BiGRU-DA-BERT Encrypted Traffic Classification Technology Based on BERT

Haobo Liang*, Yingxiong Leng, Jinman Luo, Jie Chen, Xiaoji Guo
Dongguan Power Supply Bureau, Guangdong Power Grid Corporation, Dongguan, Guangdong, China

Abstract: Compared with traditional models and methods, pre-trained models have been gradually applied to the field of network encrypted traffic classification in recent years due to automatic feature learning, strong generalization ability and dynamic adaptability. The weight of minority categories of data is increased, and the Focal Loss function is introduced to further increase the focus on minority categories. Experimental results show that the improved model improves the overall classification accuracy by 2.3% and the average F1 value of minority classes by 15.3% on the dataset, which proves the effectiveness of model architecture optimization and adjustment model classification, and also provides a new way to improve the classification accuracy of minority categories in unbalanced datasets.

Keywords: Encrypted Traffic Classification; BERT; BiGRU; Differential Attention Mechanism

1. Introduction

In many fields based on deep learning as a research method, most of them have the problem of unbalanced dataset categories, especially in the field of network encrypted traffic [1-2]. The BERT cascade BiGRU model has poor classification effect on the category imbalance dataset, and the related reasons are summarized and reasoned [3-4].

From the data level, first of all, the number of samples during training is huge, and the significant feature of the category imbalance dataset is that the number of samples in the minority class is much smaller than that in the majority. Specifically, during the training process, the majority of samples dominate feature learning due to the numerical advantage, resulting in the model tending to misjudge the sample as the majority class, while the unique features of the minority class are not fully captured due to insufficient sample size [5-7]. On the other hand, insufficient sample size of minority classes can easily lead to the risk of overfitting, and the model may over-remember the noise characteristics of minority samples rather than the intrinsic pattern. Because there are limited minority samples available for learning, the model may remember some of the noise characteristics of these samples instead of the true intrinsic features, resulting in poor performance in the face of new minority samples[8]. Then there is the fact that the feature distribution of the majority class and the minority class can be very different. In the dataset, the features of a few classes may be relatively scattered, lacking sufficient sample support to form stable feature patterns [9-12].

From the model level, the BERT cascade BiGRU model itself may be more suitable for processing balanced datasets in terms of model structure characteristics [13-14]. First, BERT is a pre-trained model based on large-scale unsupervised learning, and its original design is not specifically designed to target the problem of class imbalance, secondly, although BiGRU can capture the temporal features of the sequence, it will also be affected by the majority of class samples in the case of class imbalance, and it is more inclined to learn the temporal pattern of most classes The global normalization of the group focuses too much on the majority sample that occurs frequently, resulting in the marginalization of the minority class. From the perspective of loss function, traditional cross-entropy loss treats all samples equally, and does not take into account the difference in the number of samples of different categories. The weighted cross-entropy loss function can alleviate this problem to a certain extent by giving higher weight to a few class samples, forcing the model to pay more attention to the classification error of these samples during the training process. However, its core strategy is to assign fixed weights to minority samples, directly increase their loss contribution, and force the model to focus on minority samples, which need to be set manually and cannot distinguish the "difficult classification" of samples. Focal Loss reduces the loss contribution of easy-to-classify samples by introducing modulation factors, and dynamically adjusts the learning priority by automatically focusing on difficult-to-classify samples (including difficult cases in a few classes and noise in majority classes [15]. Therefore, in the scenario of encrypted traffic, Focal Loss is generally more suitable than weighted cross-entropy to alleviate the problem of low accuracy of minority classification.

2. Method and Model Architecture

The following model architecture is proposed, as shown in Figure 1. The following is a further explanation of the principle of the algorithm.

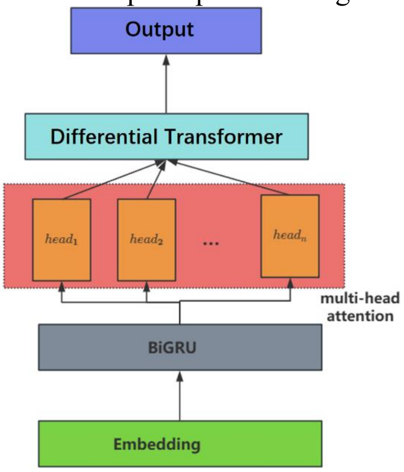

Figure 1. BiGRU-DA-BERT Architecture

BERT uses its self-attention mechanism to encode the input sequence and capture the global semantic association. Let the output of BERT be

$$\mathrm{H}_{BERT}=[\mathrm{h}_{BERT}^{1},\mathrm{h}_{BERT}^{2},\cdots,\mathrm{h}_{BERT}^{T}] \qquad (1)$$

where is the semantic encoding vector of the time step.$\mathrm{h}_{BERT}^{t}t$

However, in the category imbalance scenario, BERT's self-attention mechanism can overwhelm the unique patterns of minority samples with redundant information from the majority class. However, the semantic information encoded by BERT provides rich contextual information for subsequent processing and is the basis of the entire model.

Bidirectional Gated Recurrent Unit (BiGRU) has a bidirectional gating structure that enables modeling of the timing dependencies of sequences. The forward GRU starts processing from the beginning of the sequence, the reverse GRU starts processing from the end of the sequence, and finally splices the hidden states in the two directions.

The formula for updating the hidden state for a forward GRU is:

$$\mathrm{r}_t^f=\sigma(\mathrm{W}_r^f\mathrm{h}_{BERT}^t+\mathrm{U}_r^f\mathrm{h}_{t-1}^f+\mathrm{b}_r^f) \qquad (2)$$

$$\mathrm{z}_t^f=\sigma(\mathrm{W}_z^f\mathrm{h}_{BERT}^t+\mathrm{U}_z^f\mathrm{h}_{t-1}^f+\mathrm{b}_z^f) \qquad (3)$$

$$\tilde{\mathrm{h}}_t^f=\tanh\boxed{f0}(\mathrm{W}_h^f\mathrm{h}_{BERT}^t+\mathrm{r}_t^f\odot\mathrm{U}_h^f\mathrm{h}_{t-1}^f+\mathrm{b}_h^f)(4)$$

$$\mathrm{h}_t^f=(1-\mathrm{z}_t^f)\odot\mathrm{h}_{t-1}^f+\mathrm{z}_t^f\odot\tilde{\mathrm{h}}_t^f \qquad (5)$$

The hidden state update formula for reverse GRU is similar, except that the input sequence is reversed.

Finally, the output of BiGRU in the time step is, where is the hidden state of the reverse GRU in the time step.t$\mathrm{h}_{\mathrm{BiGRU}}^{\mathrm{t}}=[\mathrm{h}_{\mathrm{t}}^{\mathrm{f}};\mathrm{h}_{\mathrm{t}}^{\mathrm{b}}]\mathrm{h}_{\mathrm{t}}^{\mathrm{b}}\mathrm{t}$

However, when faced with noisy data, BiGRU's gating mechanism prioritizes the high-frequency majority feature patterns and suppresses the propagation of minority features. However, the timing features extracted by BiGRU are important for encrypted traffic classification, as encrypted traffic often has certain timing patterns.

The differential attention mechanism is used to solve the problems of BERT and BiGRU, highlighting the unique feature patterns of minority classes. The traditional attention mechanism calculates the similarity between each element in the input sequence and the query vector, assigns an attention weight to each element, and then weights the elements according to these weights, resulting in a representation that focuses on key information. However, when dealing with encrypted traffic data, traditional attention mechanisms can distract attention into some unrelated contexts, leading to the drowning of critical information.

The differential attention mechanism eliminates attention noise by calculating the difference between two separate Softmax attention maps. Specifically, it projects feature sequences based on BiGRU output onto two sets of Query, Key, and Value vector spaces. For each set of projections, the attention score is calculated separately. Then, through the subtraction operation, a differential attention distribution is obtained, which can highlight the elements that behave differently in the two attention distributions, that is, the key information. $\mathrm{H}^{\mathrm{BiGRU}}$

Let $\mathrm{Q}_1\mathrm{K}_1\mathrm{V}_1\mathrm{Q}_2\mathrm{K}_2\mathrm{V}_2$, are two sets of projected query, key, and value vectors, respectively. The two attention scores are calculated separately

$$\text{Attention}_1=\text{softmax}(\frac{Q_1K_1^T}{\sqrt{d}})V_1 \quad (6)$$

$$\text{Attention}_2=\text{softmax}(\frac{Q_2K_2^T}{\sqrt{d}})V_2 \quad (7)$$

where is the dimension of the vector. The output of the differential attention mechanism is d

$$\text{DiffAttn}=\text{Attention}_1-\lambda\text{Attention}_2 \quad (8)$$

where is a learnable scalar that adjusts for the degree of difference between two attention maps.λ

For better synchronous learning dynamics, reparameterization is usually performed, for example λ

$$\lambda=\exp(\lambda_{q_1}\cdot\ \lambda_{k_1})-\exp(\lambda_{q_2}\cdot\ \lambda_{k_2})+\lambda_{init} \quad (9)$$

where $\lambda_{q_1}\lambda_{k_1}\lambda_{q_2}\lambda_{k_2}\lambda_{init}\lambda$ is a learnable vector and a constant used for initialization.

Finally, the output of the differential attention mechanism is a filtered and enhanced feature representation, which focuses more on key information related to the classification task, which can significantly improve the model's classification performance for encrypted traffic. In this way, the model can accurately capture key features in complex encrypted traffic data, allowing for more accurate classification.

During the training phase, the Focal Loss function is used to calculate the loss between the model's predicted results and the true labels to guide the update of the model's parameters. With the true label as, the Focal Loss function is defined as:y_{true}

$$\mathscr{L}_{FL}=-\alpha(1-\mathbf{y}_{pred})^{\gamma}\mathbf{y}_{true}\log(\mathbf{y}_{pred}) \quad (10)$$

where is the balance factor, which is used to adjust the weights of different categories; is a modulating factor that reduces the contribution of easily classified samples.αγ

Focal Loss dynamically reduces the contribution of easy-to-classify samples (majority classes) by introducing modulation factors, while retaining the weight of difficult-to-classify samples (minority classes), so that the model pays more attention to the feature learning of minority samples during training, especially those key features disturbed by noise. Combined with the feature selection of differential attention, Focal Loss can effectively alleviate the problem of overfitting most classes by traditional cross-entropy loss.

During training, a backpropagation algorithm is used to calculate the gradient of the loss function regarding the model parameters, and an optimizer (such as the Adam optimizer) is used to update the model's parameters, iterating continuously until the model converges.

In summary, the improved model solves the problem of poor classification effect of the original model on minority classes when processing category imbalance datasets from two aspects: feature extraction and loss optimization by introducing differential attention mechanism and Focal Loss function. The differential attention mechanism highlights key features of minority classes, while the Focal Loss function ensures that the model pays more attention to minority class samples when training, improving the model's classification performance for minority classes.

3. Experimental Results and Analysis

3.1 Model Parameter Settings

The key training parameters of this model are shown in Table 1:

Table 1. Introduction to Key Parameters of the Model

parameter	Set the value	illustrate
hidden_size	768	The size of the hidden layer in the BERT model
num_attention_heads	8	Number of attention heads
Max_poisition_embedding	128	BERT input sequence length
BiGRU_hidden_size	768	The number of hidden cells in the BiGRU layer
Num_bigru_layer	2	Number of BiGRU layers
BiGRU_dropout	0.5	Dropout ratio applied between BiGRU layers
learning_rate	2e-5	Initial learning rate (dynamically adjusted during training)
warmup ratio	0.1	Preheating ratio
Batch_size	32	The number of samples processed per batch during training
Num_epochs	50	The total number of rounds trained

In the model's differential attention mechanism, the default setting of the initial value of λ (lamda) is

$$\lambda_{init}=0.8-0.6\times\exp(-0.3\cdot\ (l-1)) \quad (11)$$

Among them, the representative layer index, this setting works well in practice, which can effectively use differential denoising to eliminate attention noise and make the model focus more on key information.l∈[1,L]

The optimization algorithm used by the model is Adam, which combines the advantages of AdaGrad and RMSProp algorithms, and can calculate an adaptive learning rate for each parameter, and can dynamically adjust for the update of different parameters.

3.2 Benchmark Model Comparison Experiment

In order to evaluate the classification performance of the proposed model in the face of class imbalance datasets, several typical encrypted traffic classification models with the proposed models are selected to compare them with the proposed models on class imbalance datasets, including Deep Packet-CNN 、 Deep Packet-SAE and the use of CNN+LSTM Classification model for classification.

Let's give a basic introduction to these three models. Unlike Deep Packet Inspection (DPI), Deep Packet not only identifies encrypted traffic, but also distinguishes between VPN network traffic and non-VPN network traffic. The network architecture is based on CNN and SAE, which can simultaneously perform application identification and traffic type classification tasks. The Deep Packet framework employs two deep neural network structures, Stacked Autoencoders (SAEs) and Convolutional Neural Networks (CNNs), to classify network traffic. CNN+LSTM, which combines convolutional and recurrent networks to improve the accuracy of classification results. Convolutional networks are used to extract packet characteristics from individual packets. The recurrent neural network is trained to pick out the flow features based on the packet feature inputs of any three consecutive packets in the stream. The above three models are also classification experiments on public datasets, and the final results are better than the most advanced methods in the previous field. Therefore, the above three models are selected as benchmark models to compare the classification effects of the models in this paper.

The classification performance is comprehensively evaluated by comparing the accuracy (Precision), recall (Recall), and F1 value of different classification models in each category.

This section will comprehensively examine the performance of the model in two dimensions: the service classification dimension and the application classification dimension. To meet these two tasks, select the twelve service datasets in ISCX VPN non-VPN and the 20 application datasets in USTC-TFC to perform the corresponding tasks. Figures 2 and 3 show the distribution of sample proportions for each category in the application and service type datasets in the dataset, respectively.

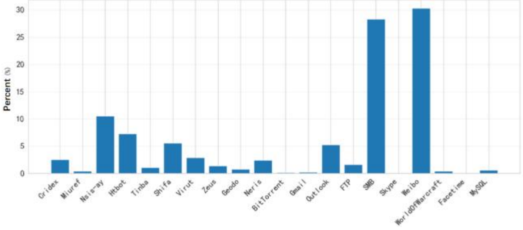

Figure 2. Distribution of USTC-TFC 2016 Application Types by Category

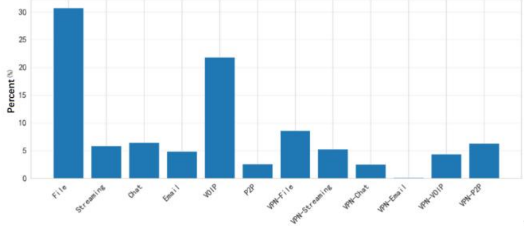

Figure 3. Distribution of the Share of ISCX-VPN SERVICE by Category

From the distribution of sample data in Fig. 2 and Fig. 3, it can be seen that there is an obvious imbalance between different types of samples in the dataset. For example, in Figure 2, the number of categories such as BitTorrent, Gmail, and Skype is significantly lower than the number of other types of samples, accounting for less than 1%, while SMB and Weibo account for more than 40% of the total dataset. In Figure 3, there is also a large gap between the number of samples of VPN-Email and P2P types and other traffic types. Sample imbalance is inevitable in some cases, but unbalanced samples will seriously affect the learning of small sample flow data features, and have a great impact on the classification effect of the model.

(1) Service type classification tasks

First, we conducted a classification experiment on the twelve categories of service datasets in ISCX VPN non-VPN, on which the performance index data of each model are paired as shown in Table 2 below

Table 2. Comparison of ISCX-VPN Service Benchmark Model Performance Indicators

Classification model	Precision	Recall	F1
Deep Packet-CNN	0.94	0.93	0.93
Deep Packet-SAE	0.92	0.92	0.93
CNN+LSTM	0.91	0.91	0.91
BiGRU-DA-BERT	0.96	0.95	0.96

By reviewing the experimental results, it can be seen that the BiGRU-DA-BERT network has the best performance in four indicators: precision, recall and F1 value, indicating that the BiGRU-DA-BERT proposed in this paper has good learning ability and good adaptability to different types and different distributions of data. The following table shows the effect of different models for classifying service types, and the following is to compare the F1 values of different classification models on each category for the problem of unbalanced datasets, which can reflect the degree to which the model is affected by unbalanced data. The results of the comparison experiment are shown in Figure 4 below

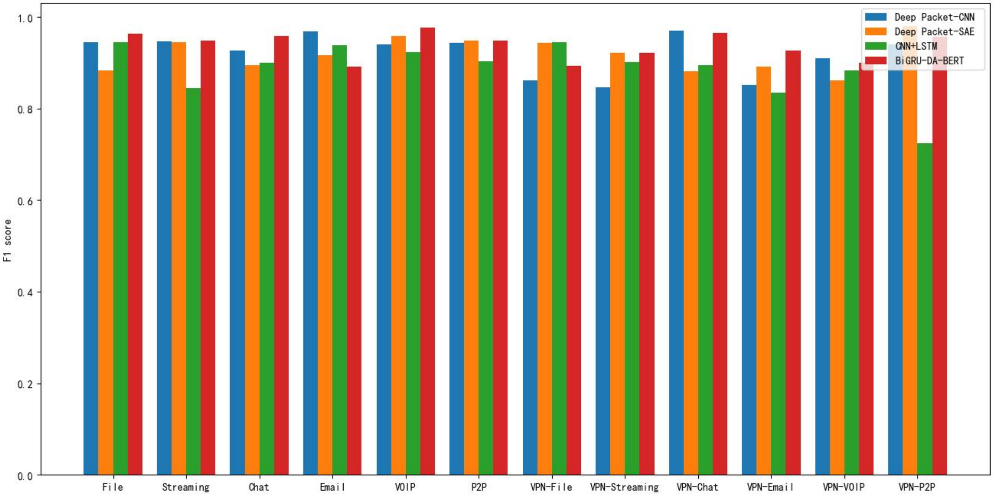

Figure 4. Comparison of the Performance of Each Model in a Specific Category of Service Type Classification Tasks

As can be seen from Fig. 4, the four classification models can achieve good classification effects in the face of most classes, such as File and VOIP, and the F1 value of Deep Packet-CNN and CNN+LSTM reaches more than 95%, and the model proposed in this paper reaches more than 98%. However, for a few categories such as VPN-Email and VPN-P2P, Deep Packet-CNN performed poorly, while CNN+LSTM even had an F1 value of less than 80% in the VPN-P2P category, and Deep Packet-SAE had an F1 value of less than 85% on VPN-VOIP. The model proposed in this paper also performs well in a few categories, even with an F1 score of 95.2% on VPN-Email, and an F1 score of nearly 99% in most categories. The above data are enough to show that the classification performance of the proposed model in this paper is significantly better than that of the typical Deep Packet-CNN, Deep Packet-SAE and CNN+LSTM combined models in the face of unbalanced traffic data.

(2) Application type classification task

Then there are classification experiments for 20 application categories in USTC-TFC, and the performance indicators of each model are paired on this dataset, as shown in Table 3 below:

Table 3. Comparison of the Performance Indicators of the USTC-TFC Application Type Benchmark Model with This Model

Classification model	Precision	Recall	F1
Deep Packet-CNN	0.93	0.92	0.93
Deep Packet-SAE	0.91	0.94	0.91
CNN+LSTM	0.91	0.92	0.93
BiGRU-DA-BERT	0.96	0.96	0.97

As can be seen from Table 3, the proposed model also achieves better classification results than the comparison model in application type classification, and performs best in four indicators: precision, recall, and F1 value.

Similarly, for the category imbalance of application classification traffic data, Figure 5 shows the comparison of the F1 values of the detailed classification results of each application category.

As can be seen from Figure 5, the F1 value of the four classification models still maintains a high level in the face of most classes, such as Weibo and SMB, and the average can reach more than 95%. However, for categories such as MySQL and Skype, which account for less than 1% of the number, the other three models do not perform well, with the F1 value of Deep Packet-SAE on MySQL even below 60%, and the F1 value of Deep Packet-CNN on minority Viruts is only 83.2%, and the F1 score of CNN+LSTM combination model on minority Miuref is only 82.9%. Although the performance of the model proposed in this paper has also decreased, it is basically above 91%. The above data can show that the model proposed in this paper has a good classification effect for the problem of unbalanced datasets, whether it is a classification task of service type or an application type.

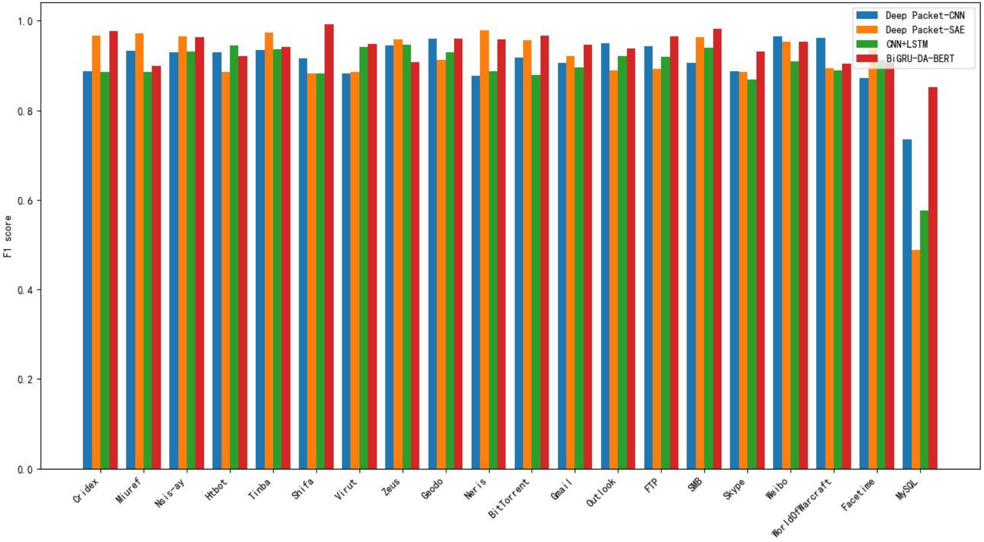

Figure 5. Comparison of the Performance of Each Model in a Specific Category for Application Classification Tasks

3.3 Ablation Experiment

3.3.1 Comparative experiment of attention mechanism

In order to verify the effectiveness of the differential attention mechanism, this section sets up a comparative experiment, starting from the training of the model, the model with the differential attention mechanism and the model without the differential attention mechanism are compared, and Figures 6 and 7 are the comparison results of the training process between the two.

From Fig. 6 and Fig. 7, it can be seen that the loss value and accuracy of the model tend to stabilize after training multiple epochs, and both models show good classification results. However, it can be seen from Fig. 6 that the final accuracy of the model with the differential attention mechanism, that is, the method proposed in this paper, is higher than that of the model without the differential attention mechanism. In addition, it can also be seen from Figure 7 that the model with the differential attention mechanism converges faster during the training process, which may be because differential attention eliminates context-independent noise by calculating the difference between two independent Softmax attention maps, making the model more focused on key information. This mechanism may optimize the direction of gradient updates and reduce invalid parameter adjustments, thereby improving training efficiency.

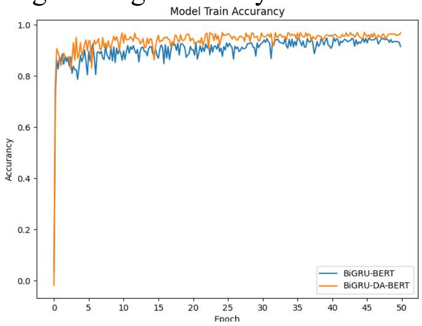

Figure 6. Comparison of the Accuracy of the Model Training Process with the Introduction of Differential Attention Mechanism and the Model without the Introduction

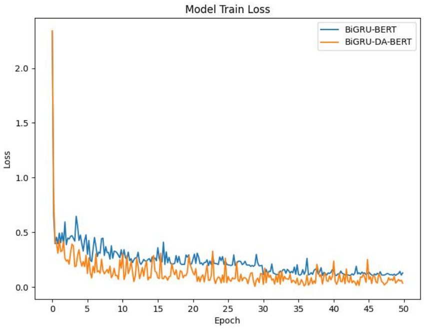

Figure 7. Comparison of the Loss of the Model Training Process with the Introduction of Differential Attention Mechanism and the Model without Introduction

Table 4 is a comparison of the experimental results of the two models on the service type classification task, and Table 5 is the comparison of the experimental results of the two models on the application type classification task.

Table 4. Comparison of Experimental Results on Service Type Classification Tasks

Classification model	Precision	Recall	F1
Model with Diff-Attention	0.94	0.94	0.95
Model without Diff-Attention	0.96	0.96	0.97

Table 5. Comparison of Experimental Results on Application Type Classification Tasks

Classification model	Precision	Recall	F1
Model with Diff-Attention	0.94	0.95	0.94
Model without Diff-Attention	0.97	0.97	0.97

From the experimental results in Tables 4 and 5, it can be seen that the indicators of the model based on the differential attention mechanism exceed the performance of the model without the introduction of the differential attention mechanism model in the two classification tasks, which fully shows that the differential attention mechanism has played a significant role in improving the classification effect of the model.

3.3.2 Focal loss comparison experiment

In view of the uneven distribution of categories in the dataset, the accuracy of some categories with relatively scarce samples is not good in the classification process. In order to effectively eliminate the negative effect of data imbalance on the classification accuracy of the model, the proposed model uses Focal Loss as the loss function in the model training process. In order to verify the effect of Focal Loss in dealing with the problem of data imbalance, this section uses the widely used cross entropy loss function as a reference benchmark to carry out relevant experimental comparison and analysis. Specifically, in the two specific scenarios of application classification task and traffic category classification task, the comparison results based on the F1 value are shown in Figures 8 and 9.

In Figure 8, the orange curve is the classification result of the service type of Cross Entropy Loss, and the blue curve is the classification result of the Focal Loss loss function. It can be seen that when using cross-entropy as a loss function, the recognition accuracy of categories with a small number of samples, such as Streaming and VPN-P2P, is significantly lower than that of other application categories, while the F1 value of the model using the Focal Loss loss function is significantly improved in these categories.

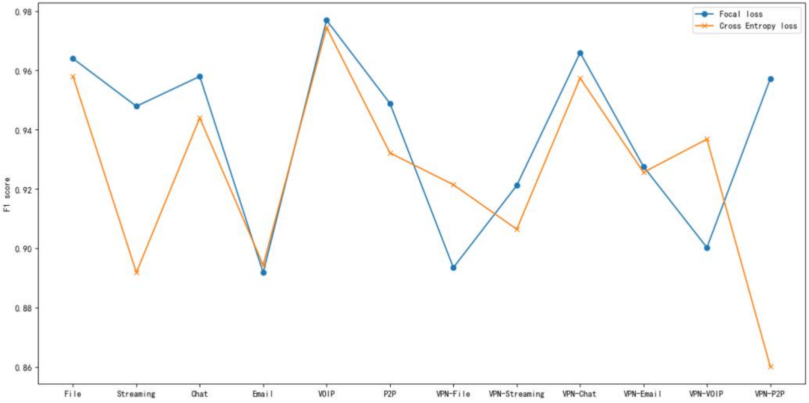

Figure 8. Experimental Comparison of Focal Loss for Service Type Classification Tasks

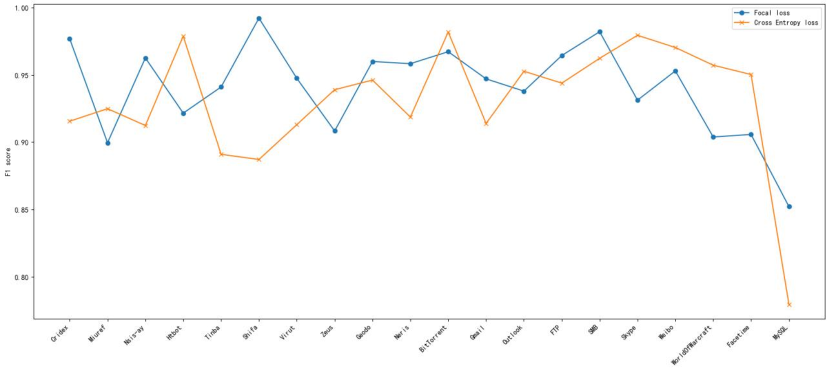

Figure 9. Comparison of Focal Loss Experiments of Application Type Classification Tasks

From the comparison curve of the F1 value of the application type classification task in Fig. 9, it can be seen that the F1 value of the model using the cross-entropy loss function is much lower than that of Focal Loss in MySQL and Tinba categories with fewer class samples.

The above experimental results show that the use of Focal Loss in the model can well solve the impact of the difference in sample number distribution on the model performance, effectively improve the classification accuracy of the category sample with fewer samples, and also enhance the generalization ability of the model as a whole.

4. Conclusion

This paper mainly proposes an improved encrypted traffic classification model based on BERT-BiGRU, which solves the problem of poor classification effect for minority classes when the traffic dataset is unbalanced, and the overall classification accuracy on the unbalanced dataset is also improved. Based on the ISCX VPN-nonVPN and USTC-TFC datasets, the overall classification accuracy is improved by 2.3% and the classification accuracy of minority classes is improved by 15.3% on average. In addition, in the multi-dimensional, that is, application type and service type classification comparison experiments, compared with the benchmark model, the overall accuracy, recall rate and F1 value are improved by 2.3%, 1.9% and 2.2%, respectively, and the classification effect of minority classes is also significantly improved, which fully demonstrates the effectiveness of the classification method proposed in this paper.

Acknowledgement

This study was financially supported by the Science and Technology Program of Southern Power Grid Corporation (031900KC23070047).

References

[1] X. Wang, "FDGM: A Lightweight Model for Privacy-Preserving Encrypted Traffic Classification," IEEE Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2025, pp. 686-691, doi: 10.1109/ITOEC63606.2025.10967753.

[2] J. Kotak, I. Yankelev, I. Bibi, Y. Elovici and A. Shabtai, "VPN-Encrypted Network Traffic Classification Using a Time-Series Approach," in IEEE Transactions on Network and Service Management, vol. 22, no. 2, pp. 2225-2242, April 2025, doi: 10.1109/TNSM.2025.3543903.

[3] Z. Chen, G. Cheng, Z. Wei, D. Niu and N. Fu, "Classify Traffic Rather Than Flow: Versatile Multi-Flow Encrypted Traffic Classification With Flow Clustering," in IEEE Transactions on Network and Service Management, vol. 21, no. 2, pp. 1446-1466, April 2024, doi: 10.1109/TNSM.2023.3322861.

[4] G. Lv, R. Yang, Y. Wang and Z. Tang, "Network Encrypted Traffic Classification Based on Secondary Voting Enhanced Random Forest," 2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 2020, pp. 60-66, doi: 10.1109/CCET50901.2020.9213165.

[5] Wang W, Zhu M, Wang J, et al. End-to-end encrypted traffic classification with one dimensional convolution neural networks//2017 IEEE international conference on intelligence and security informatics (ISI). IEEE, 2017: 43-48.

[6] Lotfollahi M, Siavoshani M J, Zade R S H, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing, 2020, 24(3):1999-2012.

[7] Y. Liu, X. Wang, B. Qu and F. Zhao, "ATVITSC: A Novel Encrypted Traffic Classification Method Based on Deep Learning," in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 9374-9389, 2024, doi: 10.1109/TIFS.2024.3433446.

[8] N. Malekghaini. "AutoML4ETC: Automated Neural Architecture Search for Real-World Encrypted Traffic Classification," in IEEE Transactions on Network and Service Management, vol. 21, no. 3, pp. 2715-2730, June 2024, doi: 10.1109/TNSM.2023.3324936.

[9] Cinar Y G, Mirisaee H, Goswami P, et al. Period-aware content attention RNNs for time series forecasting with missing values. Neurocomputing, 2018, 312: 177-186.

[10] Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview. IEEE communications magazine, 2019, 57(5): 76-81.

[11] Papadogiannaki E, Ioannidis S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. ACM Computing Surveys (CSUR), 2021, 54(6): 1-35.

[12]Salman O, Elhajj I H, Kayssi A, et al. A review on machine learning–based approaches for Internet traffic classification. Annals of Telecommunications, 2020, 75(11): 673-710.

[13]Lin P, Ye K, Hu Y, et al. A novel multimodal deep learning framework for encrypted traffic classification. IEEE/ACM Transactions on Networking, 2022, 31(3): 1369-1384.

[14]Li R, Xiao X, Ni S, et al. Byte segment neural network for network traffic classification//2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). IEEE, 2018: 1-10.

[15]S. Cui. "FG-SAT: Efficient Flow Graph for Encrypted Traffic Classification Under Environment Shifts," in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 5326-5339, 2025, doi: 10.1109/TIFS.2025.3571663.