

# SA-BiGRU: Sequential Attention and BiGRU Network for Intrusion Detection System in SDN

Guo-Qing Zhang, Gui-Qin Yang\*, Yong-Qi Hou, and Zhi-Qi Liu

School of Electronic information and Engineering, Lanzhou Jiaotong University,  
Lanzhou, Gansu 730070, China

{1452064810, 1261397198, 2573156103, 2247705299}@qq.com

*Received 11 March 2025; Revised 19 July 2025; Accepted 4 August 2025*

**Abstract.** Software-Defined Networking (SDN) decouples the data plane from the control plane, centralizing control logic and enabling programmability, which provides far greater flexibility and manageability than traditional networks. However, this architecture also makes the SDN controller a prime target for attacks that can bring down the entire network, rendering intrusion detection in SDN critically important. Further, imbalance in the relevant dataset will cause the model to learn features of the majority classes while neglecting those of the minority classes, which will severely compromise the detection rate of the intrusion detection model. To address this issue, this paper proposes a hybrid sampling strategy: applying random undersampling to majority-class samples while employing SMOTE-ENN oversampling for minority-class samples. This approach generates a more balanced dataset and enhances the detection rate for minority classes. To mitigate diverse threats in SDN, this work introduces the SA-BiGRU intrusion detection framework. The model employs Sequential Attention (SA) for dimensionality-aware feature selection to eliminate redundant information, followed by BiGRU for capturing critical temporal dynamics in network traffic. Extensive experiments on the CICIDS2017, NSL-KDD, and CICDDOS2019 datasets demonstrate that SA-BiGRU achieves superior detection accuracy and precision, reliably identifying a wide range of attack types and outperforming baseline methods.

**Keywords:** intrusion detection, Sequential Attention, SDN, BiGRU, data set balance processing

## 1 Introduction

Software Defined Networking (SDN) represents a novel network paradigm that separates the data-forwarding plane from the control plane, enabling flexible network management and centralized policy enforcement [1]. This architecture greatly improves resource utilization and programmability compared to traditional networks. However, the centralized controller also introduces a single point of failure and becomes a prime target for attackers, since any compromise of the controller can severely undermine network stability and security. Against this backdrop, designing a fast, accurate, and robust Intrusion Detection System (IDS) to identify and block diverse network attacks has become critical for SDN security.

As the Internet continues to expand, the volume and sophistication of network-based intrusions, malware, and other threats have escalated. Attackers employ ever more complex evasion techniques, demanding that IDSs adapt to both greater diversity and higher stealth. Given the rapid emergence of new threats and their subtle differences from legitimate traffic, traditional heuristic algorithms and firewalls no longer suffice. In recent years, machine-learning and deep-learning approaches have achieved notable success in conventional network IDSs [2]. Classical classifiers such as Support Vector Machines (SVM), Random Forests (RF), and Decision Trees (DT) perform well on binary classification (normal vs. attack) by extracting shallow features, but they struggle in multi-class scenarios and high-dimensional feature spaces. Deep models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants, automatically learn rich representations and show promise for real-time detection [3]. However, these models primarily target traditional networks and often suffer from low recall on minority-class samples as well as limited generalization when trained on small or imbalanced datasets. Preprocessing techniques such as Synthetic Minority Oversampling Technique (SMOTE),

---

\* Corresponding Author

Random Undersampling, or ensemble methods have been applied to mitigate class imbalance, but most focus on overall accuracy and overlook precise detection of more damaging minority-class attacks. A few studies have introduced attention mechanisms to reduce feature redundancy in deep models, but a systematic, SDN-specific solution remains lacking.

To address these challenges, we propose Sequential Attention–Bidirectional Gated Recurrent Units (SA-BiGRU), an SDN intrusion detection model that combines a Sequential Attention mechanism with a hybrid sampling strategy to detect and classify attacks in SDN environments. Our key contributions are:

1. **Hybrid Sampling Strategy:** We apply random undersampling to the majority-class and Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTE-ENN) to the minority-class samples, producing a more balanced dataset and significantly improving detection rates for rare attacks.

2. **SA-BiGRU Model Design:** We integrate a Sequential Attention mechanism for feature selection, iteratively assigning attention weights to identify and retain the most important features and thereby reducing dimensionality and noise. We then employ BiGRU to capture long-range temporal dependencies.

3. **Multi-Class Detection:** We demonstrate our approach on fourteen attack types (including DDoS, PortScan, and Botnet) within an SDN architecture, formulating a comprehensive multi-class classification framework.

4. **Extensive Evaluation:** On the CICIDS2017 and NSL-KDD datasets, SA-BiGRU outperforms several baseline models in both detection accuracy and precision.

The rest of this paper is organized as follows: Section 2 reviews related work on SDN intrusion detection; Section 3 details the SA-BiGRU model and hybrid sampling method; Section 4 presents our experimental design and analysis; and Section 5 concludes the paper and outlines future research directions.

## 2 Related Work

As cyberspace continues to expand, network intrusions are becoming increasingly diverse and frequent [4]. Protecting user privacy, and ensuring network security therefore hinge on developing fast and effective intrusion detection systems. Although numerous machine-learning and deep-learning-based IDS approaches have been proposed, they still face three key challenges: (1) most methods are confined to specific attack types and cannot handle the multi-class threats present in SDN environments; (2) inadequate treatment of class imbalance leads to poor detection rates for minority-class; and (3) the lack of an end-to-end feature selection mechanism makes it difficult to eliminate redundant noise from high-dimensional temporal traffic data.

K. Singh et al. [5] combined K-Nearest Neighbors with wrapper-based feature selection to achieve 98.3 % accuracy in SDN-based DDoS detection. M. A. Setitra et al. [6] introduced a model combining MLP (Multi-Layer Perceptron) and CNN to enhance DDoS detection in SDN. The model employs the SHAP feature selection technique to optimize features and uses Bayesian Optimization and the Adaptive Moment Estimation (ADAM) Optimizer for parameter tuning. R. Doriguzzi-Corin et al. [7] introduced LUCID, a practical, lightweight CNN-based system for real-time classification of malicious versus benign flows. Zhao et al. [8] proposed a DDoS detector that integrates self-attention with a Convolutional Neural Network-Bidirectional Long Short Term Memory (CNN-BiLSTM) pipeline, employing feature selection, spatio-temporal extraction, and attention weighting to address high dimensionality and low accuracy. M. S. Elsayed et al. [9] addressed the SDN network problem by proposing a novel regularization method combined with CNN to classify traffic as normal or attack categories, thereby tackling the overfitting problem in intrusion detection. Although these methods excel at DDoS-only detection, they lack the ability to adapt to multi-class threats such as PortScan and Botnet attacks.

G. S. Vidhya et al. [10] built a BiLSTM model capable of both binary and multi-class detection in SDN-IoT environments, and M. S. Mahdi [11] applied a GRU network to the CSECICIDS2018 dataset with good classification results. However, single-architecture RNNs often fail to fully exploit complex traffic patterns, resulting in low or inconsistent detection rates for certain attack types [12]. Approaches proposed by Mohammad et al. [12], Hnamte et al. [13], Rachid et al. [14], and other researchers have shown that building intrusion detection models based on hybrid network architectures is an effective strategy [15]. This hybrid network structure combines various types of deep learning models, leveraging their respective advantages to improve both detection accuracy and generalization [16]. Attention mechanisms have also been introduced—M. Cui et al. [17] presented CNNA-BiLSTM for parallel spatio-temporal feature extraction with attention weighting, and Wang et al. [18] converted traffic into sparse images for an attention-based CNN. While these advances enhance feature extraction, they generally lack an effective sampling strategy to balance minority-class.

Most existing models prioritize overall dataset accuracy but overlook the detection rates for minority-class attacks such as PortScan and Botnet. Because these attacks constitute only a small fraction of the data, missing them has little impact on aggregate accuracy. However, their potential to inflict disproportionate harm on the network is significantly greater than that of majority-class attacks, so it is critical to place greater emphasis on detecting minority-class intrusions [19]. M. S. Alshehri [20] applied two-stage data cleansing within a self-attention DCNN to reduce redundancy but did not address undersampling, yielding poor minority-class recall. R. Harini et al. [21] designed a three-layer IDS combining WDNN, CNN-LSTM, and Extreme Gradient Boosting (XGBoost) with One-Sided Selection and Adaptive Synthetic Sampling (ADASYN) sampling to balance classes and achieved nearly 98% overall accuracy. However, their multi-stage design increases deployment complexity, and because ADASYN generates synthetic boundary samples, it can introduce noise and impair the model's generalization.

In contrast, we introduce SA-BiGRU, a single end-to-end architecture that combines a sequential attention mechanism with a BiGRU network to automatically select and deeply model high-dimensional temporal features, thereby boosting detection rates for minority-class attacks. We also propose a hybrid sampling strategy that applies random undersampling followed by SMOTE-ENN, in which the ENN step removes noisy samples while preserving representative minority instances, achieving more effective class balancing without compromising data quality.

### 3 Proposed Methodology

This paper focuses on multi-class intrusion detection in SDN. Fig. 1 illustrates the system architecture: the SDN controller runs a network-monitoring service that captures both attack and normal traffic from switches via Wireshark. Prior to modeling, all data undergoes cleaning, label encoding, and normalization to ensure quality. To mitigate class imbalance, we employ a resampling scheme that combines random undersampling of majority-class with SMOTE-ENN oversampling and noise removal for minority-class samples. This approach reduces redundant majority-class samples while enhancing minority-class representation. We then introduce SA-BiGRU, an intrusion detection model based on the Sequential Attention mechanism. In each iteration, the attention mechanism assigns weights to features and selects the highest-weight feature, building a subset that filters out irrelevant noise and reduces dimensionality. BiGRU is subsequently used to extract temporal dependencies from the selected features. Finally, a multi-class classifier determines the specific attack type.

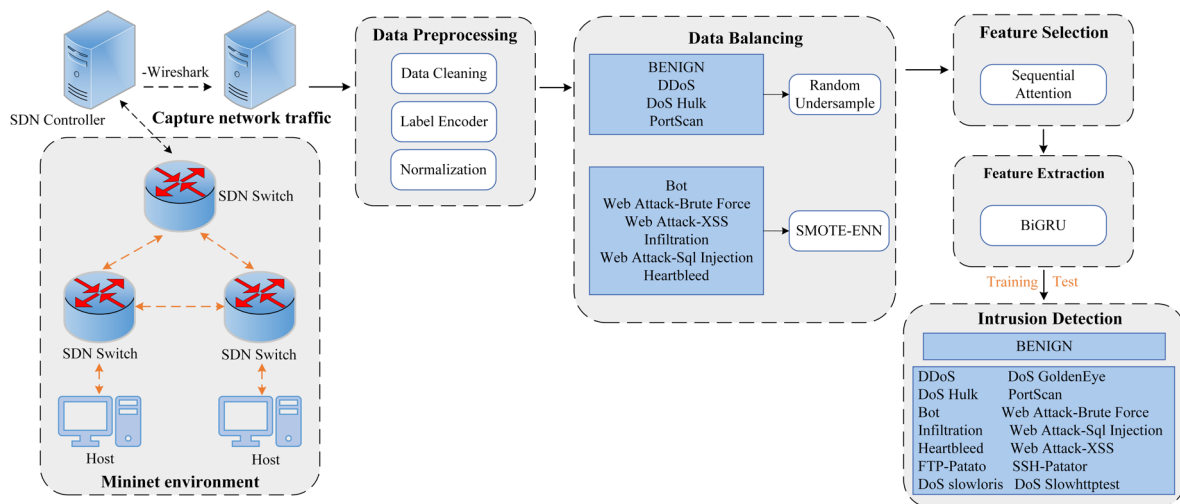


Fig. 1. The workflow diagram of multi-class intrusion detection system in SDN

### 3.1 Data Preprocessing

**Data Cleaning.** The original dataset contained values unsuitable for model input, including missing or infinite entries. Our preprocessing module addresses these issues by performing data cleansing, duplicate removal, and transformation. Invalid or inconsistent values are replaced with appropriate constants. We drop zero variance columns—such as Bwd PSH Flags, Fwd Avg Bytes/Bulk, and eight others—as they provide no discriminative information. Similarly, for groups of columns with identical values (e.g., Total Fwd Packets vs. Subflow Fwd Packets, Fwd URG Flags vs. CWE Flag Count), we retain only one representative feature per group and remove the rest.

**Label Encoding.** Categorical labels in the dataset, such as Benign, DDoS, and DoS Hulk, must be converted to integer values before model training because most machine-learning algorithms cannot process non-numeric labels directly. We use label encoding instead of one-hot encoding to avoid the additional memory overhead of one-hot vectors. During preprocessing, each class label is mapped to an integer from 0 to 14, as shown in Table 1.

Table 1. Label coding and types

0	Benign	8	Heartbleed
1	Bot	9	Injection
2	DDoS	10	PortScan
3	DoS GoldenEye	11	SSH-Patator
4	DoS Hulk	12	Web Attack-Brute Force
5	DoS Slowhttptest	13	Infiltration
6	DoS slowloris	14	Web Attack-Sql
7	FTP-Patator		

**Normalisation.** In the original dataset, the attributes are scaled differently, leading to significant variations in the size of their values. Normalization, a commonly used data preprocessing method in machine learning and deep learning, is employed to scale all attributes in the dataset to the range [0,1]. This helps reduce redundancy and improve the model’s training efficiency. In this paper, Min-Max normalization is used to scale the data features, there is

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (1)$$

Where:  $x_{norm}$  is the normalised value, in the interval [0,1],  $x$  is the original value to be normalised,  $x_{min}$  is the minimum eigenvalue and  $x_{max}$  is the maximum eigenvalue.

### 3.2 Sampling Mechanism

The dataset is severely imbalanced—there are over two million benign samples but only eleven Heartbleed instances. As a result, the model struggles to detect rare attack traffic, even though these minority-class attacks pose a disproportionately high risk to network security.

In this paper, we employ random undersampling to mitigate dataset imbalance by removing excess instances from the majority-class—Benign, DoS Hulk, PortScan, and DDoS.

For the minority-class samples such as Bot, Web Attack-Brute Force, Web Attack-XSS, Infiltration, Web Attack-SQL Injection, Heartbleed, and others, the SMOTE algorithm is applied for oversampling. The SMOTE algorithm operates by synthetically generating new samples from the minority-class, which helps avoid the overfitting problem caused by simply duplicating existing samples. However, SMOTE can introduce noisy samples that fall into majority-class regions and harm classification performance. To address this, we integrate Edited Nearest Neighbors (ENN) with SMOTE. After SMOTE generates new minority-class samples, ENN removes any instances misclassified by their nearest majority neighbors. This SMOTE-ENN combination reduces both overfitting and noise, thereby improving overall classifier performance.

This paper proposes a hybrid sampling method that combines random undersampling and SMOTE-ENN oversampling techniques. Data balancing is achieved by reducing the number of majority-class samples while increasing the number of minority-class samples.

### 3.3 Dynamic Feature Selection Algorithm

Feature selection is a crucial step in intrusion detection, primarily due to the high-dimensional nature of network data [22]. High-dimensional data can lead to computational inefficiencies and overfitting during model training. It may also contain irrelevant or noisy features, which can negatively impact the accuracy and reliability of machine learning models. Feature selection methods generally fall into three categories: filter, wrapper, and embedded [23]. These methods aim to select a subset of the original features that are most relevant to the learning task and boost model performance, while eliminating irrelevant, redundant, or noisy features. This process simplifies the model, improves its generalization ability, and reduces computational complexity.

Dimensionality reduction methods such as Principal Component Analysis (PCA) may not effectively preserve the original feature information [24]. To eliminate feature redundancy in the CICIDS2017 dataset, we apply a dynamic feature-selection algorithm called Sequential Attention. This method uses a greedy forward strategy and leverages attention weights to gauge feature importance. Rather than selecting the entire subset at once, Sequential Attention iteratively evaluates each feature's marginal contribution, enabling it to identify and retain only those features that significantly enhance model performance.

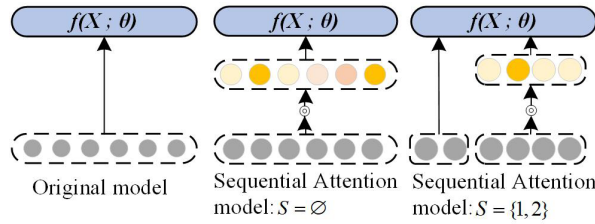


Fig. 2. Sequential attention mechanism

As shown in Fig. 2, for a given dataset after preprocessing  $X \in \mathbb{R}^{n \times d}$ , let the data be represented as a matrix with  $n$  rows corresponding to data streams and  $d$  columns representing features. The algorithm is based on a greedy forward selection approach, where  $k$  features are selected iteratively. Let the neural network model be denoted as  $f(X; \theta)$ , where  $\theta$  is the set of model parameters. Let  $y \in \mathbb{R}^n$  be the set of labels, and  $\ell(f(X; \theta), y)$  represent the loss between the predicted values  $\hat{y}$  and the actual labels  $y$ . Initialize an empty set  $S = \emptyset$  to store the indices of the selected features. Repeat the following steps for  $k$  iterations, selecting one feature at a time:

- (1) Calculate feature weights

The optimal set of parameters  $(\theta^*, w^*)$  is selected by minimising the loss function  $\ell(f(X \odot W; \theta), y)$ , calculated as equation (2):

$$f((X \odot W; \theta), y) = \|f(X; \theta) - y\|_2^2 = \|X\theta - y\|_2^2. \quad (2)$$

Where: the symbol  $\odot$  represents the Hadamard product. The weight matrix for feature selection is represented as  $W = 1_n \text{softmax}(w, \bar{S})^T$ , which is used to weight the input features. The importance of the features is controlled by combining the result of  $\text{softmax}_i(w, \bar{S})$ .  $1_n$  represents a full 1 vector of length  $n$  as shown in equation (3):

$$\text{softmax}_i(w, \bar{S}) := \begin{cases} 1, & i \in S; \\ \frac{\exp(w_i)}{\sum_{j \in \bar{S}} \exp(w_j)}, & i \in \bar{S} := [d] \setminus S. \end{cases} \quad (3)$$

(2) Selection of features

Selection of features  $i^*$  with maximum attentional weight:

$$i^* = \arg \max_{i \in S} \text{softmax}_i(w, S). \quad (4)$$

$i^*$  will be added to the set of selected features:  $S = S \cup \{i^*\}$ .

(3) Update the model

The model  $f(X; \theta)$  is trained using the currently selected feature set  $S$  and the attention weights  $w$  are updated. After  $k$  iterations of selection, the Sequential Attention algorithm outputs a feature subset  $S$ , which contains  $k$  feature indices.

### 3.4 Feature Extraction

RNN is an extension of traditional feedforward neural networks and are primarily used to analyze time series data, enabling the learning of long-term temporal features. However, when training RNN using backpropagation, the problem of gradient vanishing can occur [25]. To address this issue, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were proposed. BiGRU, based on GRU, has fewer gates and parameters compared to LSTM, making it more computationally efficient. Additionally, BiGRU can process input data both forward and backward. Given these advantages, this paper selects the BiGRU algorithm to focus on learning the filtered features and extracting time-series features from the samples.

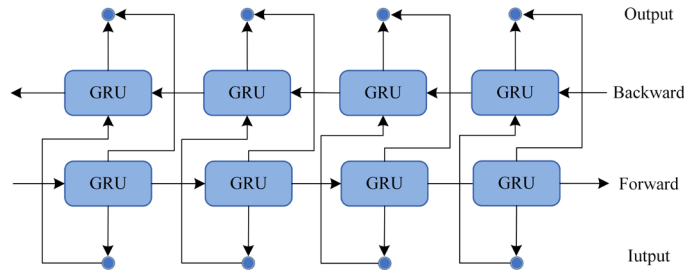


Fig. 3. BiGRU modelling framework

BiGRU consists of two GRUs operating in different directions, as shown in Fig. 3. One GRU captures information from the previous time step in the forward direction, while the other captures information from the future time step in the reverse direction. This dual-direction processing enables a more accurate judgment. The outputs of the two GRU, working in opposite directions, together determine the output at the current time step.

The GRU is primarily composed of an update gate and a reset gate. The update gate controls how past information is written into the current state, while the reset gate controls how previous state information is written into the current candidate set. At the current time step, denoted as  $t$ , the GRU's calculations are given by the following formulas (5-8):

$$r_t = \sigma(\omega_r x_t + U_r h_{t-1} + b_r). \quad (5)$$

$$z_t = \sigma(\omega_z x_t + U_z h_{t-1} + b_z), t \in [1, m]. \quad (6)$$

$$\tilde{h}_t = \tanh[\omega_h x_t + U(r_t \times h_{t-1}) + b_h]. \quad (7)$$

$$h_t = z_t \times h_{t-1} + (1 - z_t) \times \tilde{h}_t. \quad (8)$$

Where,  $r_t$  is the reset gate,  $z_t$  is the update gate,  $\tilde{h}_t$  is the new candidate state, and  $h_t$  is the updated hidden state.  $\sigma$  is the sigmoid function.  $\omega_r, \omega_z, \omega_h$  are the input weight matrices,  $b_r, b_z, b_h$  are the bias terms, and  $U_r, U_z, U$  are the state weight matrices.

The formulae (9-11) for BiGRU are as follows and the final output is expressed as  $H_t = [h_1, h_2, \dots, h_t]$ :

$$\vec{h}_t = \text{GRU}(h_t, \vec{h}_{t-1}). \quad (9)$$

$$\overleftarrow{h}_t = \text{GRU}(h_t, \overleftarrow{h}_{t-1}). \quad (10)$$

$$h_t = [\vec{h}_t, \overleftarrow{h}_t]. \quad (11)$$

where,  $\vec{h}_t, \overleftarrow{h}_t$  are the hidden states of the forward and reverse GRU.

### 3.5 Output Layer

Since intrusion detection is a multi-classification problem, the output layer is responsible for outputting the feature vector  $x$  containing traffic information obtained through the BiGRU layer to the fully-connected layer for traffic classification, and the predicted output is normalised using the softmax function to obtain the predicted classified traffic labels. Calculation of equation (12):

$$p = \text{softmax}(W_i x + b_i). \quad (12)$$

where  $W_i$  is the matrix of weight coefficients output from the fully connected layer,  $b_i$  is the bias term, and  $p$  is the output predicted traffic label.

## 4 Experiment and Analysis

### 4.1 Experimental Environment and Dataset

The experiments in this paper are based on the Tensorflow2.0 framework to build deep learning hybrid models, the hardware environment is Intel Core i5-10210U, GeForce MX350 Laptop GPU, and the software environment is a Linux virtual machine with Ubuntu 18.04.

In this paper, we use the public dataset CICIDS2017 published by Canadian Institute for Cybersecurity (CIC), the experiment lasted for five days and collected more than seventy traffic including average message length, maximum message length, destination port, number of packets sent per second, type of transport protocol, and other traffic characteristics.

As can be seen from Table 2, due to the great disparity in the number of samples of each class in the CICIDS2017 dataset, for example, Benign has more than 2 million samples and Heartbleed has only 11, which leads to the data imbalance problem. This data imbalance problem causes the model to overlearn the features of the majority-class while ignoring the features of the minority-class, which reduces the accuracy of the model's intrusion detection.

**Table 2.** Sample size of the dataset

Category	Label	Sample size	Dataset 1	Dataset 2
Benign	0	2273097	431391	48132
DoS Hulk	4	231073	172846	29510
PortScan	10	158930	128014	29650
DDoS	2	128027	90694	19649
DoS GoldenEye	3	10293	10286	10183
FTP-Patator	7	7938	5931	5835
SSH-Patator	11	5897	5385	2991
DoS slowloris	6	5796	5228	5115
DoS Slowhttptest	5	5499	3219	5057
Bot	1	1966	1948	4724
Web Attack-Brute Force	12	1507	1470	2678
Web Attack-XSS	14	652	652	2546
Infiltration	9	36	36	4919
Web Attack-Sql Injection	13	21	21	4731
Heartbleed	8	11	11	5000

To address this issue, random undersampling is applied to the majority-class samples, specifically reducing the number of samples in the four classes of Benign, DoS Hulk, PortScan, and DDoS, as shown in dataset 1 in Table 2. Furthermore, dataset 2 is created by combining random undersampling with SMOTE-ENN oversampling techniques, in order to investigate the detection capability of the SA-BiGRU model for minority-class attacks.

## 4.2 Evaluation Criteria

In this paper, five evaluation metrics—Accuracy (Acc), Precision (P), Recall (R), F1-score (F1), and Confusion Matrix—are used to assess the model’s performance, with calculations provided in equation (13-16). Accuracy, precision, and recall primarily measure the model’s ability to predict normal and attack traffic. The F1-score offers a comprehensive metric that balances precision and recall. The confusion matrix shows the detailed classification results of the model predictions, as shown in Table 3.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}. \quad (13)$$

$$P = \frac{TP}{TP + FP}. \quad (14)$$

$$R = \frac{TP}{TP + FN}. \quad (15)$$

$$F1 = \frac{2 \times P \times R}{P + R}. \quad (16)$$

**Table 3.** Parameter definitions

Actual value	Predicted value	
	Attack flow	Normal flow
Attack flow	TP	FN
Normal flow	FP	TN

## 4.3 Evaluation Results

The SA-BiGRU model first applies sequential attention to reduce feature dimensionality, filtering out irrelevant noise before feeding the data into a BiGRU network that extracts long-range temporal dependencies. To mitigate

class imbalance, we combine random undersampling of majority-class with SMOTE-ENN oversampling of minority-class, producing a more balanced dataset and improving detection rates for rare attack types.

**Impact of Data Balance on Test Results.** To investigate the impact of the mixed sampling approach on the detection performance of minority-class attacks, Table 4 presents experiments conducted on dataset 1 (undersampling only) and dataset 2 (undersampling and SMOTE-ENN) for a fixed number of features,  $k = 17$ . As shown in the experimental results for dataset 1 in Table 4, most DoS attacks (e.g., DoS-Hulk, DoS GoldenEye) as well as SSH-Patator have relatively large sample sizes, and their Precision, Recall, and F1 scores remain high, indicating that the model exhibits strong discriminative ability in detecting these attacks. However, for some attack categories with fewer total samples, such as Bot, Web-Attack-XSS, and Infiltration, the metrics are significantly lower or even zero. Therefore, to improve the detection ability for these minority-class attacks, it is necessary to combine undersampling with oversampling techniques to generate more trainable samples for these underrepresented attacks.

The results in Table 4 on dataset 2 show that, after applying the oversampling technique to balance the data, the model's overall multi-category attack detection performance is markedly better than on dataset 1. This improvement is particularly noticeable for attacks such as Bot, Web Attack-XSS, Infiltration, Web Attack-SQL Injection, and Heartbleed. Our results show that SMOTE-ENN effectively addresses class imbalance on the CICIDS2017 dataset. Its precision for detecting Botnet, Web Attack-Brute Force, Infiltration, and Web Attack-SQL Injection attacks exceed that reported in a recent study [26] employing a gated self-attention mechanism with BiGRU.

**Table 4.** Results of classification experiments on datasets 1/2

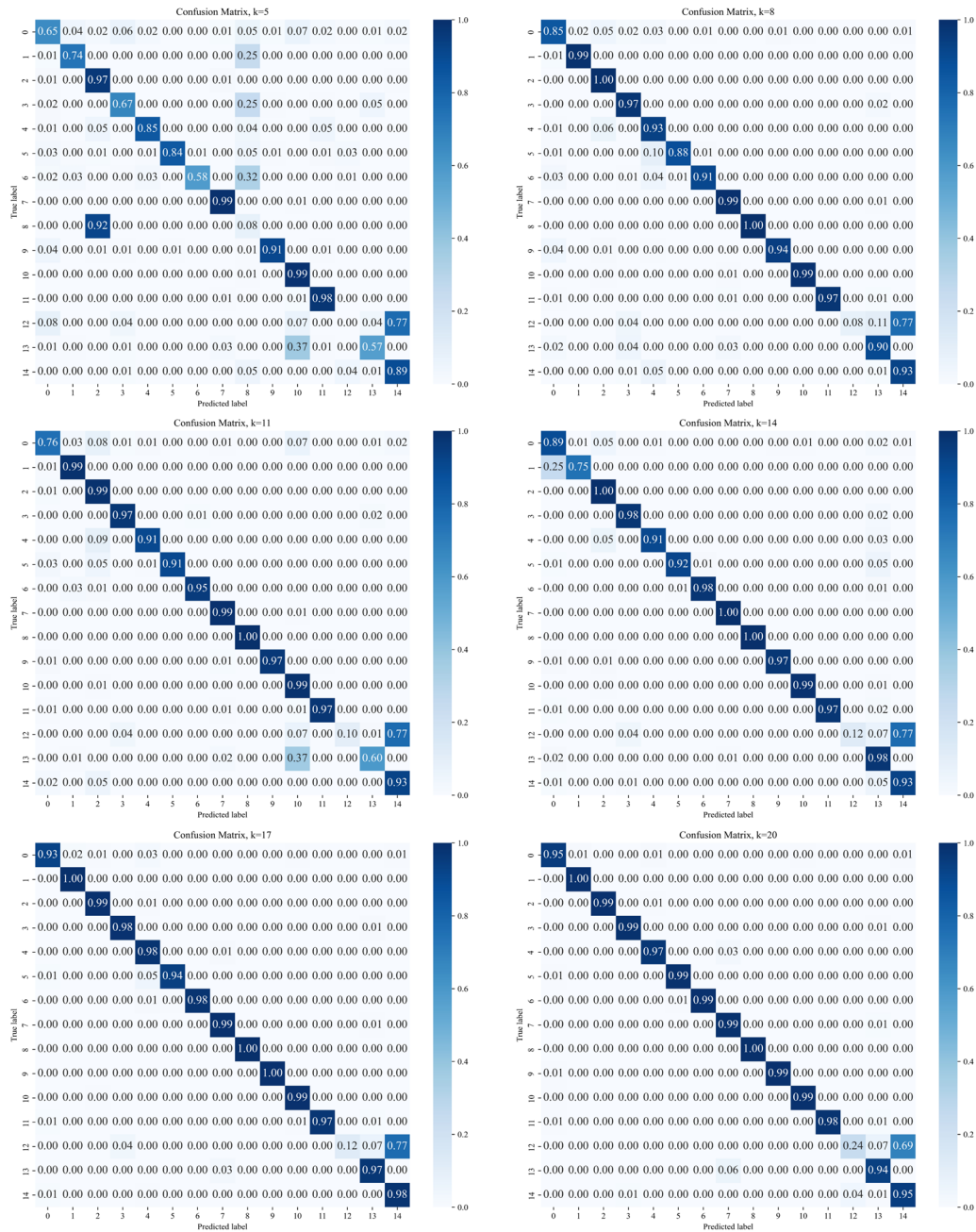
Category	Dataset 1			Dataset 2		
	P	R	F1	P	R	F1
Benign	0.78	0.97	0.87	0.99	0.92	0.95
DoS Hulk	0.96	0.93	0.94	0.95	0.98	0.96
PortScan	0.95	0.94	0.94	0.99	0.99	0.99
DDoS	0.93	0.96	0.94	0.98	0.99	0.99
DoS GoldenEye	0.96	0.86	0.91	0.98	0.98	0.98
FTP-Patator	0.98	0.65	0.79	0.92	0.99	0.95
SSH-Patator	0.98	0.91	0.94	0.99	0.97	0.98
DoS slowloris	0.91	0.54	0.68	0.98	0.98	0.98
DoS Slowhttptest	0.93	0.83	0.87	0.98	0.94	0.96
Bot	0.00	0.00	0.00	0.93	0.99	0.96
Web Attack-Brute Force	0.71	0.05	0.09	0.94	0.11	0.21
Web Attack-XSS	0.00	0.00	0.00	0.73	0.98	0.83
Infiltration	0.00	0.00	0.00	0.98	0.99	0.99
Web Attack-Sql Injection	0.00	0.00	0.00	0.92	0.97	0.94
Heartbleed	0.00	0.00	0.00	1.00	1.00	1.00

**Impact of Feature Selection on Detection Results.** To evaluate feature selection's impact on classification performance, classification metrics were analyzed across feature subsets ( $k = 5, 8, 11, 14, 17, 20$ ) using dataset 2, with confusion matrices plotted (Fig. 4). It was observed that model performance improved progressively as  $k$  increased, with higher feature counts enabling the capture of more discriminative information. At  $k = 5$ , suboptimal recall rates were recorded for multiple attack categories, particularly minority-class exhibiting significant misclassification. Optimal detection accuracy (0.99 recall for most categories) was achieved at  $k = 17$ . However, further expansion to  $k = 20$  yielded no significant performance improvement, while misclassification rates increased marginally in specific classes, suggesting feature redundancy-induced interference.

Notably, Web Attack-Brute Force attacks demonstrated persistently low recall rates. This phenomenon is attributed to their characteristic repetitive request patterns, which exhibit high similarity to benign traffic anomalies. Consequently, these attacks are frequently misclassified as noise during training, resulting in inadequate feature learning and reduced recognition capability.

Considering detection accuracy, computational resource consumption, and model stability, this paper selects  $k = 17$  as the optimal number of features for the final model in the multi-classification research. Therefore, the experiments investigating the effect of data balance on detection results are all conducted with  $k = 17$  to ensure that the experimental outcomes accurately reflect the optimal impact of data balance on classification performance. Fig. 5 presents the 17 features selected by the SA-BiGRU model on dataset 2. These features capture distinctions between various attack types and normal traffic across multiple dimensions—port identification, packet-

length distribution, inter-arrival timing, flow duration, and packet rate. The destination port filters out services under concentrated attack; packet-length statistics reveal small-packet or large-packet flooding behaviors; flow duration and active time capture traffic burst patterns; and forward/backward inter-arrival times and packet rates quantitatively characterize anomalies in sender–receiver interaction rhythms. Together, these indicators distinguish DDoS, PortScan, Botnet, and other attack modes, while also helping to eliminate redundant noise and enhance both the accuracy and generalization of multi-class intrusion detection.



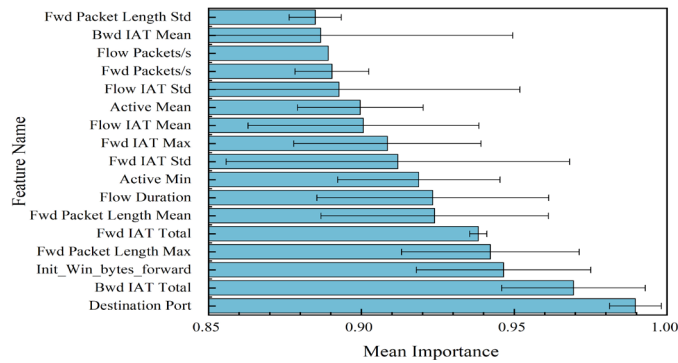


Fig. 5. Importance of the 17 features selected from dataset 2

**Ablation Experiment of the Proposed Model.** To evaluate the effectiveness of each model component, the BiGRU served as the baseline. The model combining random undersampling and the Sequential Attention mechanism with BiGRU was designated as RSB. Expanding this framework, the integration of the SMOTE produced the RSSB. Further enhancement via the ENN method yielded the final model, termed the RSESB. An ablation study on the CICIDS2017 dataset analyzed the incremental contributions of these components.

As shown in Fig. 6, the BiGRU model, while capable of capturing temporal features through its bidirectional architecture, exhibited limitations in detecting certain types of attacks, resulting in a lower accuracy of 0.88. To address these limitations, the RSB model, which incorporates random undersampling and feature selection, achieved an improved accuracy of 0.92. However, with only undersampling, the model struggles to sufficiently learn from the minority-class due to the small number of minority-class samples. After incorporating SMOTE oversampling, the detection accuracy of the RSSB model rises to 0.94, indicating that generating new minority-class samples through SMOTE oversampling enhances the model’s ability to detect minority-class attacks. Building upon this improvement, this paper further proposes the RSESB model, which incorporates SMOTE-ENN oversampling. Unlike RSSB, the RSESB model not only generates additional minority-class samples but also effectively removes noisy samples, thus elevating the detection accuracy to 0.98. Compared to other models, the proposed RSESB achieves superior performance in terms of Accuracy, Precision, Recall, and F1-score metrics.

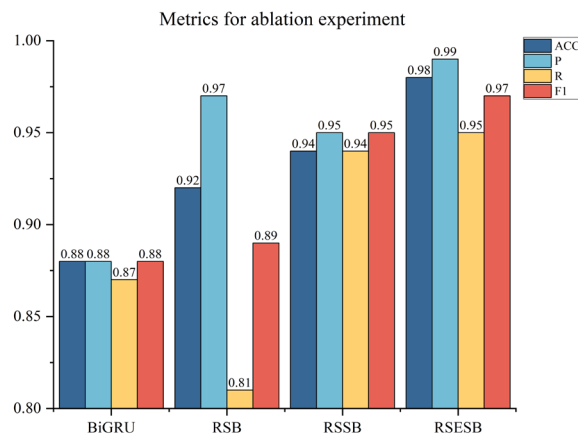


Fig. 6. Ablation experiment on the CICIDS2017

**Comparison of Different Models on the CICIDS2017 Dataset.** Based on the ablation experiments, a further comparison was performed between the proposed SA-BiGRU model and several common machine learning models, including RF, Multi-Layer Perceptron (MLP), KNN, GRU, and LSTM, with the results presented in Table 5. Since web attack data often exhibit imbalanced class distributions, the primary focus of this section is placed on evaluating the detection effectiveness of each model for minority-class attacks, which mainly include Bot, Web Attack-Brute Force, Web Attack-XSS, Infiltration, Web Attack-Sql Injection, and Heartbleed.

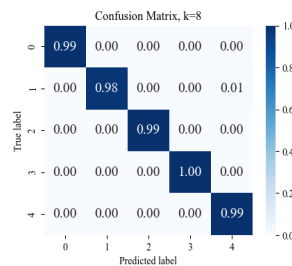
As indicated in Table 5, more pronounced differences are observed among the models in detecting these minority-class attacks. The detection accuracy of SA-BiGRU on Web Attack-Brute Force, Web Attack-XSS, Web Attack-Sql Injection, and Infiltration is notably superior to that of other methods, suggesting that the introduction of the attention mechanism facilitates more precise extraction of key features, while the hybrid sampling technique effectively alleviates the class imbalance, thereby improving the detection accuracy of these minority-class. In summary, SA-BiGRU achieves higher detection accuracy in identifying network attacks, particularly for minority-class attacks, and demonstrates stronger adaptability to datasets with class imbalance in practical scenarios.

**Table 5.** Precision of experimental results of different models on the CICIDS2017

Category	RF	MLP	KNN	GRU	LSTM	SA-BiGRU
Benign	0.95	0.99	0.97	0.98	0.99	0.99
DoS Hulk	0.97	0.85	0.99	0.94	0.95	0.95
PortScan	0.97	0.98	0.96	0.99	0.93	0.99
DDoS	0.98	0.98	0.99	0.98	0.97	0.98
DoS GoldenEye	0.97	0.90	0.98	0.94	0.98	0.98
FTP-Patator	0.97	0.90	0.99	0.89	0.91	0.92
SSH-Patator	0.94	0.96	0.94	0.99	0.96	0.99
DoS slowloris	0.88	0.97	0.96	0.96	0.96	0.98
DoS Slowhttptest	0.89	0.89	0.97	0.85	0.97	0.98
Bot	0.99	0.90	0.80	0.90	0.86	0.93
Web Attack-Brute Force	0.00	0.75	0.68	0.72	0.68	0.94
Web Attack-XSS	0.00	0.28	0.42	0.35	0.52	0.73
Infiltration	0.74	0.66	0.00	0.67	0.61	0.98
Web Attack-Sql Injection	0.00	0.49	0.00	0.76	0.00	0.92
Heartbleed	0.00	1.00	0.25	1.00	1.00	1.00

**Performance of the Model on Other Datasets.** To evaluate the generalization of our proposed SA-BiGRU model and hybrid sampling strategy, we conducted experiments on the NSL-KDD and CICDDOS2019 datasets. As shown in Table 6 and Fig. 7, combining random undersampling with SMOTE-ENN effectively balanced the five major classes (DoS, Probe, R2L, U2R, and Normal) in NSL-KDD. The model achieved at least 0.98 on all three evaluation metrics for each class, with the rarest class (U2R) attaining a perfect recall of 1.00—demonstrating the model’s sensitivity to scarce samples. Further, Table 7 and Fig. 8 report detection performance on thirteen DDoS attack types in CICDDOS2019. Even for small-sample attacks—WebDDoS with just 51 instances, Syn with 533, and DrDoS\_NetBIOS with 598—SA-BiGRU achieved F1 scores of 0.96, 0.94, and 0.95, respectively. These results confirm the robustness and adaptability of our approach across diverse attack patterns and extreme class imbalance.

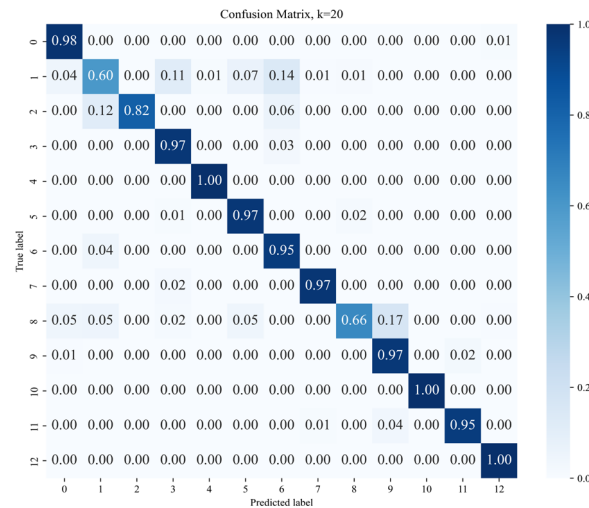
In summary, the SA-BiGRU model and its data-balancing scheme not only significantly improve anomaly detection accuracy but also maintain high performance and stability in cross-dataset evaluations, offering a reliable and efficient solution for intrusion detection in large-scale, complex network environments.



**Fig. 7.** Confusion matrix on the NSL-KDD

**Table 6.** Experimental results of the model on the NSL-KDD

Category	Raw data	After mixed sampling	P	R	F1
DoS	46116	19990	0.99	0.99	0.99
Probe	2662	5976	0.99	0.98	0.99
R2L	1437	4988	0.99	0.99	0.99
U2R	110	5000	0.99	1.00	0.99
Normal	64351	19934	0.99	0.99	0.99

**Fig. 8.** Confusion matrix on the CICDDOS2019**Table 7.** Experimental results of the model on the CICDDOS2019

Category	Raw data	After mixed sampling	P	R	F1
Benign	55824	19353	0.98	0.98	0.98
DrDoS_DNS	3669	2205	0.76	0.60	0.67
DrDoS_LDAP	1440	1320	0.66	0.82	0.73
DrDoS_MSSQL	6212	3648	0.98	0.96	0.97
DrDoS_NTP	121368	18319	0.99	0.99	0.99
DrDoS_NetBIOS	598	1609	0.93	0.97	0.95
DrDoS_SNMP	2717	2575	0.92	0.95	0.93
DrDoS_UDP	10420	6299	0.99	0.98	0.98
Portmap	685	2455	0.82	0.66	0.73
Syn	533	3916	0.88	0.99	0.94
TFTP	98917	19750	1.00	1.00	1.00
UDP-lag	8872	5734	0.99	0.96	0.98
WebDDoS	51	4870	0.93	0.99	0.96

## 5 Conclusion

A SA-BiGRU model based on the Sequential Attention mechanism is proposed in this paper for intrusion detection in SDNs, where the issue of imbalanced data distribution is addressed through a hybrid approach that combines random undersampling with SMOTE-ENN oversampling (RSE). Superior performance is demonstrated by the RSE algorithm compared to common sampling methods, particularly in enhancing the detection accuracy for minority-class attacks. Detection performance is further improved through effective feature selection, noise filtering, key feature extraction, and temporal feature learning using BiGRU. To validate the proposed model's performance, experiments on the CICIDS2017 dataset show our approach achieves 0.98 accuracy and 0.99 precision in multi-class intrusion detection, surpassing comparable hybrid models. Furthermore, on the NSL-KDD dataset, SA-BiGRU achieves no less than 0.98 for all three evaluation metrics across every category, with the minority-class U2R (having the fewest samples) attaining 1.0 recall. When evaluating 13 attack types in

CICDDoS2019, SA-BiGRU maintains robust F1-scores of 0.96 for WebDDoS (51 samples), 0.94 for Syn (533 samples), and 0.95 for DrDoS\_NetBIOS (598 samples), demonstrating exceptional adaptability in complex networks. However, our evaluation is limited to these two public datasets, and generalization in real-world deployments remains to be verified. In future work, we will conduct online evaluations on additional real-traffic datasets to assess robustness, develop a continual learning framework for online adaptation to new attacks, and enhance transparency and trustworthiness in SDN security detection.

## References

- [1] M.A. Aladaileh, M. Anbar, I.H. Hasbullah, Y.W. Chong, Y.K. Sanjalawe, Detection techniques of distributed denial of service attacks on software-defined networking controller—A review, *IEEE Access* 8(2020) 143985-143995. <https://doi.org/10.1109/ACCESS.2020.3013998>
- [2] Z. Ahmad, A.S. Khan, C. W. Shiang, J. Abdullah, F. Ahmad, Network intrusion detection system: a systematic study of machine learning and deep learning approaches, *Transactions on Emerging Telecommunications Technologies* 32(1) (2021) e4150. <https://doi.org/10.1002/ett.4150>
- [3] R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system, *IEEE Access* 7(2019) 41525-41550. <https://doi.org/10.1109/ACCESS.2019.2895334>
- [4] I.H. Sarker, Y.B. Abushark, F. Alsolami, A.I. Khan, Intrudtree: a machine learning based cyber security intrusion detection model, *Symmetry* 12(5)(2020) 754. <https://doi.org/10.3390/sym12050754>
- [5] K. Singh, B. Kumar, S. Kumar, V.P. Singh, A. Singh, Mitigation of cyber attacks in SDN-based IoT systems using machine learning techniques, *International Journal of Intelligent Systems and Applications in Engineering* 12(8s)(2024) 482-492. <https://ijisae.org/index.php/IJISAE/article/view/4149>
- [6] M.A. Setitra, M. Fan, B.L.Y. Agbley, Z.E.A. Bensalem, Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment, *Network* 3(4)(2023) 538-562. <https://doi.org/10.3390/network3040024>
- [7] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, D. Siracusa, LUCID: a practical, lightweight deep learning solution for DDoS attack detection, *IEEE Transactions on Network and Service Management* 17(2)(2020) 876-889. <https://doi.org/10.1109/TNSM.2020.2971776>
- [8] J. Zhao, Y. Liu, Q. Zhang, X. Zheng, CNN-AttBiLSTM mechanism: a DDoS attack detection method based on attention mechanism and CNN-BiLSTM, *IEEE Access* 11(2023) 136308-136317. <https://doi.org/10.1109/ACCESS.2023.3334916>
- [9] M.S. Elsayed, N.A. Le-Khac, A.D. Jurcut, A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique, *Journal of Network and Computer Applications* 191(2021) 103160. <https://doi.org/10.1016/j.jnca.2021.103160>
- [10] G.S. Vidhya, R. Nagarajan, A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network, *Computing* 106(8)(2024) 2613-2642. <https://doi.org/10.1007/s00607-024-01295-w>
- [11] F.F. Kamel, M.S. Mahdi, Intrusion detection systems based on RNN and GRU models using CSE-CIC-IDS2018 dataset in AWS cloud, *Journal of Al-Qadisiyah for Computer Science and Mathematics* 16(4)(2024) 141-160. <https://doi.org/10.29304/jqscsm.2024.16.41780>
- [12] M.A. Razib, D. Javeed, M.T. Khan, R. Alkanhel, M.S.A. Muthanna, Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework, *IEEE Access* 10(2022) 53015-53026. <https://doi.org/10.1109/ACCESS.2022.3172304>
- [13] V. Hnamte, J. Hussain, DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system, *Telematics and Informatics Reports* 10(2023) 100053. <https://doi.org/10.1016/j.teler.2023.100053>
- [14] R.B. Said, Z. Sabir, I. Askerzade, CNN-BiLSTM: a hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection, *IEEE Access* 11(2023) 138732-138747. <https://doi.org/10.1109/ACCESS.2023.3340142>
- [15] S.K. Sahu, D.P. Mohapatra, J.K. Rout, K.S. Sahoo, Q.V. Pham, N.N. Dao, A LSTM-FCNN based multi-class intrusion detection using scalable framework, *Computers & Electrical Engineering* 99(2022) 107720. <https://doi.org/10.1016/j.compeleceng.2022.107720>

- [16] J. Han, W. Pak, Hierarchical LSTM-based network intrusion detection system using hybrid classification, *Applied Sciences* 13(5)(2023) 3089.  
<https://doi.org/10.3390/app13053089>
- [17] M. Cui, J. Chen, X. Qiu, W. Lv, H. Qin, X. Zhang, Multi-class intrusion detection system in SDN based on hybrid BiLSTM model, *Cluster Computing* 27(7)(2024) 9937-9956.  
<https://doi.org/10.1007/s10586-024-04477-5>
- [18] Z. Wang, F.A. Ghaleb, An attention-based convolutional neural network for intrusion detection model, *IEEE Access* 11(2023) 43116-43127.  
<https://doi.org/10.1109/ACCESS.2023.3271408>
- [19] K.R.M. Fernando, C.P. Tsokos, Dynamically Weighted Balanced Loss: Class Imbalanced Learning and Confidence Calibration of Deep Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems* 33(7)(2022) 2940-2951.  
<https://doi.org/10.1109/TNNLS.2020.3047335>
- [20] M.S. Alshelhri, O. Saidani, F.S. Alrayes, S.F. Abbasi, J. Ahmad, A self-attention-based deep convolutional neural networks for IIoT networks intrusion detection, *IEEE Access* 12(2024) 45762-45772.  
<https://doi.org/10.1109/ACCESS.2024.3380816>
- [21] R. Harini, N. Maheswari, S. Ganapathy, M. Sivagami, An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach, *Alexandria Engineering Journal* 78(2023) 469-482.  
<https://doi.org/10.1016/j.aej.2023.07.063>
- [22] M. Bateni, L. Chen, M. Fahrback, G. Fu, V. Mirrokni, T. Yasuda, Sequential attention for feature selection. <<https://arxiv.org/abs/2209.14881v1>>, 2022 (accessed 09.11.24).
- [23] B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection, *Computers & Security* 81(2019) 148-155.  
<https://doi.org/10.1016/j.cose.2018.11.005>
- [24] R. Abdulhammed, H. Musafar, A. Alessa, M. Faezipour, A. Abuzneid, Features dimensionality reduction approaches for machine learning based network intrusion detection, *Electronics* 8(3)(2019) 322.  
<https://doi.org/10.3390/electronics8030322>
- [25] T.A. Tang, D. McLoren, L. Mhamdi, S.A.R. Zaidi, M. Ghogho, Intrusion detection in sdn-based networks: deep recurrent neural network approach, *Deep Learning Applications for Cyber Security* (2019) 175-195.  
[https://doi.org/10.1007/978-3-030-13057-2\\_8](https://doi.org/10.1007/978-3-030-13057-2_8)
- [26] Z. Hu, G. Liu, Y. Li, S. Zhuang, SAGB: self-attention with gate and BiGRU network for intrusion detection, *Complex & Intelligent Systems* 10(6)(2024) 8467-8479.  
<https://doi.org/10.1007/s40747-024-01577-y>