al. [26] proposed a CNN that uses SMOTE to achieve a remarkable accuracy of 99.08% on 24 imbalanced datasets, including KEEL, Breast Cancer, and Z-Alizadeh Sani sets.

B. Algorithm-Level Mitigation Efforts

Most current algorithm-level mitigation efforts aim to intuitively process input data algorithmically for better classification results. Modern techniques match the model's internal structure to the distribution characteristics of the original dataset as much as possible. For example, CNN-based autoencoders are extensively used for IDSs, resulting in high detection performance [27]. Yin et al. [28] proposed a recurrent neural network (RNN)-based IDS that provides impressive breakthroughs in accuracy. Vigneswaran et al. [29] used a deep neural network (DNN) to predict attacks directed at network IDSs (NIDS). The famous KDD-CUP99 [30] dataset was used to train and benchmark, revealing that a DNN with three layers outperformed all other classical machine learning algorithms at the time. XIAO et al. [31] proposed IDS to reduce the required CNN features for computational efficiency. The KDD-CUP99 dataset was again used, showing reduced FPs and improved speeds. Belarbi et al. [32] proposed a multi-class NIDS based on a deep belief network (DBN) using the CICIDS2017 dataset to train and evaluate performance. The experimental results demonstrated that DBNs can surpass traditional multilayer perceptron classification performance, significantly improving overall recall. In 2017, Vaswani et al. [33] proposed the transformer model, originally designed to solve the tasks of language modeling and machine translation, achieving good results; this model has also been gradually applied to network IDSs. Wang et al. [34] proposed a robust unsupervised IDS (RUIDS) by introducing a masked context reconstruction module into a transformer-based, self-supervised learning scheme. Extensive experiments on four intrusion datasets were conducted to demonstrate the effectiveness and robustness of the RUIDS. Yang et al. [5] proposed IDS based on an improved vision transformer, demonstrating superior results on the NSL-KDD public intrusion detection via simulation experiments.

C. Hybrid Solutions

As noted, CNNs, RNNs, (Recurrent Neural Networks) and DBNs (Deep Belief Networks) are among the most common IDS solutions used to mitigate imbalanced data problems [36]. Hybrid models have recently become popular, based on their observed improvements to symbiotic and amplified model strength [37]. Indeed, research has shown that combined models consistently perform better than individual algorithms [38]. Table I list the best representative hybrid IDS models and summarize their basic algorithmic models, dataset properties, classification types, and accuracy results. This listing is fully explained in the subsequent narrative.

1) Focused neural network combinations: Zhang et al. [39] proposed an IDS model based on an improved genetic algorithm with a DBN trained and evaluated using the NSL-KDD dataset, demonstrating effective improvements in intrusion recognition rates (> 99%). Wu et al. [40] proposed a hierarchical CNN + RNN model (i.e., LuNet) that effectively extracts spatial and temporal data features, providing higher detection accuracy and fewer FPs than peer methods. LuNet's verification accuracies on the NSL-KDD and UNSW-NB15 datasets were 99.24% and 97.40%, respectively. Souza et al. [41] proposed a hybrid binary classification model comprising a DNN with a k-nearest-neighbors (kNN) function. This method achieved higher accuracy than classical machine learning methods, with 99.77% on the NSL-KDD dataset and 99.85% on the CICIDS2017 dataset. Albahar et al. [42] proposed an approach that combines a regularization algorithm with an artificial neural network, achieving all-time-high true-positive (TP) and accuracy rates on the NSL-KDD, UNSW-NB15, and CIDDS-001 datasets (i.e., 98.53, 94.58, and 97.87%, respectively) using 10-fold cross-validation. Ahsan et al. [43] proposed a hybrid CNN with a long short-term memory (LSTM) network, achieving the highest known accuracy (at the time) of 99.70% on the NSL-KDD dataset. Banaamah et al. [44] adopted a CNN with an LSTM and a gated recursive unit (GRU) model to improve internet-of-things (IoT) security. Using the highly reputable Bot-IoT dataset, the proposed model surpassed the highest accuracy, with a 99.8% ratio. Kamalakkannan et al. [45] developed an improved CNN + LSTM model that learns spatial and temporal data characteristics, demonstrating 98% accuracy and a 98.14% average detection rate on the NSL-KDD dataset.

Shivhare et al. [46] proposed a CNN + LSTM + SVM model to tackle multiclass tasks on the CICIDS 2017 dataset, achieving an accuracy of 97.29%. Qazi et al. [47] proposed a deep-layered CNN + RNN model to detect and classify malicious traffic using the CICIDS-2018 dataset, achieving an average accuracy of 98.90%. Recently, the use of transformers has provided new feature extraction methods. Transformers are deep neural networks wholly based on attention mechanisms that have shown great success in natural language processing (NLP) fields. Their versatility allows them to be applied to other domains, such as image classification, cybersecurity, and more. Xing et al. [48] sought to improve unknown attack learning and detection by extracting data features from different perspectives using CNN and transformer models. Xiang et al. [49] later proposed a transformer-based fusion deep learning architecture in which the transformer is used to adjust the ML-CNN-BiLSTM model to enhance its feature encoding ability. Ullah et al. [50] proposed an IDS using transformer-based transfer learning for imbalanced network traffic (INT). The resulting DS-INT uses transformer-based transfer learning to learn feature interactions in network feature representations, even with imbalanced data. A hybrid CNN-LSTM model was then developed to detect attacks from deep features.

TABLE I. SUMMARY OF THE HYBRID INTRUSION DETECTION SYSTEM

Ref.	Year	Authors	Classification Algorithms	Dataset	Classes	Accuracy (%)
[39]	2019	Zhang et al.	DBN, Impr. Genetic	NSL-KDD	Multiclass	>99.00
[40]	2019	Wu et al.	CNN, RNN	NSL-KDD	Binary, Multiclass	99.24 (Bin.) 99.05 (Multi.)
				UNSW-NB15		97.40 (Bin.) 84.98 (Multi.)
[41]	2020	Souza et al.	DNN, KNN	NSL-KDD	Binary	99.77
				CICIDS-2017		99.85
[42]	2020	Albahar et al.	ANN, Regularization	NSL-KDD	Multiclass	98.53
				UNSWNB15		94.58
				CIDDS-001		97.87
[43]	2020	Ahsan et al.	CNN, LSTM	NSL-KDD	Multiclass	99.70
[44]	2022	Banaamah et al.	CNN, LSTM, GRUs	Bot-IoT	Binary	99.80
[45]	2023	Kamalakkannan et al.	2D LSTM, CNN	NSL-KDD	Multiclass	98.00
[46]	2023	Shivhare et al.	CNN, LSTM, SVM	CICIDS-2017	Binary	97.29
[47]	2023	Qazi et al.	CNN, RNN	CICIDS-2018	Binary	98.90
[48]	2023	Xing et al.	CNN, Transformer	UNSW-NB15	Multiclass	88.47
[49]	2023	Xiang et al.	ML-CNN, BiLSTM, Transformer	UNSW-NB15	Binary	90.3
[50]	2023	Ullah et al.	Transformer, CNN, LSTM	UNSW-NB15, CICIDS-2017, NSL-KDD	Multiclass	99.21

2) Focused Data- and Algorithm-Level combination: Yan et al. [51] proposed a novel combinatorial IDS model based on a deep RNN and a region-adaptive SMOTE technique. This model significantly improved the detection rate of low-frequency attacks and overall efficiency while improving unknown attack detection. Al et al. [52] proposed a hybrid CNN + LSTM + SMOTE and the Tomek–Link sampling method (i.e., STL) to improve system performance to an impressive extent. Cao et al. [36] designed a CNN + GRU model that extracts spatiotemporal features from network data traffic. This model combines adaptive synthetic sampling (ADASYN) and repeatedly edits its nearest neighbors to process positive and negative sample imbalances in the original dataset. This model resolves both low classification accuracy and imbalance problems.

D. Motivation for and Purpose of this Study

Through the research and discussion of the above literature, we can see that model systems combining two or more algorithms can often obtain better detection capabilities than single algorithms. Of course, with that comes an increase in the cost of computation. Therefore, how to achieve better detection results at the exact computational cost, the reasonable choice of classification algorithm will be the key to the problem.

The CNN model has become one of the classification algorithms selected in this paper because it can comprehensively map the data features, mine the relationship between the features, and improve the accuracy of feature extraction. However, the CNN model focuses more on spatial local features and has time series characteristics for the traffic data studied in this paper. Therefore, the processing ability of sequence data will be emphasized in selecting the second classification algorithm. RNN, GRU, LSTM, and Transformer are all sequential models in deep learning. Compared with RNN and LSTM, the Transformer model can obtain the relationship between all the information in the sequence through the self-attention mechanism, which can better cope with the long-term dependency problem and has higher accuracy. The model can be operated in parallel, and the calculation speed is faster. Based on the above reasons, the CNN and the Transformer models have become the algorithm choices for this paper's hybrid intrusion detection system.

In addition, previous studies have primarily focused on the overall detection rate of the system, but for the typical unbalanced network traffic data, identifying a small number of attack samples is the key to detection classification. Therefore, the difference between this paper and previous studies is that the system focuses more on the identification rate of minority species without significantly affecting the overall detection rate. To achieve this goal, the system balances the sample size of the majority class and the minority class at the data level through data resampling technology to adapt to the common classifier that pursues global accuracy.

III. Proposed Model

The model proposed in this study uses the NSL-KDD and CICIDS2017 datasets as the research targets. New training, validation, and testing sets were divided by random sampling to digitize and normalize the original data. Most class samples were randomly undersampled to stress the sample imbalance problem.

The focus of this model is on the classification research of imbalanced data, which are divided into two levels for operation. First, at the data level, a data reconstruction strategy is used to adjust the internal distribution structure of the data so that the imbalanced dataset tends toward a balanced state. The measure is obtained by randomly undersampling the majority class samples in the training set and oversampling the minority class samples with SMOTE to achieve balanced data.

Second, at the algorithmic level, the model adjusts the traditional classification algorithm or proposes the optimization and improvement of existing classification ideas as an adaption technique to handle the inherent characteristics of imbalanced datasets, thereby improving the overall recognizability of the model. Research has shown that combined models consistently perform better than individual algorithms [38]. As mentioned, we combined the classic CNN with a transformer self-attention module to achieve optimization by combining multiple classifiers that adapt to the internal distributed structure of imbalanced datasets. Hence, the detection rate of the model will be improved.

This model accounts for both data- and algorithm-level aspects of the problem and utilizes their combined advantages to achieve superior recognition accuracy with minority class samples. Fig. 1 presents a schematic diagram of our proposed model.

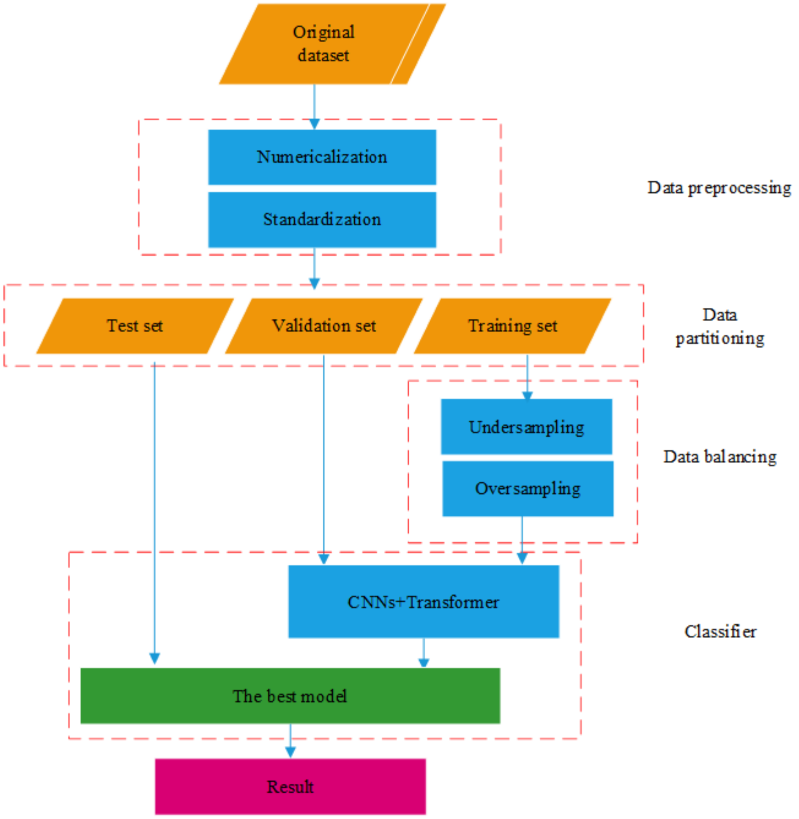

Fig. 1. Schematic diagram of the proposed model.

A. Dataset Description

1) NSL-KDD Dataset: According to [5], NSL-KDD [53] and KDD-CUP99 [54] are the most widely used datasets in IDS research (ca. 2012–2022). The NSL-KDD dataset was generated in 2009 and is commonly used to train models for anomaly detection. It is a revised version of the classic KDD99 dataset but retains its structure. The new dataset consists of four subsets: KDDTest+, KDDTrain+, KDDTest-21, and KDDTrain+_20%, where the latter two are subsets of the first two, respectively.

In the NSL-KDD dataset, each sample record contains 41 attribute features and a classification identifier. Normal and abnormal network connections are marked with the classification identifier. The normal type is represented as "normal," and the dataset contains many anomalies and 39 attack identifiers. These identifiers are divided into four categories by type: denial of service (DoS), probe, root-to-local (R2L), and unauthorized-to-root (U2R).

Our experiment uses the original data sources of KDDTrain+ (125,973 sample records) and KDDTest+ (22,544 sample records). Table II presents the sample size distributions of each attack type.

2) CICIDS2017 Dataset: Table II shows that the NSL-KDD dataset is a typical imbalanced dataset. Notice the small proportion of Probe, R2L, and U2R attack-type samples, especially for U2R attacks. Although this dataset is very popular in IDS studies, some researchers have pointed out that it is somewhat outdated.

Emerging datasets include UNSW-NB15, CICIDS2017, Bot-IoT, and others. Among them, CICIDS2017 is the most popular. Therefore, we chose CICIDS2017 as our second benchmark to gauge performance differences.

The CICIDS2017 dataset was released in 2017 [55], providing normal data and the latest common attack types, similar to real-world data. It contains 2,830,743 network traffic samples, each containing 83 network traffic features. It also includes one benign and 14 attack categories, including the standard DoS, botnet, web, infiltration, file transfer protocol patator, and SSH patator types [56]. Among the 14 attack categories, tags with similar features and behaviors are merged to form five new categories. The distribution of the number of samples in the CICIDS2017 dataset is shown in Table III. The CICIDS2017 dataset is also imbalanced, with bot-and-web attack class samples being particularly scarce.

TABLE II. DISTRIBUTION OF VARIOUS SAMPLES FROM THE NSL-KDD DATASET

Dataset	The number and proportion of various types of samples					
	Total	***Normal***	***DoS***	***Probe***	***R2L***	***U2R***
KDD Train+	125973	67343 (53.46%)	45927 (36.46%)	11656 (9.25 %)	995 (0.79 %)	52 (0.04%)
KDD Test+	22544	9711 (43.08%)	7458 (33.08%)	2421 (10.74%)	2754 (12.22 %)	200 (0.89%)

TABLE III. DISTRIBUTION OF VARIOUS SAMPLES IN THE CICIDS2017 DATASET

Dataset	Number and proportion of various types of samples					
	BENIGN	***Bot***	***BruteForce***	***DoS /DDoS***	***Port Scan***	***Web Attack***
CICIDS2017	2035505 (83.91 %)	1943 (0.08 %)	8551 (0.35%)	320269 (13.20 %)	57341 (2.36 %)	2118 (0.09 %)

B. Data Preprocessing

Using the NSL-KDD dataset as an example, data preprocessing was introduced, and the operation of the CICIDS2017 dataset was similarly manipulated.

1) Numericalization: The NSL-KDD dataset contains 41 attribute features (i.e., 38 digital and three non-digital types). Because the input value of the model should be a digital matrix, it was necessary to use a numerical method to map data with symbolic features into digital feature vectors. We used the LabelEncoder method of the preprocessing module in the sklearn library to convert the three non-digital features (i.e., protocol_type, service, and flag) into digital features.

2) Standardization: Unlike normalization, which is easily affected by outliers, standardization is relatively stable; thus, it is suitable for noisy big data scenarios. Therefore, standardization was used for data preprocessing. The original data were transformed into a range with a mean of zero and a standard deviation of one so that the processed data would conform to a standard normal distribution. The StandardScaler method of the preprocessing module in the sklearn library uses a standard z-score scaling calculation formula, expressed using Eq. (1):

$$X' = \frac{\chi - \ mean}{\sigma}\,, \tag{1}$$

where, X' represents the converted data value, χ is the original data value, mean is the mean value of the column data, and σ is the standard deviation of the column data.

C. Dataset Partitioning

The KDDTrain+ and KDDTest+ subsets of the NSL-KDD dataset were used as the original data, and new training, validation, and testing sets were formed by random sampling. It lists the number and proportions of each sample set after division. The CICIDS2017 dataset was also divided according to the same ratio, and the numbers after the division are listed in Table IV. To achieve good data balance, undersampling and oversampling were performed on the training set samples.

TABLE IV. NUMBER AND PROPORTION OF DATASETS AFTER PARTITIONING

Dataset	#training set	#validation set	#testing set
NSL-KDD	103,961	14,852	29,704
CICIDS2017	1,698,008	242,573	485,146
Proportion	70%	10%	20%

D. Data Balancing

1) Undersampling: The undersampling method achieves data equalization by randomly removing a certain proportion of majority instances from the RUS dataset [23]. This process consists of the following steps:

a) The numbers of majority samples, N1, and minority samples, N2, are calculated.

b) Based on the set sampling ratio, r, we calculate the number of majority class samples needing deletion (N1 - N2 * r).

c) Randomly selected samples from the majority class, S_{maj}, to form the sample set E; remove sample set E from S_{maj}; generate a new dataset $S_{new-maj} = S_{maj} - E$.

In the NSL-KDD dataset, NORMAL and DoS samples belong to the majority class, and undersampling was performed using RUS samples. The BENIGN and DoS/DDoS samples of the CICIDS2017 dataset belong to the majority class and are undersampled.

2) Oversampling: Oversampling is used to rebalance a dataset by creating fake minority instances, and SMOTE [22] is the best method [57] in our case as it effectively compensates for the shortcomings of random oversampling and is superior to simple replication, which can easily cause model overfitting and weaken generalizability. SMOTE also has the advantages of a simple design and strong robustness. Moreover, it uses interpolation between minority class samples and their nearest neighbors to generate new synthetic samples [58]. The SMOTE steps are as follows:

a) For each sample X in the minority class, a k-NN is used to sample each minority class sample.

b) We determine the sampling rate, N, based on the sample imbalance ratio and randomly select N samples from K nearest neighbors for random linear interpolations.

c) We construct a new minority class sample using Eq. (2):

$$\mathrm{New} = \mathrm{x_i} + rand(0,1) \times \left(\mathrm{y_j} - \mathrm{x_i}\right), \mathrm{j} = 1, 2, \ldots \mathrm{N} \quad , \tag{2}$$

where, $\mathrm{x_i}$ is an observation point in the minority class, $\mathrm{y_j}$ is a randomly selected K -nearest neighbor, and $rand(0,1)$ represents a random number generated between zero and one.

d) New samples are combined with the original data to form a new dataset.

In the NSL-KDD dataset, Probe, R2L, and U2R samples belong to a minority class and were oversampled with SMOTE to increase the number of class samples. For the CICIDS2017 dataset, the Bot, Brute Force, PortScan, and Web Attack samples belong to the minority class and were oversampled. The training set samples were balanced at the data level via undersampling and oversampling.

E. Model Structure

1) CNN: CNNs are feedforward neural networks with convolution calculations and a deep structure that extract features accurately and efficiently [59]. The error function is obtained by calculating the difference between the actual and predicted values. Network parameters are adjusted retroactively until the model reaches an optimal solution [60]. This method has been widely used in several fields, such as NLP and computer vision.

A CNN generally comprises a convolution layer, activation function, pooling layer, and a fully connected layer [61] as shown in Fig. 2. The convolution layer extracts high-level features from the input data, and the pooling layer performs feature selection and information filtering on the graph data output by the convolution layer, thereby reducing the amount of data processing.

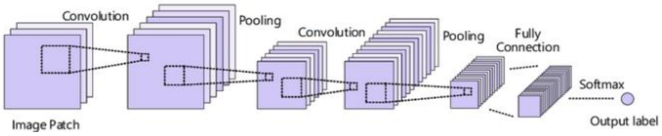

Fig. 2. CNN structure.

2) Transformer: A transformer is a deep learning model [33] that is widely used for NLP and other sequential data processing tasks.

The transformer differs from traditional RNNs and CNNs in that they adopt a novel self-attention mechanism that allows the model to assign different weights to different elements when processing input sequences. It calculates the similarity score between elements and uses the score to calculate the weighted averages of relationships among elements. Notably, the transformer supports parallel computing, this allows it to handle long sequences easily without step-by-step iterations. The self-attention mechanism also allows the transformer to incorporate information from the entire sequence into its calculations, which leads to better long-range dependencies.

CNNs are particularly adept at modeling fine-grained local features due to their convolutional operations and hierarchical structure. Nevertheless, their global modeling ability is weak, whereas the transformer excels at modeling global contextual information [62]. The proposed framework utilizes complementary CNN characteristics to extract local, spatial, and time series features.

3) Hybrid model: This article adopts a hybrid architecture that combines the CNN and the transformer as illustrated in Fig. 3. Spatial features are extracted after preprocessing and sample balancing in one-dimensional (1D) convolutional and pooling layers. Then, by using the self-attention mechanism of the transformer to process the data, the shortcomings of the RNN's short-term memory and the CNN's difficulties in learning remote dependencies are overcome, and temporal and global features are extracted. Finally, using flattening and fully connected functions, the data are classified according to attack type. For the NSL-KDD dataset, the data were divided into five categories: one normal and four attack. The CICIDS2017 dataset was divided into six categories: one benign and five attacks.

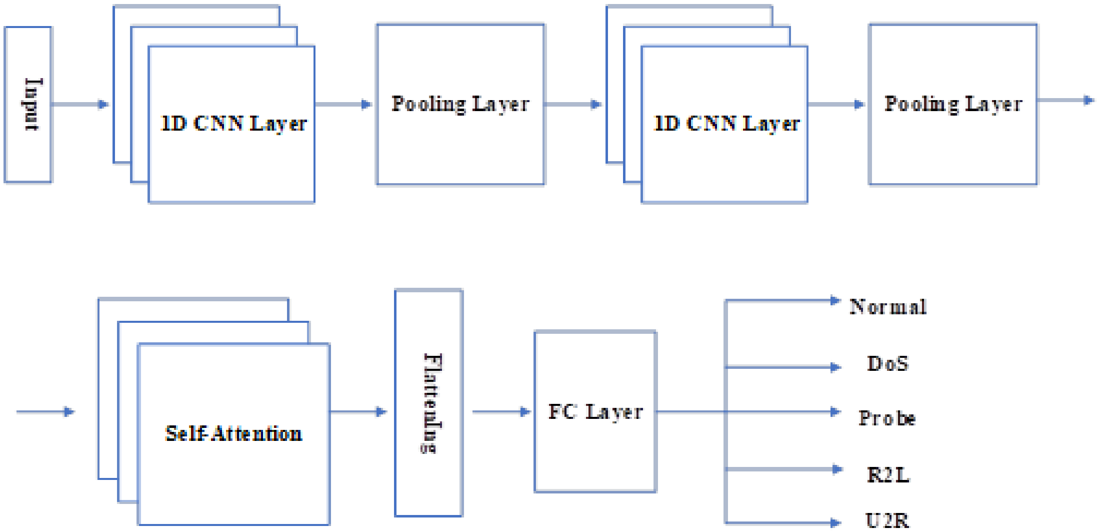

Fig. 3. Hybrid CNN–Transformer architecture.

IV. Experimental Results and Discussion

A. Evaluation Indicators

Commonly used evaluation indicators for classification problems are accuracy (ACC), precision (PRE), recall (i.e., TPR), false-positive rate (FPR), and F1-measure. It is necessary to adopt reasonable evaluation criteria for unbalanced data, including the F1-measure, G-mean, receiver operating characteristic (ROC) curve, and area under the ROC curve (AUC) values.

Accuracy is defined by Eq. (3), which reflects the percentage of correctly predicted samples among the total number of predicted samples:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} . \tag{3}$$

Precision is the ratio of correctly predicted positive samples to the total number of positive samples, as shown in Eq. (4):

$$\text{Precision} = \frac{TP}{TP+FP} . \tag{4}$$

Recall describes the ratio of the number of correctly predicted positive samples to the total number of positive samples as formulated in Eq. (5):

$$\text{Recall} = \frac{TP}{TP+FN} . \tag{5}$$

FPR is the number of false positive samples detected divided by the total number of TN samples, as defined by Eq. (6):

$$FPR = \frac{FP}{FP+TN} . \tag{6}$$

The F1-measure is a comprehensive assessment of precision and recall and represents the harmonic average between them, as defined by Eq. (7):

$$\text{F1-Measure} = 2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2TP}{2TP+FP+FN} \tag{7}$$

The G-mean is a standard that comprehensively considers both recall and accuracy. A high G-mean value indicates good modularity, reflecting the geometric mean of sensitivity (i.e., hit rate or recall) and precision. The G-mean is defined in Eq. (8):

$$G-\text{Mean} = \sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}} . \tag{8}$$

The ROC curve defines TPR and FPR in terms of horizontal and vertical coordinates, respectively. Each threshold corresponds to a point (FPR, TPR), and all points are connected as the threshold changes.

Although the ROC curve can comprehensively and intuitively express the performance of a classifier, it cannot provide a specific value. Therefore, it is usually evaluated using the area AUC, as defined in Eq. (9):

$$AUC = \frac{TP \times FP + 2TP \times TN + FN \times TN}{2(TP + FN) \times (FP + TN)} . \quad (9)$$

AUC values range from zero to one; the larger the AUC, the better the classification performance.

B. Experimental Results

The experiment was conducted on a desktop Intel 3.10 GHz processor with 64-GB memory, no GPU acceleration, and a 64-bit Windows 11 operating system. The programming tool was Keras 2.9.0, based on TensorFlow. The NSL-KDD and CICIDS 2017 datasets were used to train the model shown in Fig. 1. For the NSL-KDD dataset, owing to the small amount of data, the batch size was set to 256, and the training epochs were set to 200. The CICIDS2017 dataset contains a considerable amount of data. To accelerate the convergence speed of the model, the batch size was set to 512, and 40 epochs of training were performed. Finally, the model parameters with the best effects on the corresponding datasets were obtained. Subsequently, the model with the optimal parameters was tested on the testing set to obtain classification results, and the confusion matrix was constructed as shown in Fig. 4 and Fig. 5.

Multiple classification experiments were conducted for different attack categories. The NSL-KDD dataset included normal, DoS, Probe, U2R, and R2L classes. The CICIDS 2017 dataset consisted of BENIGN and five attack classes: bot, brute-force, DoS/DDoS, PortScan, and web types. The experimental results are presented in Tables V and VI, respectively. For most class samples, the classification performance of the model was good. For the minority class samples, the model's classification performance decreased to some extent; however, the degree of decrease was not significant. The model does not sacrifice the classification performance of other categories to improve the classification accuracy of any specific category. Therefore, the overall classification performance of the model is very well-balanced.

The overall classification results of the model are presented in Table VII. Although the overall accuracy was not very high, the model did not sacrifice the classification effects of a few classes in exchange for higher overall accuracy, which is a unique demonstration of superior classification procedures. Therefore, the model showed little difference in the classification effects between the majority and minority classes. Moreover, it tended to improve the recognition rate of minority classes (e.g., U2R and R2L) in the NSL-KDD dataset and Bot and Web classes in the CICIDS2017 Dataset).

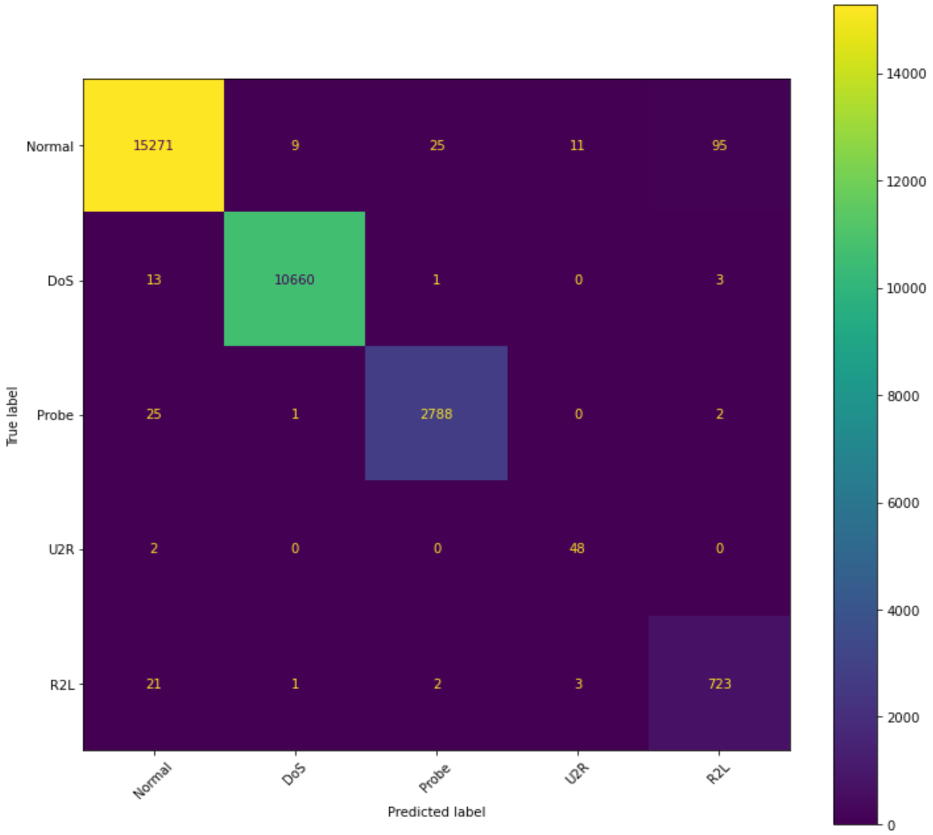

Fig. 4. Confusion matrix of classification results of the NSL-KDD dataset.

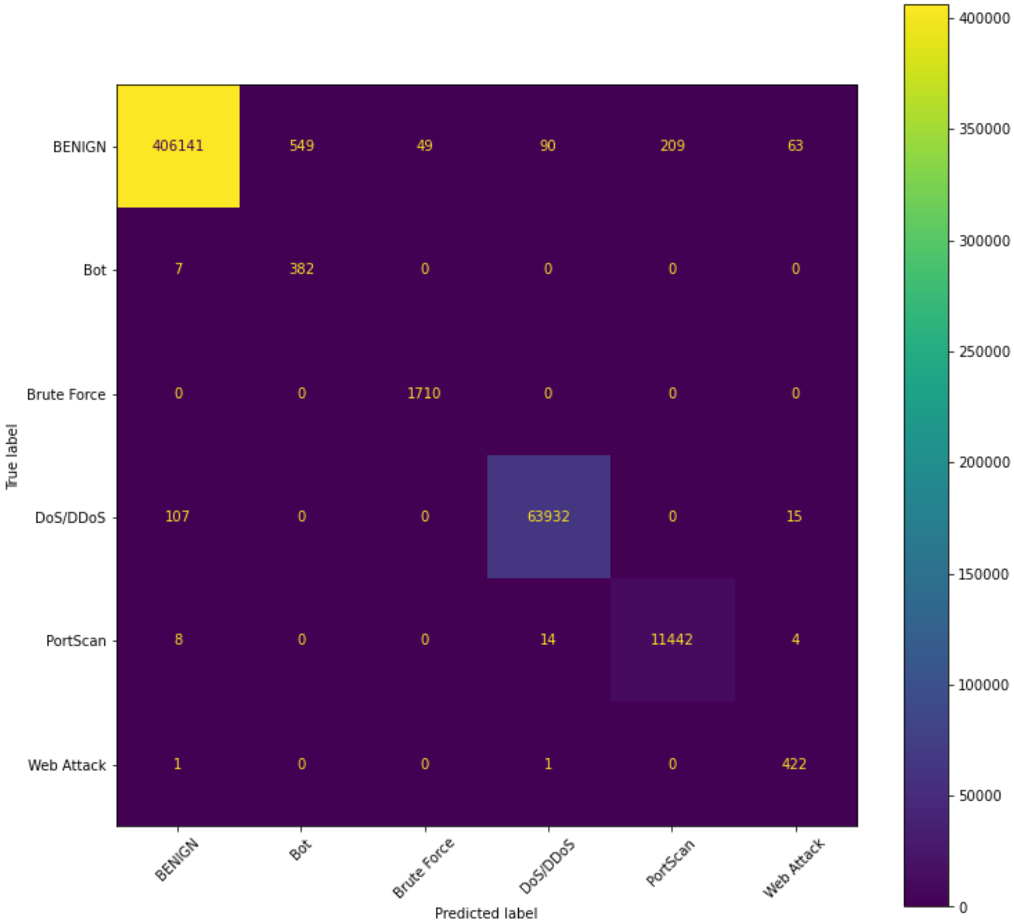

Fig. 5. Confusion matrix of classification results of the CICIDS2017 dataset.

TABLE V. FIVE CLASSIFICATION RESULTS FOR THE NSL-KDD DATASET

Type	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1 (%)	G-mean (%)	AUC
Normal	99.32	99.60	99.09	0.43	99.35	99.33	99.97
DOS	99.91	99.90	99.84	0.06	99.87	99.89	99.99
Probe	99.81	99.01	99.01	0.10	99.01	99.45	99.98
U2R	99.95	77.42	96.00	0.05	85.71	98.01	99.95
R2L	99.57	87.85	96.40	0.35	91.93	97.96	99.77

TABLE VI. SIX CLASSIFICATION RESULTS FOR THE CICIDS2017 DATASET

Type	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	F1 (%)	G-mean (%)	AUC
BENIGN	99.78	99.97	99.76	0.16	99.87	99.80	99.99
Bot	99.89	41.03	98.20	0.11	57.88	99.04	99.89
Brute Force	99.99	97.21	100.00	0.01	98.59	99.99	1.00
DoS / DDoS	99.95	99.84	99.81	0.02	99.82	99.89	1.00
PortScan	99.95	98.21	99.77	0.04	98.98	99.86	99.99
Web Attack	99.98	83.73	99.53	0.02	90.95	99.76	1.00

TABLE VII. MODEL CLASSIFICATION RESULTS

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
NSL-KDD	99.28	92.75	98.07	95.17
CICIDS 2017	99.77	86.66	99.51	91.01

C. Analysis and Discussion

1) Impact of model structure on results: In this section, the structure of the proposed model is discussed. We compared the classification effects of the model before and after data balancing and the single-network model with the hybrid model of both. The following conclusions were drawn from the NSL-KDD dataset, as listed in Table VIII.

The overall effect of the model after data balancing was better than that of the model without data balancing. Moreover, the impact of the hybrid model was better than that of the single-network model.

At the same time, data balancing is beneficial for improving the classification effect of minority classes. Fig. 6 and Fig. 7 show the comparison of precision before and after data balancing for the minority classes U2R and R2L, respectively. From the figures, we can see that regardless of whether it is a single algorithm model or a hybrid model, the classification accuracy after data balancing has increased to varying degrees. This also confirms the necessity of data balancing operations.

Similar conclusions were drawn for the CICIDS2017 dataset. The effect of the hybrid model was better than that of the single network model. Data balancing provided better improvements to accuracy and precision indicators, as shown in Table IX.

Fig. 8 and Fig. 9 present a comparative analysis of the precision of rare classes—Bot and Web Attack—in the CICIDS2017 dataset, both before and after the application of data balancing techniques. Similar to the NSL-KDD dataset, the conclusion drawn from these figures is that data balancing is beneficial for improving the classification accuracy of minority classes.

TABLE VIII. COMPARISON OF THE RESULTS OF THE NSL-KDD DATASET UNDER DIFFERENT MODEL CONFIGURATIONS

	Before data balancing			After data balancing		
Model	***CNN***	***Transformer***	***CNN+Transformer***	***CNN***	***Transformer***	***CNN+Transformer***
Accuracy (%)	98.47	98.41	98.56	99.24	98.81	99.22
Precision (%)	82.36	80.82	82.62	91.94	86.18	91.68
Recall (%)	97.98	97.47	98.07	98.04	97.82	98.19
F-Measure (%)	87.31	85.35	87.51	94.64	90.47	94.56

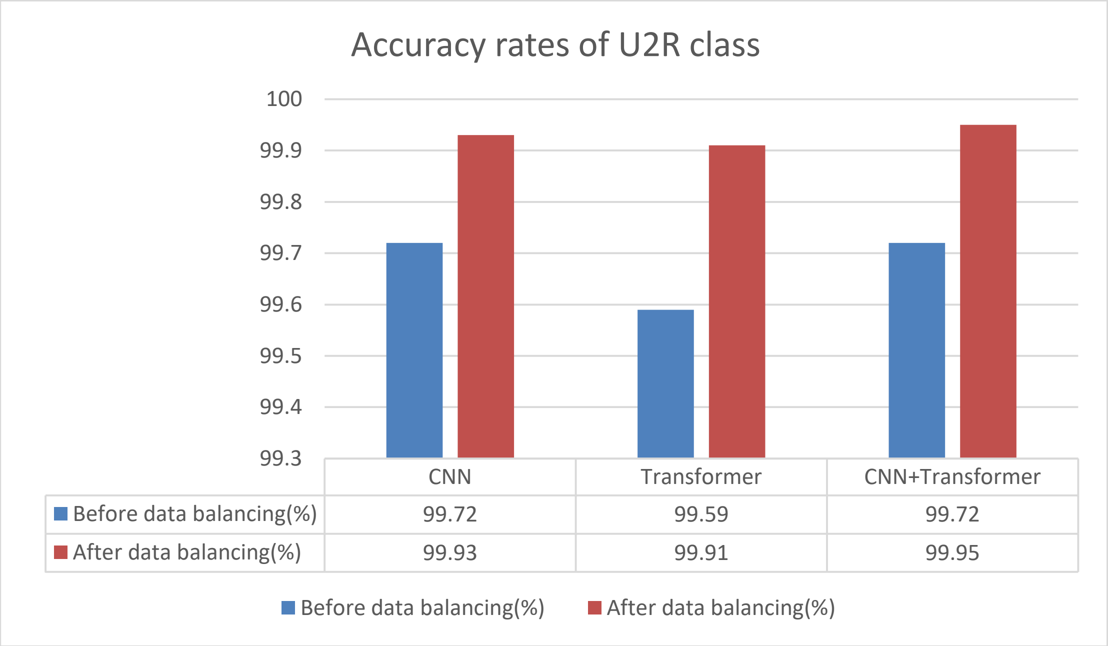

Fig. 6. Comparison of the U2R class accuracy rate before and after data balancing.

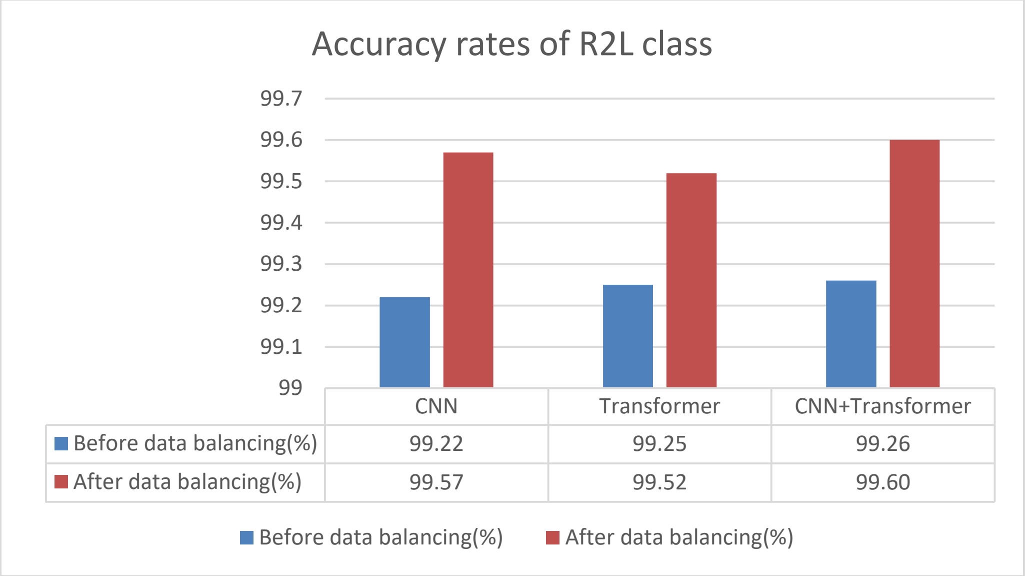

Fig. 7. Comparison of the R2L class accuracy rate before and after data balancing.

TABLE IX. COMPARISON OF THE RESULTS OF THE CICIDS2017 DATASET UNDER DIFFERENT MODEL CONFIGURATIONS

	Before data balancing			After data balancing		
Model	***CNN***	***Transformer***	***CNN+Transformer***	***CNN***	***Transformer***	***CNN+Transformer***
Accuracy (%)	98.55	97.15	98.36	99.73	98.24	99.77
Precision (%)	69.38	67.12	71.24	85.84	69.53	86.66
Recall (%)	99.34	98.14	99.52	99.55	99.16	99.51
F-Measure (%)	73.21	69.34	75.15	90.26	72.52	91.01

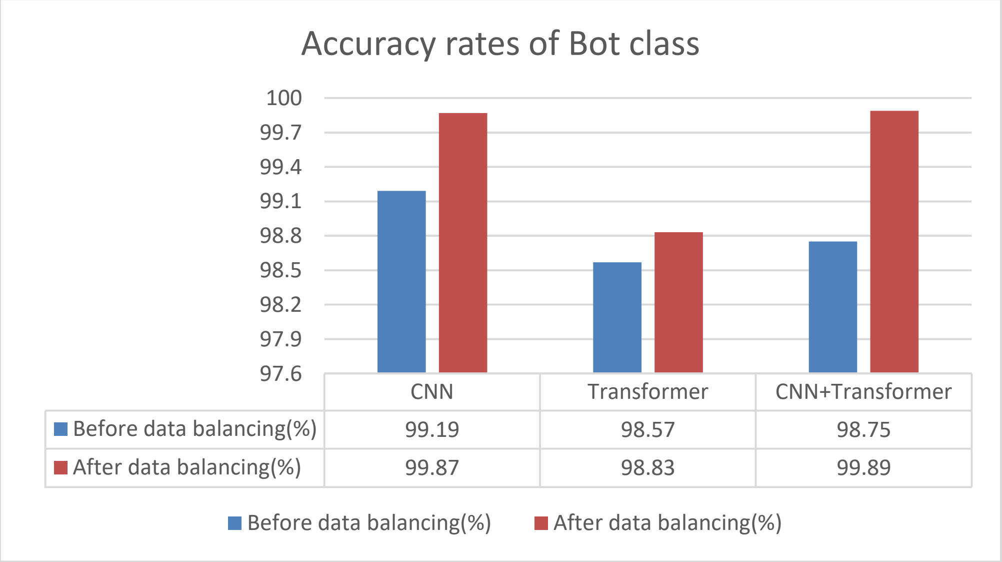

Fig. 8. Comparison of Bot class accuracy rates before and after data balancing.

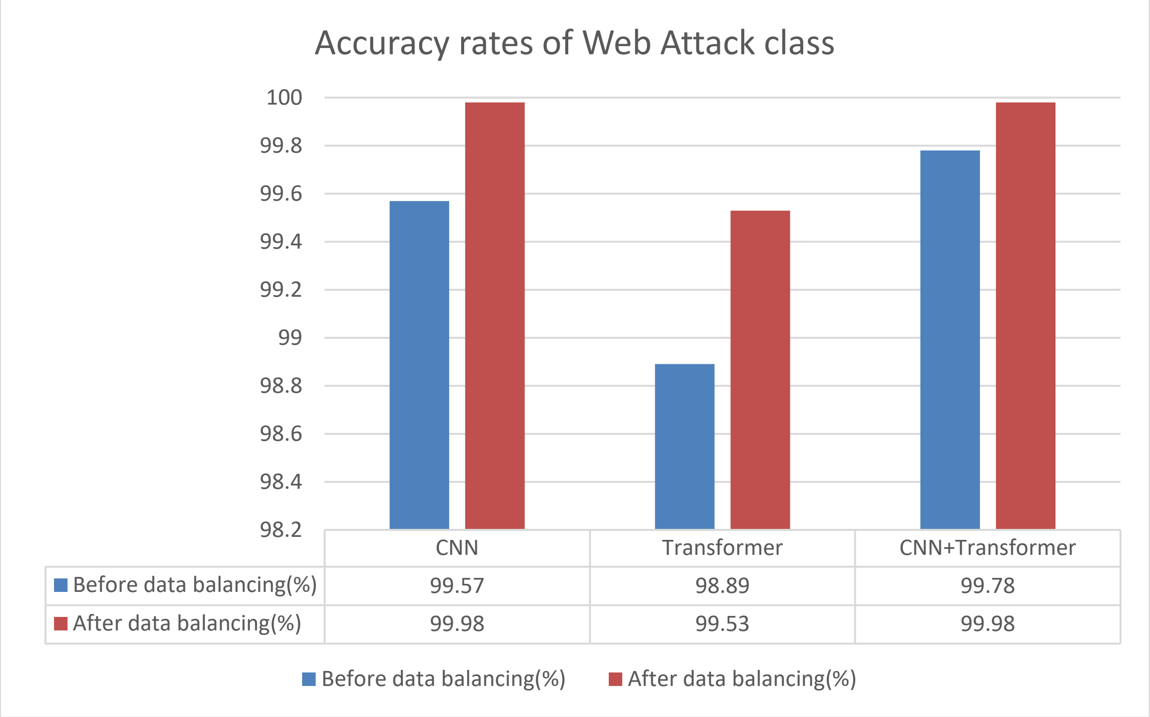

Fig. 9. Comparison of Web Attack class accuracy rate before and after data balancing.

2) Impact of sampling rate on results: The previous section showed that data balancing benefits minority class detection. In this section, we focus on comparing the different sampling rates of rare classes to explore the impact of sampling rates. For the NSL-KDD dataset, we checked the U2R category. In contrast, for the CICIDS2017 dataset, we checked the Bot and Web Attack categories due to their low representation. During model training, 100, 300, 500, and 1,000% samples were considered for the given categories, and the optimal model parameters generated were predicted using the testing set. The experimental results are presented in Tables X to XII.

TABLE X. COMPARISON OF RESULTS FOR THE U2R CATEGORY UNDER DIFFERENT SAMPLING RATES

Sampling rate (%)	Recall (%)	Accuracy (%)	F-Measure (%)	G-mean (%)
100	70.00	99.93	76.09	83.66
300	72.00	99.92	75.79	84.84
500	74.00	99.91	74.00	86.00

TABLE XI. COMPARISON OF RESULTS FOR THE BOT CATEGORY UNDER DIFFERENT SAMPLING RATES

Sampling rate (%)	Recall (%)	Accuracy (%)	F-Measure (%)	G-mean (%)
100	32.65	99.95	52.24	57.14
300	61.44	99.94	62.41	78.37
500	94.34	99.93	67.82	97.09
1000	94.86	99.93	69.82	97.19

TABLE XII. COMPARISON OF RESULTS FOR THE WEB ATTACK CATEGORY UNDER DIFFERENT SAMPLING RATES

Sampling rate (%)	Recall (%)	Accuracy (%)	F-Measure (%)	G-mean (%)
100	82.08	99.99	96.43	90.59
300	95.52	99.99	96.36	97.72
500	97.41	99.99	95.10	98.67
1000	97.64	99.99	94.49	98.80

These results show that increasing the sampling rate significantly improved the recall rate, F-measure, and G-mean for rare categories. However, this had little impact on overall classification accuracy. Due to the small proportions of rare classes in the original dataset, it was difficult for the model to train effectively for class recognition. Therefore, increasing the sampling rate is equivalent to increasing the training opportunities of the model for that category, thereby improving the recall of subsequent testing data. Thus, improving the detection rate for minority classes comes at the cost of increasing training time.

D. Comparisons of Experimental Results

We compared the above experimental results with methods from the relevant literature to verify our model’s effectiveness with multiclassification problems using unbalanced data. We first compared NSL-KDD data, as the related literature is abundant.

First, the classification accuracy of multiple classifications was compared, as presented in Table XIII. Our model had the highest classification accuracy for all five categories, and there were no cases in which the accuracy of a specific category was particularly low. Again, the accuracy of a few categories was not sacrificed in exchange for higher overall accuracy.

TABLE XIII. ACCURACY COMPARISONS OF FIVE CLASSIFICATIONS

References	Normal (%)	DOS (%)	Probe (%)	U2R (%)	R2L (%)
Zhang-2019- [39]	-	99.45	99.37	98.68	97.78
Ahsan-2020- [43]	98.5	98.8	0	99.4	94.6
LIU-2023- [63]	97.7	94.6	94.7	0.3	0.4
Proposed model	99.32	99.91	99.81	99.95	99.57

Next, multi-classification recall rates were compared, and the results are listed in Table XIV. It can be seen from the table that the recall rates of the DOS and R2L categories were the highest compared with those reported in the relevant literature. The difference between the other three categories and the highest values in the literature was insignificant.

TABLE XIV. RECALL COMPARISONS OF FIVE CLASSIFICATIONS

References	Normal (%)	DOS (%)	Probe (%)	U2R (%)	R2L (%)
Zhang-2019- [39]	-	99.7	99.4	98.2	93.4
Albahar-2020- [42]	98	97.8	95.6	96.9	92.4
Ahsan-2020- [43]	98.5	98.8	0	99.4	94.6
Onah-2021- [64]	97.5	96.9	93.4	73.5	77.1
LIU-2023- [63]	97.7	94.6	94.7	0	0
Kamalakkannan-2023- [45]	99.57	99.76	99.15	25	88.41
Proposed model	99.09	99.84	99.01	96	96.40

The classification FPRs of multiple classifications were then compared, as shown in Table XV. It can be seen that, apart from a few R2L cases, the FPR of our model was the lowest of all.

TABLE XV. FPR COMPARISONS OF FIVE CLASSIFICATIONS

References	Normal (%)	DOS (%)	Probe (%)	U2R (%)	R2L (%)
Zhang-2019- [39]	-	0.8	0.7	1.8	7.3
Albahar-2020- [42]	0.73	0.54	0.67	0.33	0.87
Ahsan-2020- [43]	1.5	1.2	1	0.6	5.4
Onah-2021- [64]	0.6	0.6	0.4	0.2	0.1
Proposed model	0.43	0.06	0.10	0.05	0.35

Finally, for imbalanced data classification problems, the F1 measure is often more important than other metrics. Table XVI presents the results of the multicategory F1-measure comparisons. Apart from the U2R category, the F1 measure of our model was the best.

TABLE XVI. F1-MEASURE COMPARISON OF FIVE CLASSIFICATIONS

References	Normal (%)	DOS (%)	Probe (%)	U2R (%)	R2L (%)
Albahar-2020- [42]	98.5	98.3	94.8	97.3	66.5
Ahsan-2020- [43]	99.1	98.3	0	99.2	85.4
LIU-2023- [63]	98.9	97.2	97.3	0.5	0.7
Proposed model	99.35	99.87	99.01	85.71	91.93

Using the CICIDS2017 dataset, our model also showed advantages in accuracy and recall, as shown in Table XVII.

TABLE XVII. COMPARISON OF THE RESULTS OF THE CICIDS2017 DATASET

References	Accuracy (%)	Recall (%)
Abdel-Basset-2021- [65]	99.69	96.29
Khan-2021- [66]	98.76	98.69
Chen-2022- [67]	99.73	79.13
Wu-2022- [68]	99.35	98.83
Proposed model	99.77	99.51

Through the above comparative analyses, our hybrid model, based on data balancing and two deep learning networks, has clear advantages and achieved excellent results in multiclassification problems with unbalanced data.

V. CONCLUSIONS AND FUTURE RECOMMENDATIONS

NIDS plays vital network security roles in identifying, preventing, and countering network threats. Owing to the large amount of unbalanced data collected in network datasets, FPs and omissions significantly reduce the detection efficiency of extant IDSs. This paper proposed a deep learning model that combines data balancing and a CNN + Transformer hybrid to improve the data distribution of the original dataset via undersampling and oversampling techniques. Our data redistribution method increases the likelihood of identifying minority classes based on model training, and the experimental results show that our innovations effectively improve this detection rate. Our hybrid model’s algorithm-level improvements increased recognition training based on fused spatiotemporal features, and the experimental results show that the proposed system, combined with multiple combined processes, identifies anomalies more efficiently and accurately than any single network model.

For the classic NSL-KDD and modern CICIDS2017 datasets, our model was more effective in multiclassification data applications and was superior to existing IDS models in terms of accuracy, FPR, F1-mean, and other indicators. Notably, the CICIDS2017 dataset showed superiority in training compared with existing models in terms of accuracy and recall.

Although the model proposed in this paper has advantages over existing systems, several other data balancing activities, such as the edited nearest neighbor, Tomek–Links, SMOTEBoost, and ADASYN methods described, should be tested. Many LSTM, GRU, DBN, and other variants should also be tested. The objective is to improve the detection effects of data classifications based on innovative model structures so that network security professionals and scholars can obtain better IDS results, even in the face of scarce data.

ACKNOWLEDGMENT

This study was supported by the Natural Science Foundation of Fujian Province, China [grant no. 2021J01332].

REFERENCES

[1] N. Gupta, V. Jindal, P. Bedi, A survey on intrusion detection and prevention systems, SN Comput. Sci. 4 (2023) 439. https://doi.org/10.1007/s42979-023-01926-7.

[2] A. Das, S.G. Balakrishnan, 2021 International RTEICT Conference, Bangalore, India, 2021, pp. 555–562. https://doi.org/10.1109/RTEICT52294.2021.9573685.

[3] Alkasassbeh, M., Al-Haj Baddar, S. Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey. Arab J Sci Eng 48, 10021–10064 (2023). https://doi.org/10.1007/s13369-022-07412-1

[4] H.K. Shaikha, W.M. Abdullah, A review of intrusion detection systems, Acad. J. Nawroz Univ. 6 (2017) 101–105. https://doi.org/10.25007/AJNU.V6N3A90.

[5] E.M. Maseno, Z. Wang, H. Xing, A systematic review on hybrid intrusion detection system, Sec. Commun. Netw. 2022 (2022) article ID 9663052. https://doi.org/10.1155/2022/9663052.

[6] G.M. Weiss, Mining with rarity: A unifying framework, SIGKDD Explor. Newsl. 6 (2004) 7–19. https://doi.org/10.1145/1007730.1007734.

[7] Dataset link, NSL-KDD dataset, 2009. http://nsl.cs.unb.ca/KDD/NSL-KDD.html.

[8] Dataset link, CICIDS2017 dataset, 2017. https://www.unb.ca/cic/datasets/IDS-2017.html.

[9] H. Sharma, A. Gosain, Oversampling methods to handle the class imbalance problem: A review, 2023. https://doi.org/10.1007/978-3-031-27609-5_8.

[10] S. Sharma, A. Gosain, S. Jain, A review of the oversampling techniques in class imbalance problem, in: A. Khanna, D. Gupta, S. Bhattacharyya, A.E. Hassanien, S. Anand, A. Jaiswal (Eds.), Adv. Intell. Syst. Comput., 1387 International Conference on Innovative Computing and Communications, Springer, Singapore, 2022. https://doi.org/10.1007/978-981-16-2594-7_38.

[11] S. Szeghalmy, A. Fazekas, A highly adaptive oversampling approach to address the issue of data imbalance, Computers. 11 (2022) 73. https://doi.org/10.3390/computers11050073.

[12] A. Islam, S.B. Belhaouari, A.U. Rehman, H. Bensmail, KNNOR: An oversampling technique for imbalanced datasets, Appl. Soft Comput. 115 (2022) 108288, ISSN 1568-4946. https://doi.org/10.1016/j.asoc.2021.108288.

[13] Y. Liu, Y. Liu, B.X.B. Yu, S. Zhong, Z. Hu, Noise-robust oversampling for imbalanced data classification, Pattern Recog. 133 (2023) 109008,ISSN 0031-3203. https://doi.org/10.1016/j.patcog.2022.109008.

[14] N. Altwaijry, Probability-based synthetic minority oversampling technique, IEEE Access. 11 (2023) 28831–28839. https://doi.org/10.1109/ACCESS.2023.3260723.

[15] D. Devi, S.K. Biswas, B. Purkayastha, A review on solution to class imbalance problem: Undersampling approaches, 2020 International ComPE, Shillong, India, 2020, pp. 626–631. https://doi.org/10.1109/ComPE49325.2020.9200087.

[16] M.A. Arefeen, S.T. Nimi, M.S. Rahman, Neural network-based undersampling techniques, IEEE Trans. Syst. Man Cybern. Sys. 52 (2022) 1111–1120. https://doi.org/10.1109/TSMC.2020.3016283.

[17] L. Jiang, P. Yuan, J. Liao, Q. Zhang, J. Liu, K. Li, Undersampling of approaching the classification boundary for imbalance problem, Concurrency Comput. Pract. Experience. 35 (2023) 1. https://doi.org/10.1002/cpe.7586.

[18] T. Liang, J. Xu, B. Zou, Z. Wang, J. Zeng, LDAMSS: Fast and efficient undersampling method for imbalanced learning, Appl. Intell. 52 (2022) 6794–6811. https://doi.org/10.1007/s10489-021-02780-x.

[19] M. Bach, New undersampling method based on the kNN approach, Procedia Comput. Sci. 207 (2022) 3403–3412. https://doi.org/10.1016/j.procs.2022.09.399.

[20] C. Cui, J. Wang, W. Wei, J. Liang, Hybrid sampling-based contrastive learning for imbalanced node classification, Int. J. Mach. Learn. Cybernet. 14 (2023) 989–1001. https://doi.org/10.1007/s13042-022-01677-6.

[21] L. Wang, S. Liu, An improved random forest algorithm based on hybrid sampling and feature selection. Nanjing Youdian Daxue Xuebao (Ziran Kexue Ban), J. Nanjing Univ. Posts Telecommun. (Nat. Sci.). 42 (2022) 81–89.

[22] R.A. Sowah, B. Kuditchar, G.A. Mills, A. Acakpovi, R.A. Twum, G. Buah, R. Agboyi, HCBST: An efficient hybrid sampling technique for class imbalance problems, ACM Trans. Knowl. Discov. Data. 16 (2022) 1–37. https://doi.org/10.1145/3488280.

[23] M. Han, A. Li, Z. Gao, D. Mu, S. Liu, Hybrid sampling and dynamic weighting-based classification method for multi-class imbalanced data stream, Appl. Sci. 13 (2023) 5924. https://doi.org/10.3390/app13105924.

[24] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357. https://doi.org/10.1613/jair.953.

[25] D. Dablain, B. Krawczyk, N.V. Chawla, DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data, IEEE Trans. Neural Netw. Learn. Sys. 34 (2023) 6390–6404. https://doi.org/10.1109/TNNLS.2021.3136503.

[26] J.H. Joloudari, A. Marefat, M.A. Nematollahi, S.S. Oyelere, S. Hussain, Effective class-imbalance learning based on SMOTE and convolutional neural networks, Appl. Sci. 13 (2023) 4006. https://doi.org/10.3390/app13064006.

[27] H. Kheddar, Y. Himeur, A.I. Awad, Deep transfer learning applications in intrusion detection systems: A comprehensive review (2023). https://doi.org/10.48550/arXiv.2304.10550.

[28] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access. 5 (2017) 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418.

[29] R.K. Vigneswaran, R. Vinayakumar, K.P. Soman, P. Poornachandran, Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security, 9th ICCCNT, Bengaluru, India, 2018, pp. 1–6. https://doi.org/10.1109/ICCCNT.2018.8494096.

[30] Dataset link, KDD CUP 1999 dataset, 1999. https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data.

[31] Y. Xiao, C. Xing, T. Zhang, Z. Zhao, An intrusion detection model based on feature reduction and convolutional neural networks, IEEE Access. 7 (2019) 42210–42219. https://doi.org/10.1109/ACCESS.2019.2904620.

[32] O. Belarbi, A. Khan, P. Carnelli, T. Spyridopoulos, An intrusion detection system based on deep belief networks. Sci. Cyber Sec. (2022). https://doi.org/10.48550/arXiv.2207.02117.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need[J].arXiv, 2017. https://DOI.org/10.48550/arXiv.1706.03762.

[34] W. Wang, S. Jian, Y. Tan, Q. Wu, C. Huang, Robust unsupervised network intrusion detection with self-supervised masked context reconstruction, Comput. Sec. 128 (2023)103131, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2023.103131.

[35] Y.G. Yang, H.M. Fu, S. Gao, Y.H. Zhou, W.M. Shi. Intrusion detection: A model based on the improved vision transformer. Trans. Emerg. Telecommun. Technol. 33 (2022). https://doi.org/10.1002/ett.4522.

[36] B. Cao, C. Li, Y. Song, Y. Qin, C. Chen, Network intrusion detection model based on CNN and GRU, Appl. Sci. 12 (2022) 4184. https://doi.org/10.3390/app12094184.

[37] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, Intrusion detection by machine learning: A review, Expert Syst. Appl. 36 (2009) 11994–12000, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2009.05.029.

[38] U.S. Musa, S. Chakraborty, M.M. Abdullahi, T. Maini, A review on intrusion detection system using machine learning techniques, 2021 ICCCIS, Greater Noida, India, 2021, pp. 541–549. https://doi.org/10.1109/ICCCIS51004.2021.9397121.

[39] Y. Zhang, P. Li, X. Wang, Intrusion detection for IoT based on improved genetic algorithm and deep belief network, IEEE Access. 7 (2019) 31711–31722. https://doi.org/10.1109/ACCESS.2019.2903723.

[40] P. Wu, H. Guo, LuNet: A deep neural network for network intrusion detection, IEEE SSCI, Xiamen, China, 2019, pp. 617–624. https://doi.org/10.1109/SSCI44817.2019.9003126.

[41] C.A. de Souza, C.B. Westphall, R.B. Machado, J.B.M. Sobral, G. Vieira, Hybrid approach to intrusion detection in fog-based IoT environments, Comput. Netw. 180 (2020). https://doi.org/10.1016/j.comnet.2020.107417.

[42] M.A. Albahar, M. Binsawad, J. Almalki, S. El-etriby, S. Karali, Improving Intrusion Detection System using Artificial Neural Network, IJACSA. 11 (2020). http://doi.org/10.14569/IJACSA.2020.0110670.

[43] M. Ahsan, K.E. Nygard, Convolutional neural networks with LSTM for intrusion detection, International Conference on Computers and their Applications, 2020. https://doi.org/10.13140/RG.2.2.24796.82567.

[44] A.M. Banaamah, I. Ahmad, Intrusion detection in IoT using deep learning, Sensors (Basel). 22 (2022) 8417. https://doi.org/10.3390/s22218417, http://www.ncbi.nlm.nih.gov/pubmed/36366115, PMC9658941.

[45] D. Kamalakkannan, D. Menaga, S. Shobana, K.V. Daya Sagar, R. Rajagopal, M. Tiwari, A detection of intrusions based on deep learning, Cybern. Sys. (2023) 1–15. https://doi.org/10.1080/01969722.2023.2175134.

[46] I. Shivhare, J. Purohit, V. Jogani, S. Attari, M. Chandane, Intrusion detection: A deep learning approach (2023). https://doi.org/10.48550/arXiv.2306.07601.

[47] E.U.H. Qazi, M.H. Faheem, T. Zia, HDLNIDS: Hybrid deep-learning-based network intrusion detection system, Appl. Sci. 13 (2023) 4921. https://doi.org/10.3390/app13084921.

[48] N. Xing, S. Zhao, Y. Wang, K. Ning, X. Liu, A dynamic intrusion detection system capable of detecting unknown attacks, IJACSA, 14(2023). http://dx.doi.org/10.14569/IJACSA.2023.0140743.

[49] Z. Xiang, X. Li, 2023. Fusion of transformer and ML-CNN-BiLSTM for network intrusion detection. EURASIP J. Wirel. Commun. Netw. 2023, p. 1. https://doi.org/10.1186/s13638-023-0227

[50] F. Ullah, S. Ullah, G. Srivastava, J.C.W. Lin, IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic, Dig. Commuin. Netw. 2023, ISSN 2352-8648. https://doi.org/10.1016/j.dcan.2023.03.008.

[51] Y. Binghao, H. Guodong, Combinatorial intrusion detection model based on deep recurrent neural network and improved SMOTE algorithm, Chin. J. Netw. Inf. Sec. 4 (2018) 48–59. https://doi.org/10.11959/j.issn.2096-109x.2018056.

[52] S. Al, M. Dener, STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment, Comput. Secur. 110 (2021). https://doi.org/10.1016/j.cose.2021.102435.

[53] S. Revathi, A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. 2 (2013).

[54] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, 2009, pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528.

[55] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, International Conference on Information Systems Security and Privacy, 2018. https://doi.org/10.5220/0006639801080116.

[56] P. Gao, M. Yue, Z. Wu, A novel intrusion detection method based on WOA optimized hybrid kernel RVM, 2021 IEEE 6th ICCCS, Chengdu, China, 2021, pp. 1063–1069. https://doi.org/10.1109/ICCCS52626.2021.9449199.

[57] Z. Chen, J. Duan, L. Kang, G. Qiu, A hybrid data-level ensemble to enable learning from highly imbalanced dataset, Inform. Sci. 554 (2021) 157–176, ISSN 0020-0255. https://doi.org/10.1016/j.ins.2020.12.023.

[58] B. Mirzaei, B. Nikpour, H. Nezamabadi-pour, CDBH: A clustering and density-based hybrid approach for imbalanced data classification, Exp. Sys. Appl. 164 (2021) 114035, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2020.114035.

[59] M.A. Khan, M.R. Karim, Y. Kim, A scalable and hybrid intrusion detection system based on the convolutional-LSTM network, Symmetry. 11 (2019) 583. https://doi.org/10.3390/sym11040583.

[60] W. Cui, Q. Lu, A.M. Qureshi, W. Li, K. Wu, An adaptive LeNet-5 model for anomaly detection, Inf. Sec. J. Glob. Perspect. 30 (2021) 19–29. https://doi.org/10.1080/19393555.2020.1797248.

[61] S. Al, M. Dener, STL-HDL, STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment, Comput. Sec. 110 (2021) 102435. https://doi.org/10.1016/j.cose.2021.102435.

[62] J. Yuan, A. Zhu, Q. Xu, K. Wattanachote, Y. Gong. CTIF-Net: A CNN-transformer iterative fusion network for salient object detection. IEEE Trans. Circ. Sys. Video Technol. (2023). https://DOI.org/10.1109/tcsvt.2023.3321190.

[63] J.S. Liu, D.Y. Zhan, J. Deng, L.N. Wang, Network intrusion detection based on deep neural network and federated learning, Comput. Eng. 49 (2023) 15–21, 30.

[64] J.O. Onah, S.M., Abdulhamid, M. Abdullahi, I.H. Hassan, A. Al-Ghusham, Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment, Mach. Learn. Applic.. 6 (2021) 100156, ISSN 2666-8270. https://doi.org/10.1016/j.mlwa.2021.100156.

[65] M. Abdel-Basset, H. Hawash, R.K. Chakrabortty, M.J. Ryan, Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks, IEEE Internet Things J. 8, 1 Aug. (2021) 12251–12265. https://doi.org/10.1109/JIOT.2021.3060878.

[66] A.S. Khan, Z. Ahmad, J. Abdullah, F. Ahmad, A spectrogram image-based network anomaly detection system using deep convolutional neural network, IEEE Access. 9 (2021) 87079–87093. https://doi.org/10.1109/ACCESS.2021.3088149.

[67] Y. Chen, Q. Lin, W. Wei, J. Ji, K.-C. Wong, C.A. Coello, Intrusion detection using multi-objective evolutionary convolutional neural network for internet of things in fog computing. Knowl. Based Sys. 244 (2020). https://doi.org/10.1016/j.knosys.2022.108505.

[68] Z. Wu, H. Zhang, P. Wang, Z. Sun, RTIDS: A robust transformer-based approach for intrusion detection system, IEEE Access, 10 (2022), 64375–64387. https://doi.org/10.1109/ACCESS.2022.3182333.