Traffic Prediction for 5G: A Deep Learning Approach Based on Lightweight Hybrid Attention Networks

Jian Su[a,*], Huimin Cai[a], Zhengguo Sheng[b], A. X. Liu[c] and Abdullah Baz[d]

[a]School of software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[b]Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, U.K.
[c]Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
[d]Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia

ARTICLE INFO

Keywords:
Network Traffic Prediction
Hybrid Attention
Lightweight

ABSTRACT

The maturity of 5G technology provides a guarantee for increasingly large communication networks, while the resources required for communication and computation are also increasing, and reasonable resource allocation can improve the efficiency of network communication and reduce the consumption of communication resources. Existing deep learning methods have been able to predict network traffic to a certain extent, so as to solve the communication efficiency and resource consumption problems in the field of integrated sensing, communication and computation (ISCC) through rational resource allocation. However, the following problems still exist: (1) The feature learning ability of the prediction model is insufficient, and the prediction accuracy needs to be improved. (2) Powerful and complex deep learning methods lead to an increase in the prediction cost of the model. To address these problems, this paper proposes a deep learning method based on a lightweight hybrid attention network. In order to capture the key features of 5G data more effectively, an efficient hybrid attention mechanism (EHA) is proposed. After this attention is applied to convolution, the key information can be well enhanced. We use depthwise separable convolution in feature extraction, which greatly improves the efficiency of lightweight convolution layer (LC) in feature extraction. Combined with the efficient hybrid attention mechanism (EHA), the proposed model has better lightweight properties. Experimental results show that the model proposed in this paper has lower RMSE and MAE values on the three datasets, as well as fewer parameters and computational effort compared to the baseline scheme.

1. Introduction

With the rapid development of science and technology, the way of life of human beings is gradually changing. The development of the Internet of Things (IoT) gradually push human society into the era of Internet of Everything, and more and more devices are connected to the Internet, which makes the scale and influence of the network expanding. With the access of more and more devices, the traditional 4G network can no longer meet the existing demand. With the continuous development of 5G network in recent years, its advantages gradually appear. Compared with 4G networks, 5G networks have higher data transmission speeds, lower latency, greater network capacity, and higher reliability. Along with the development of 5G technology, the scale of the Internet of Things (IOT) is also getting bigger, which greatly facilitates the rapid development of some fields such as autonomous driving, smart cities, virtual reality, etc. [1]. However, the proliferation in the number of various devices and sensors lead to an increasingly large amount of data to be communicated, and the problem of optimizing the allocation of resources in the field of integrated sensing, communication, and computation (ISCC) become more and more important. The proposal of integrated sensing and communication (ISAC) solves the problem of transmission efficiency of communication data to a certain extent [2, 3]. Meanwhile, along with the development of artificial intelligence, the application of machine learning algorithms to the field of integrated sensing, communication and computation (ISCC) become a research direction worth exploring [4]. Through the learning of historical data can be obtained to a certain extent, the data trend in a certain period of time in the future, a reasonable analysis of this trend can often greatly optimize the allocation of resources, so the prediction of network traffic become the key to the problem. At this stage, many methods emerge for network traffic prediction research. In general, network traffic prediction methods are divided into two main categories: traditional statistics-based methods and machine learning-based methods.

In earlier times, the problem of flow prediction is mainly dealt with using traditional statistical methods. These methods rely mainly on the statistical characteristics and patterns of historical data to make predictions. They usually use a number of statistical techniques to analyze and model the data and then use statistical models to make predictions. An example is the autoregressive model (AR), which is best characterized by its ability to make full use of prior data to regress forecasts on later data. The proposal of differential autoregressive moving average model (ARIMA) [5] makes the time series problem more effectively dealt with, and ARIMA also achieves good results in the neighborhood of network traffic prediction. The article [6] made full use of

*Corresponding author
sj890718@gmail.com (J. Su); 202212490774@nuist.edu.cn (H. Cai); z.sheng@sussex.ac.uk (Z. Sheng); alexliu@cse.msu.edu (A.X. Liu); aobaz01@uqu.edu.sa (A. Baz)
ORCID(s):

the ARIMA model, in order to cope with the highly dynamic nature of network traffic, the author divides the traffic signal into two parts consisting of normal changes and abnormalities consisting of sudden changes for separate treatment. This enables better network traffic prediction and anomaly detection. Article [7] proposed a network traffic prediction model based on nonlinear time series ARIMA/GARCH. The model combines the ARIMA model with the nonlinear GARCH model. Thereby, the model can capture salient traffic features not only on large time scales, but also on small time scales, and has better prediction accuracy compared to the FARIMA model. Article [8] utilized the autocorrelation function for the exploration of trending and cyclical features, while the product seasonal autoregressive integrated moving average (ARIMA) model was used to achieve better prediction results. Article [9] noticed the α-stable modeling property in the time domain and the sparsity in the spatial domain, proposed the α-stable model for network traffic prediction. And good prediction performance is obtained after the validation of simulation experiments. However, the limitations of the linear model are gradually manifested with the deepening of the research. Due to the complexity of network traffic data characteristics, it is difficult for a single linear model to achieve better prediction results. The proposal of nonlinear prediction models alleviates this problem to some extent. For example, Autoregressive Conditional Heteroskedasticity (ARCH) model [10]. The article [11] proposed a probabilistic jump prediction algorithm based on the ARCH model to characterize the traffic data rate dynamics of the dataset, which ensures a strong dynamic configuration performance of the framework. Although the nonlinear prediction model shows some degree of improvement compared to the linear prediction model, it is still some distance away from the expectation.

With the rapid development of machine learning, especially deep learning, the performance of network traffic prediction is greatly improved. Meanwhile, along with the continuous expansion of network size, the increasingly large network traffic data also makes machine learning-based methods increasingly superior to traditional methods. Compared with traditional statistical methods, early shallow learning methods such as Support Vector Regression (SVR) [12] and Gaussian Process Model have many improvements in network traffic prediction. And then, with the rise of deep learning, its powerful prediction ability makes more and more researchers start to apply it to network traffic prediction. The proposal of Convolutional Neural Network (CNN) [13] laid the foundation for a series of researches. Later, Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM) [14], and Gated Recurrent Units (GRU) [15] are widely used in time series prediction problems. The proposal of Temporal Convolutional Networks (TCNs) [16] also greatly contributed to the study of time series forecasting problems.

In summary, at this stage, most of the 5G network traffic prediction as a kind of network traffic prediction is based on deep learning methods. Although the huge dataset and powerful arithmetic power make the deep learning models have good results, there are still some problems:

- At this stage, many deep learning methods do not accurately grasp the key data in the feature learning stage of the model and enhance the ability to learn the features of the key data, which leads to poor prediction results.
- With the development of deep learning, models become more complex. Too complex prediction model will lead to the rise of prediction cost, which brings some obstacles to the practical application of prediction model.

In order to solve the above problems, this paper proposes a deep learning method based on lightweight hybrid attention network. The main contributions are as follows.

- In the feature extraction stage, an efficient hybrid attention mechanism (EHA) is proposed to enhance the weight of key features in 5G data, so that the model can better learn the spatio-temporal characteristics.
- Depthwise separable convolution is used in lightweight convolution layer (LC), which greatly reduces the parameters and computation cost of LC layer compared with traditional convolution layer. And 1×1 convolution is used in the efficient hybrid attention mechanism (EHA) to achieve lightweight.
- By comparing the proposed method with other mainstream prediction methods on the datasets provided by Telecom Italia, it is verified that the proposed method has better prediction performance. Moreover, the computational consumption experiments also demonstrate the better lightweight property of the proposed method.

2. Related work

Time series prediction and time series analysis have been popular research topics, and 5G network traffic prediction is an important part of the time series prediction. With the rapid development of IoT, the growth rate of network traffic is also considerable. At this time, reasonable and effective 5G network traffic prediction becomes crucial. Efficient network traffic prediction can better lead to rational allocation of network resources.

Most of the latest network traffic prediction is based on deep learning methods, and this class of methods has significantly better performance than traditional methods. Jaffry et al. [17] proposed a network traffic prediction model based on LSTM and compared it with ARIMA and FFNN on a real dataset. The results show that LSTM has higher accuracy and the model converges more easily. However, this method only takes into account the temporal characteristics of network traffic data and ignores its spatial characteristics.

Zhang et al. [18] proposed a network traffic prediction model based on a densely connected convolutional neural

network for the application scenario of network traffic prediction on a city-wide scale. The model is able to capture the spatio-temporal characteristics of the traffic data better, and the parameter matrix based fusion scheme proposed by the authors is able to make the performance of the model go further. In validation experiments on the Telecom Italia dataset, the model has lower RMSE values compared to the traditional HA, ARIMA and LSTM models. However, this method cannot fully learn complex spatial features through traditional convolution alone, and complex neural networks also increase the consumption of computing resources.

Zhang et al. [19] proposed a hybrid spatio-temporal network (HSTNet) to address the above problems. By introducing deformable convolutional units, temporal features, and an attention mechanism to improve the ability to extract complex spatial features, the accuracy of prediction, and the robustness of the model. However, the use of deformable convolutions has to some extent increased the computational cost of the prediction model.

Mohseni et al. [20] investigated the effectiveness of various deep learning methods for network traffic prediction. After experimental evaluation, FCSN and 1D-CNN have the smallest MAE value (0.29) . But 1D-CNN has less number of parameters, complexity and smaller execution time. Rao et al. [21] proposed a deep learning method that considers dynamic non-local spatial correlation, self-attention and correlation of spatio-temporal feature fusion. In this method, NLG-NLAM is proposed to accurately capture the correlation between features in non-local spatial areas, and a calibration layer is designed to clarify the key role of different periodic features and eliminate the influence of irrelevant features on prediction.

Compared with traditional methods, the above deep learning methods have different aspects of progress, but they also have shortcomings in some aspects. 5G network traffic data has strong spatiotemporal characteristics, but the previous methods can not fully learn the dependency of 5G network traffic data in the spatio-temporal domain, especially the key features in the time domain lack a certain ability to capture. Therefore, we hope to propose a solution to the problem of insufficient ability to capture key features in the spatio-temporal domain. In addition, most of the above deep learning methods have a high number of parameters and complexity, so we hope to reduce the computational consumption of the model as much as possible on the premise of ensuring the prediction performance.

3. Data

3.1. Dataset

The dataset used in this paper is from Telecom Italia [22]. The Telecom Italia dataset is the most widely used open dataset in the network traffic prediction research literature. The dataset consists of traffic time series from November 1, 2013 to January 1, 2014, at 10 min intervals, and consists of three segments: short message service (SMS), call service (Call), and internet access (Internet). The entire spatial area is divided into 100×100 grids, indicating that the Milan area is a superposition of 10,000 cells, each of which has a size of approximately 235×235 square meters, and whose values represent the statistics of incoming and outgoing traffic to and from the area. By analyzing the call detail records generated by the Telecom Italia cellular network, different attributes are extracted for each grid every 10 minutes, including SMS, Call and Internet usage data. Based on this dataset, univariate and multivariate spatio-temporal prediction problems can be considered. The time span is from 00:00 on November 1, 2013 to 00:00 on January 1, 2014. The dataset can be denoted as $\mathbf{V}_{c,t}$, where $\mathbf{V}_{c,t}$ has four dimensions $[c, t, H, W]$, c denotes the type of data, including SMS, Call, and Internet. t denotes the current moment, where t belongs to $\{0, 1, 2, ..., T\}$, and T denotes the maximum value of the moment. The whole area is divided into $H \times W$ blocks, H and W denote the number of rows and columns of the cell respectively. $\mathbf{V}_{c,t}$ can be described as Eq. (1).

$$\mathbf{V}_{c,t} = \begin{bmatrix} v_{c,t}^{(1,1)} & \cdots & v_{c,t}^{(1,w)} & \cdots & v_{c,t}^{(1,W)} \\ \vdots & \ddots & \vdots & & \vdots \\ v_{c,t}^{(h,1)} & \cdots & v_{c,t}^{(h,w)} & \cdots & v_{c,t}^{(h,W)} \\ \vdots & & \vdots & \ddots & \vdots \\ v_{c,t}^{(H,1)} & \cdots & v_{c,t}^{(H,w)} & \cdots & v_{c,t}^{(H,W)} \end{bmatrix}, \tag{1}$$

where $v_{c,t}^{(h,w)}$ represents the 5G network traffic data at moment t on partition (h, w).

3.2. Data Analysis

The complexity of 5G network traffic increases the difficulty of feature extraction, especially its nonlinear relationship in time and space domains. For this reason, a detailed data analysis is needed to fully exploit its characteristics in the time and space domain, thus making the prediction of 5G network traffic more accurate.

3.2.1. Time Domain Correlation

We select 24 hours of data from the same region to analyze, with a time interval of 10 minutes, the results of which are shown in Figure 1, Figure 2 and Figure 3. It contains three data types (SMS, Call, Internet) from the Telecom Italia dataset. The horizontal axis of the graph represents the sampling time. The vertical axis of the graph represents the activity of the different types of data. As can be seen from the graph, the traffic data itself shows a heterogeneous distribution within 24 hours in the same area. For example, the number of hours of text messages (SMS) and calls (Call) is significantly lower at night than during the day. On the other hand, internet traffic data is not significantly lower at night than during the day due to its own characteristics. However, the three types of traffic data are generally correlated, and all of them have the characteristic of more daytime and less nighttime, which is also in line with the actual situation.

Then we select the data of the region in a week's time to be analyzed, and the results are shown in Figure 4, Figure 5

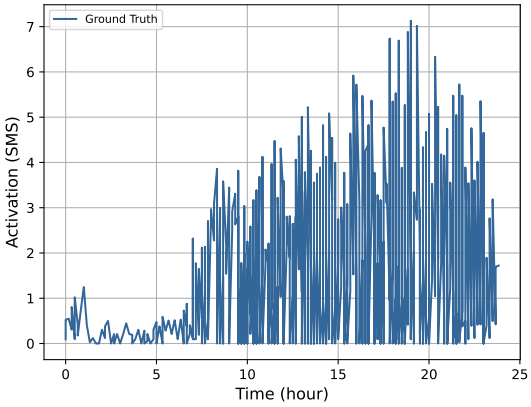


Figure 1: Time-domain distribution of 24-hour flow data (SMS).

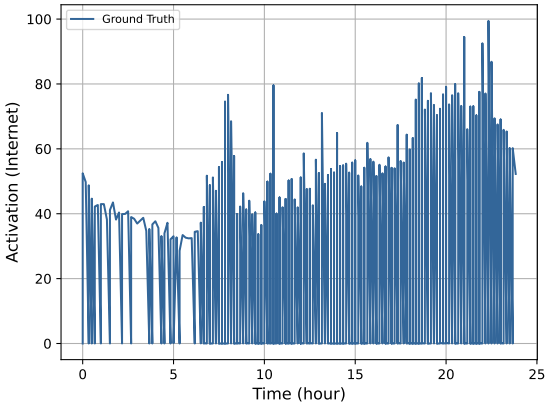


Figure 3: Time-domain distribution of 24-hour flow data (Internet).

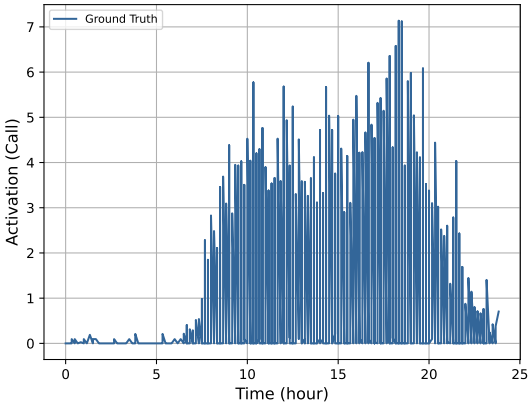


Figure 2: Time-domain distribution of 24-hour flow data (Call).

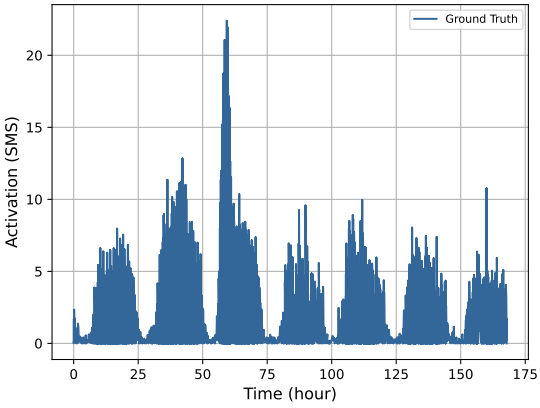


Figure 4: Time-domain distribution of 168-hour flow data (SMS).

and Figure 6. From the figure, we can easily observe that the traffic data has obvious periodicity, such as the periodicity presented by every 24 hours. At the same time, the traffic data is also characterized by burstiness. For example, in the 168-hour time-domain distribution graph of SMS, the traffic data from 50-75 hours is significantly more active than the other hours. This is due to the fact that the day was Christmas, so it lead to a surge in traffic data. Therefore, the processing of sudden traffic data is also the key to improve the prediction ability of the model.

3.2.2. Spatial Domain Correlation

5G network traffic data has both temporal and spatial characteristics, so the analysis of spatial characteristics is equally important. Figure 7, 8 and 9 show the spatial distribution of three datasets in different regions at the same time. We can find that although the data in the three datasets are different in spatial distribution, there are similar spatial differences in general. From a global perspective, the higher activity tends to be in the center of the city, while the activity is relatively low at the edge of the city. From the local point of view, 5G network traffic is correlated within a certain range. For example, the center of the city is more active but with similar values. This shows that 5G data is globally different and locally relevant. The spatial characteristics of 5G network traffic require targeted processing to obtain better prediction results.

We use the Pearson correlation coefficient measure [23] to further analyze the spatial correlation of the data. The Pearson correlation coefficient measure is widely used in correlation analysis and is defined as shown in Eq. (2).

$$\rho = \frac{cov(x_t^{(r,c)}, x_t^{(r',c')})}{\sigma_{x_t^{(r,c)}} \sigma_{x_t^{(r',c')}}}, \tag{2}$$

where $cov(x_t^{(r,c)}, x_t^{(r',c')})$ denotes the covariance of neighboring regions $x_t^{(r,c)}$ and $x_t^{(r',c')}$, and $\sigma_{x_t^{(r,c)}}$ and $\sigma_{x_t^{(r',c')}}$ denote the standard deviation of regions $x_t^{(r,c)}$ and $x_t^{(r',c')}$, respectively.

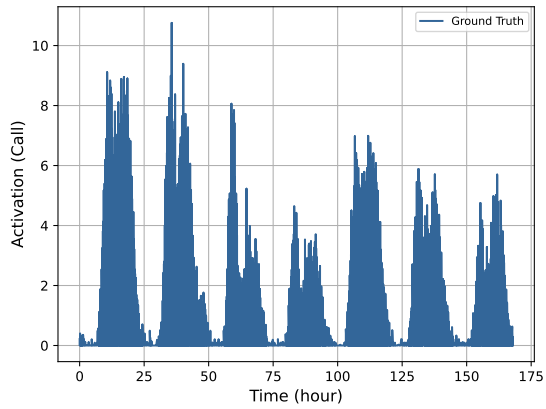


Figure 5: Time-domain distribution of 168-hour flow data (Call).

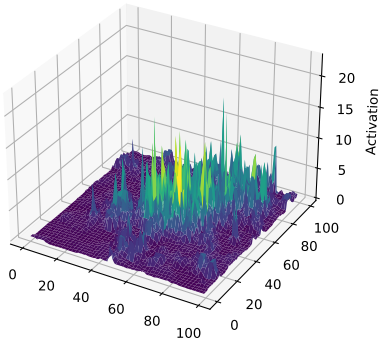


Figure 7: Distribution of 5G network traffic data in terms of spatial activeness (SMS).

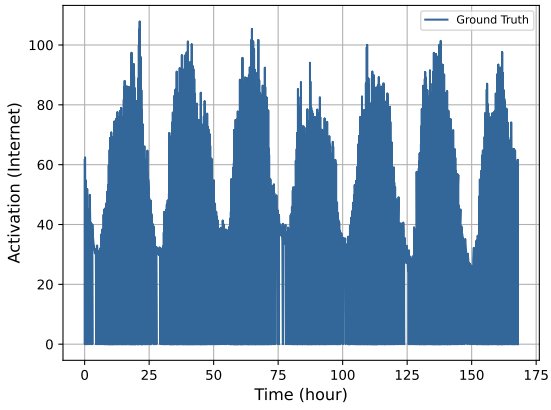


Figure 6: Time-domain distribution of 168-hour flow data (Internet).

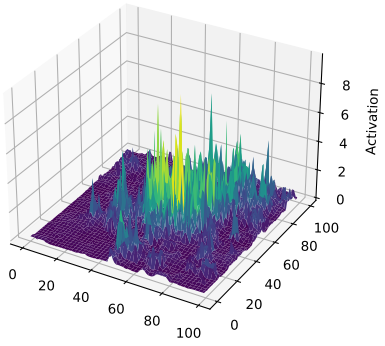


Figure 8: Distribution of 5G network traffic data in terms of spatial activeness (Call).

We perform correlation analysis and use cell (7, 7) as the target cell, and the result is shown in Figure 10. From the figure, we can find that the regions near the target cell usually have high correlation coefficients, which indicates the strong correlation of the neighboring regions. However, the distance is not the only factor that affects the correlation. Some regions that are far from the target cell still have strong correlation. This shows that 5G network traffic does not only have local correlation, but also has non-local correlation. Therefore, it is necessary to capture the corresponding key features for these factors to improve the predictive ability of the model.

4. Proposed Approach

5G network traffic data is a kind of complex data with both temporal and spatial characteristics, and most of the existing deep learning methods can handle time-series data well. However, there are still shortcomings in the ability to capture key features in the temporal and spatial domain. Moreover, as the computational volume of deep learning models becomes larger, their computational costs also increase. Aiming at the above problems, This paper proposes a deep learning method based on hybrid attention and lightweight. The framework of the method is shown in Figure 11, where LC stands for lightweight convolution and EHA stands for efficient hybrid attention. The first is the preprocessing of the data, so that the data is made more suitable for neural network feature extraction by means of regularization and normalization, etc., after which the data is divided into two groups to be processed separately and independently. One group of data is short-term data, specifically the first 1 hour, 2 hours and 3 hours of the prediction point of the sampling point data. The other set of data is long term data, specifically the sampling point data at the same time 1 day, 2 days and 3 days before the prediction point. The data in this dataset are grid data collected according to time intervals, and the data of each time point is a 100×100 grid. The article [18] utilizes DenseNet for network traffic prediction and proves that DenseBlock has powerful performance. In the feature learning stage,

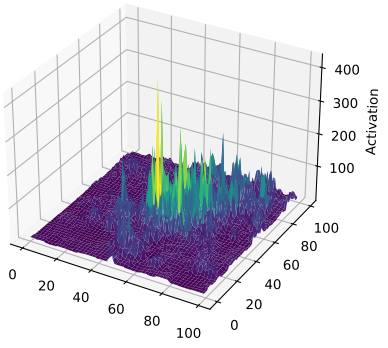


Figure 9: Distribution of 5G network traffic data in terms of spatial activeness (Internet).

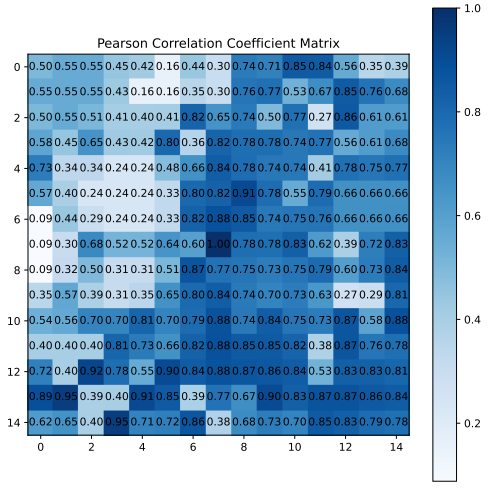


Figure 10: 5G network traffic spatial correlation analysis map.

based on DenseNet, we use LC layer for efficient feature extraction and EHA layer for weight enhancement of key features. In the LC layer, we use 1×1 convolution and depthwise separable convolution [24] to reduce the number of parameters and calculation consumption on the premise of ensuring a certain feature extraction capability. In the EHA layer, we use hybrid attention combining channel attention and spatial attention after the convolutional operation to improve the model's learning ability. Moreover, we use efficient channel attention (ECA) [25] as the channel attention mechanism in EHA. We apply SEBlock [26] to the end of feature learning to capture key features by strengthening the weights of important channels. A fusion scheme based on parameter matrix is proposed to fuse features of different granularities [18]. We use this fusion method to fuse the output features of different granularities, and the result is processed by the sigmoid function to obtain the prediction result. Finally, the prediction result and the real value are input into the evaluation function to get the loss, so as to evaluate the prediction ability of the model.

4.1. Lightweight Convolution Module

The excellent contribution of convolutional neural networks to fields such as computer vision proves their powerful feature extraction ability and allows convolution to be applied to more fields. 5G network traffic has strong local correlation, and properties such as local awareness of convolution can capture the features of 5G network traffic very well. Therefore, we use convolution for its feature extraction. However, the effect of pure convolutional neural network is not outstanding, which makes the model has large limitations. The residual network (ResNet) proposed by He et al. [27] breaks through the depth limitation of neural networks, which makes the development of deep neural networks guaranteed. Subsequently, the article [28] proposed DenseNet, in which a densely connected DenseBlock is used to implement and enhance feature reuse, which can make the information flow between the layers of the network to be maximized. Therefore, in the feature learning module, we use the optimized DenseBlock for feature extraction.

While most convolution modules have powerful performance, they are often accompanied by large computational effort and high complexity. We reduce the computational cost of the model by using depthwise separable convolution. Depthwise separable convolution divides the standard convolution into two steps: depthwise convolution is responsible for processing the input feature maps on different channels, and pointwise convolution is responsible for linearly combining the outputs of depthwise convolution. This design allows the model to reduce the number of parameters and computational complexity while still maintaining good feature representation.

We set the shape of the input image as $H_i \times W_i \times C_i$, where C_i is the number of channels, H_i and W_i are the width and height of the image respectively. The convolution kernel size is $F \times F$ and the output feature map format is $H_o \times W_o \times C_o$. The step size of the convolution process is 1, no padding is used, and the bias is 0. At this point, the formula for the number of parameters P_c of the regular convolution can be expressed as follows,

$$P_c = F \times F \times C_i \times C_o. \tag{3}$$

The computational volume C_c of the regular convolution can be expressed as follows,

$$C_c = F \times F \times C_i \times W_o \times H_o \times C_o. \tag{4}$$

Depthwise separable convolution consists of two steps, depthwise convolution and pointwise convolution. It divides the ordinary convolution into two separate operations for processing spatial regions and channels. In the depthwise convolution part, there are C_i single-channel convolution kernels. Without changing the depth of the input feature

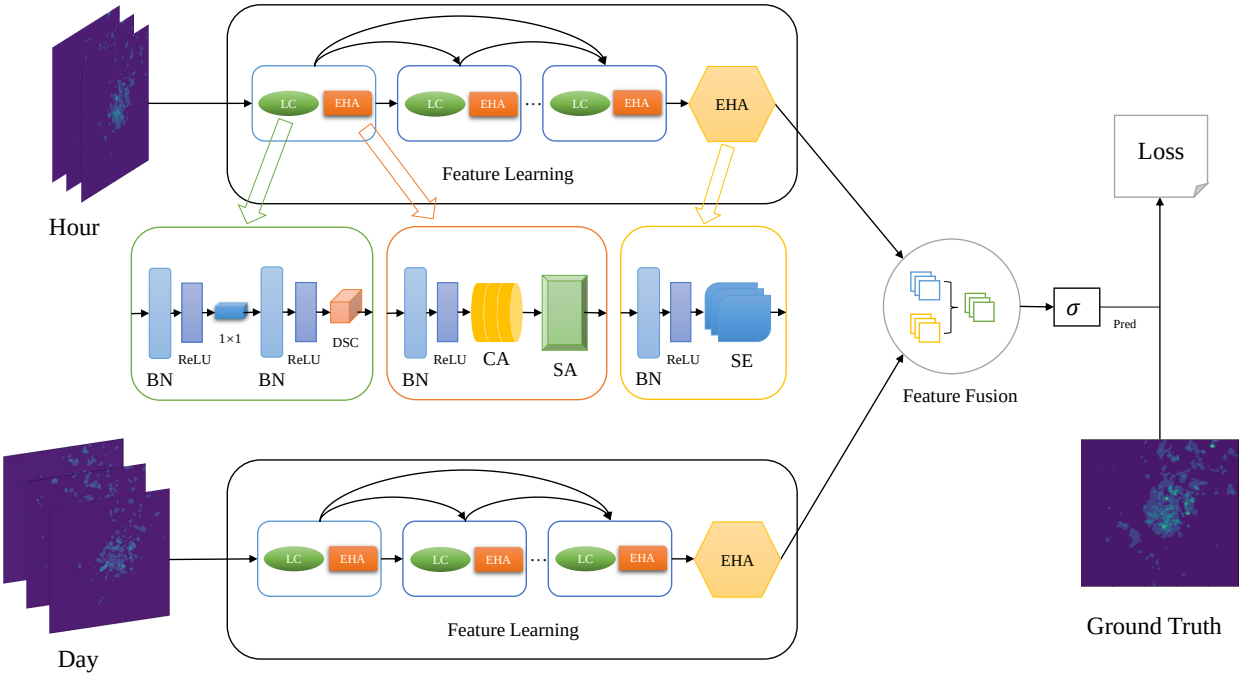


Figure 11: Overall structure of the model.

map, the convolution kernels perform the convolution operation on each channel separately, and the number of channels of the output feature map is still C_i. At this point the shape of the feature map is $H_t \times W_t \times C_i$.The formula for the number of parameters P_d of the depthwise convolution can be expressed as follows,

$$P_d = F \times F \times C_i. \tag{5}$$

The computational volume C_d of the depthwise convolution can be expressed as follows,

$$C_d = F \times F \times C_i \times W_o \times H_o. \tag{6}$$

Next is the part of pointwise convolution, the function of pointwise convolution is mainly through the use of 1×1 convolution kernel to upgrade or downgrade the feature map, so as to merge the messages between the channels. The shape of the convolution kernel is $1 \times 1 \times C_i$ and the number of convolution kernels C_o is the number of channels in the output feature map. The number of parameters P_p for pointwise convolution can be expressed as follows,

$$P_p = 1 \times 1 \times C_i \times C_o. \tag{7}$$

The computational volume C_P for pointwise convolution can be expressed as follows,

$$C_P = 1 \times 1 \times W_o \times H_o \times C_i \times C_o. \tag{8}$$

The parametric quantity P_a of the depthwise separable convolution can be expressed as follows,

$$P_a = F \times F \times C_i + 1 \times 1 \times C_i \times C_o. \tag{9}$$

Table 1
Comparison of depthwise separable convolution examples.

Method	Parameters	Computational volume
DSConv	411	14796
Conv	3456	124416

The formula for its calculation quantity C_a can be expressed as follows,

$$C_a = F \times F \times C_i \times W_o \times H_o + 1 \times 1 \times W_o \times H_o \times C_i \times C_o. \tag{10}$$

We perform an example algorithm using the above formulas, and the algorithm results are shown in Table 1.We hypothetically assume that the size of the input feature map is $8 \times 8 \times 3$, the kernel size is 3×3, the step size is 1, there is no padding, and there is no bias. The size of the output feature map is $6 \times 6 \times 128$. From the table, we can find that both the number of parameters and the amount of computation, the value of the depthwise separable convolution is much smaller than that of the conventional convolution. It can be proved that the depthwise separable convolution can effectively reduce the computational cost compared with the conventional convolution.

The structure of the lightweight convolution module is shown in Figure 12. As can be seen from the figure, we do not use traditional convolution in the feature extraction stage, but a more efficient depthwise separable convolution. At the same time, we also use 1×1 convolution operation to reduce dimension, so as to further reduce the number of parameters

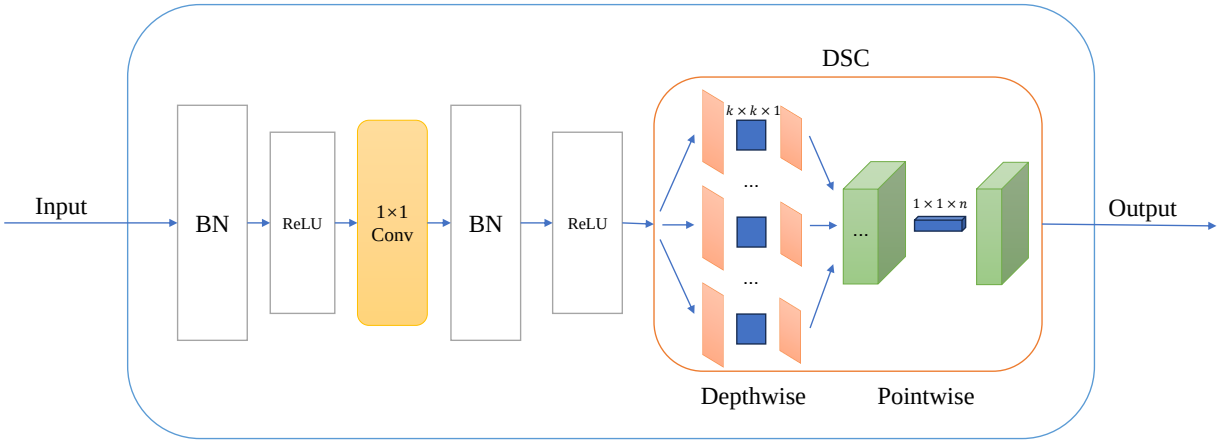


Figure 12: Structure of the lightweight convolution module.

in the model. We first use the BN layer to normalize the features, then pass the obtained feature data through the ReLU activation function, and then use 1×1 convolution operation reduces the dimensionality of features. The ReLU activation function is expressed as,

$$ReLU(x) = \max\{0, x\}. \tag{11}$$

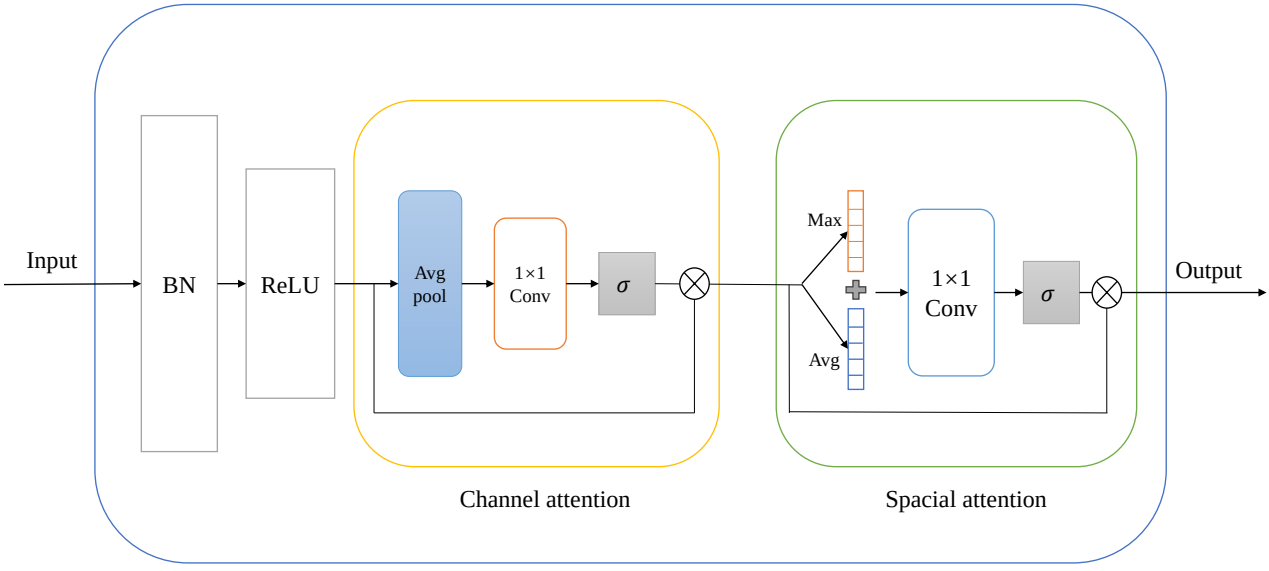
Then the feature data is also passed through the BN layer and the ReLU activation function, and the feature extraction is performed using the depthwise separable convolution. The feature data is first deeply convolved, and the operation of the convolution kernel does not change the dimension of the input feature. After the depthwise convolution operation, the 1×1 convolution kernel is used for pointwise convolution to change the feature dimension and finally output the extracted feature.

4.2. Efficient Hybrid Attention Module

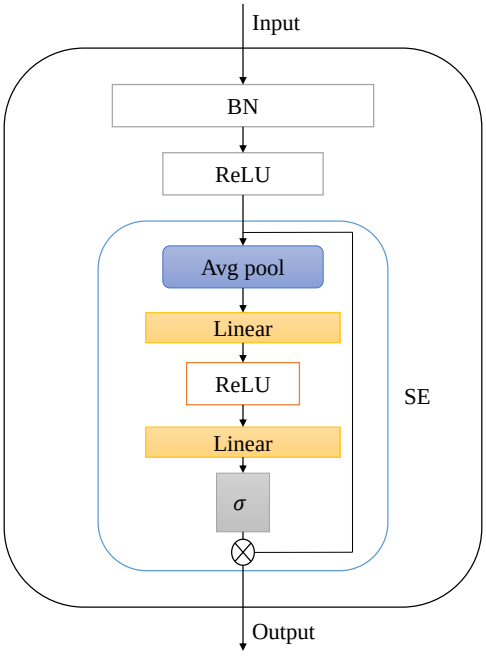
However, relying solely on convolution does not achieve good results in performing feature learning for 5G network traffic. The emergence of the attention mechanism allows the model to adaptively select and weight the importance of features to further improve the performance of the model. We add channel attention and spatial attention after the lightweight convolution module, and redistribute the weights of the features extracted by convolution through the attention mechanism, so that the model focuses on the regions with higher contribution values.

We consider the use of efficient hybrid attention mechanisms to enhance the performance of the model. The channel attention mechanism is responsible for paying attention to the relationship between different channels in the feature map, and can dynamically adjust the importance of different channels to better capture key feature information between different channels. The spatial attention mechanism is responsible for focusing on the relationship between different spatial locations in the feature map, and dynamically adjusting the importance of different channels to have higher attention weights on the more important regions. The weight of key features in 5G data is enhanced through hybrid attention, so as to better learn the characteristics of 5G data in the spatio-temporal domain. The convolutional block attention module (CBAM) proposed by Woo et al. [29], combines the two types of attention mechanisms to improve the model performance significantly. However, the channel attention mechanism in CBAM uses a large number of fully connected layers, which increases the number of parameters and computation cost of the model. To achieve a lightweight model, we use efficient channel attention (ECA) as the channel attention mechanism in the EHA, which reduces the amount of parameter computation by using a one-dimensional convolution. After the channel attention processing, we use the spatial attention mechanism to enhance the weight of key features. In the final stage of feature learning, we use SEBlock to adjust the importance of different channels, so that the model can better capture the key features.

The structure of the efficient hybrid attention module is shown in Figure 13. Where (a) represents the attention mechanism for key feature enhancement after the convolution operation, and (b) represents the attention mechanism



(a)



(b)

Figure 13: Structure of the efficient hybrid attention module.

for adjusting the importance of channels after the entire feature learning stage. We call these two parts efficient hybrid attention mechanism. We take the feature map output from the convolutional layer as the input to the hybrid attention module $X_i^{(h,w)}$. First, a normalization operation is performed on it and the ReLU activation function is used to obtain the feature map $X_p^{(h,w)}$. The computation is expressed as follows:

$$X_p^{(h,w)} = ReLU(BN(X_i^{(h,w)})). \tag{12}$$

Subsequently, the feature map $X_p^{(h,w)}$ enters the efficient channel attention, and after average pooling, one-dimensional convolution, and sigmoid operations to obtain the channel weights $W_c^{(h,w)}$, multiply the weights $W_c^{(h,w)}$ with the input

$X_p^{(h,w)}$ to obtain the output $X_c^{(h,w)}$. Which is expressed as:

$$X_c^{(h,w)} = W_c^{(h,w)} \cdot X_p^{(h,w)}. \tag{13}$$

Finally, the feature map $X_c^{(h,w)}$ enters into the spatial attention part, which is respectively subjected to maximum pooling and average pooling operations and spliced, the spliced data is input into the convolution kernel size of 7×7 convolution, and finally a sigmoid operation is performed to get the spatial weights $W_s^{(h,w)}$. multiply the weights $W_s^{(h,w)}$ with the input $X_c^{(h,w)}$ to get the output $X_f^{(h,w)}$. The computation is expressed as follows:

$$X_f^{(h,w)} = W_s^{(h,w)} \cdot X_c^{(h,w)}. \tag{14}$$

At the end of feature learning, we pass the input feature $X_r^{(h,w)}$ through BN layer and ReLU activation function to get $X_a^{(h,w)}$. Which is expressed as:

$$X_a^{(h,w)} = ReLU(BN(X_r^{(h,w)})), \tag{15}$$

and then input it into channel attention. First, the feature data is processed with average pooling, then it passes through the linear layer, then ReLU activation function is used to process it, and then through a linear layer, it is entered into the sigmoid function to obtain the weight $W_o^{(h,w)}$. Finally, multiply $X_a^{(h,w)}$ by $W_o^{(h,w)}$ to get $O_f^{(h,w)}$, which is expressed as:

$$O_f^{(h,w)} = W_o^{(h,w)} \cdot X_a^{(h,w)}. \tag{16}$$

The efficient hybrid attention module makes the processing of 5G network traffic data more targeted, with the model giving higher weights to key data as a way to improve the model's prediction accuracy. It also has lower parameters and computation costs while ensuring predictive performance.

5. Experiments

This section shows the experimental part in detail. First, we describe the data preprocessing procedure and the setting of experimental parameters. And we introduce the evaluation indexes. Then, we compare and analyze the performance of the method proposed in this paper with previous prediction models and conduct ablation experiments. We also conduct experiments to compare the parameters and computational quantities. Finally, we analyze the comparison of the predicted results.

5.1. Data Pre-processing and Parameter Selection

The dataset we used is an open dataset provided by Telecom Italia. The original dataset has a sampling interval of 10 min, and such a sampling interval leads to a lower efficiency of the whole experimental process and a higher overhead of the prediction model [18]. Together with the fact that a large amount of traffic data in this dataset has a value of 0, we therefore aggregate the traffic data on an hourly basis. We use the last 168 hours of data for model prediction and the remaining portion of the dataset for model training. In fact, due to the lack of historical data for the first 72 hours of training, the actual available training dataset is 1248 hours after 72 hours. In addition, in this paper, sigmoid is used for the activation output of the model, so min-max is used for normalization to scale the flow into [0, 1], and then finally the predicted values are rescaled back to normal values for evaluation.

The environment used for the experiments in this paper is python 3.9.16, numpy 1.23.5, pytorch 1.13.1. The basic hardware configurations of the experimental platforms are AMD R7-6800H, NVIDIA RTX 3060, and 16GB of memory. The optimizer used for the model is ADAMW [30], which is an improved version of the ADAM [31] optimizer. It improves the performance and robustness of the model by introducing a weight decay mechanism to better control the update of weights. The initial learning rate is set to 0.01 and then decays as the epoch increases. In the LC layer, a filter with a kernel size of 1×1 is used before the depthwise separable convolution operation, a depthwise convolution filter has a kernel size of 3×3, and a pointwise convolution filter has a kernel size of 1×1. In the hybrid attention layer, average pooling and a filter with a kernel size of 7×7 are used, and the activation functions are sigmoid and ReLU.

5.2. Evaluation Indicators

In this paper, two evaluation metrics are used to assess the model, root mean squared error (RMSE) and mean absolute error (MAE).

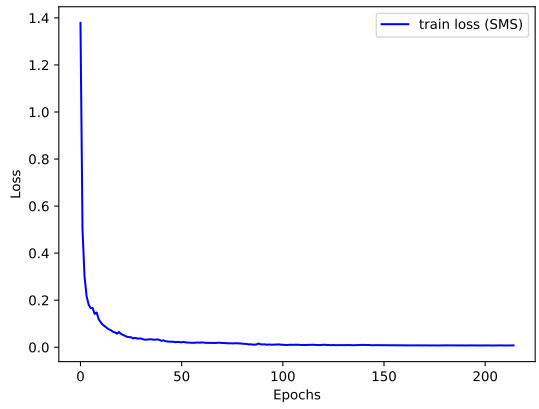
RMSE stands for Root Mean Square Error and is a commonly used metric to assess the performance of regression models. It is used to measure the difference between the predicted values and the actual observed values. RMSE is suitable for assessing the performance of a regression model in a continuous numerical prediction task and its formula is shown below:

$$RMSE = \sqrt{\frac{1}{X \times Y} \sum_{x=1}^{X} \sum_{y=1}^{Y} (\hat{P}^{(x,y)} - P^{(x,y)})^2}. \tag{17}$$

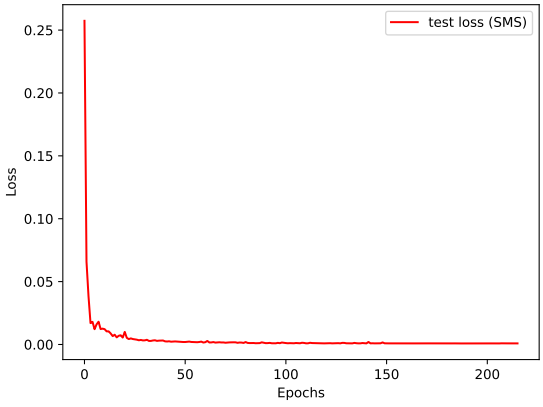
MAE stands for Mean Absolute Error, which is a commonly used metric for assessing the performance of regression models. Unlike RMSE, MAE does not involve squaring operations, but rather calculates the mean of the absolute differences between the predicted values and the actual observed values. Therefore MAE is more robust and insensitive to outliers, and thus can better reflect the average difference between predicted and actual values.The formula for MAE is shown below:

$$MAE = \frac{1}{X \times Y} \sum_{x=1}^{X} \sum_{y=1}^{Y} \left| \hat{P}^{(x,y)} - P^{(x,y)} \right|. \tag{18}$$

Where $\hat{P}^{(x,y)}$ is the predicted value of the model and $P^{(x,y)}$ is the true value.

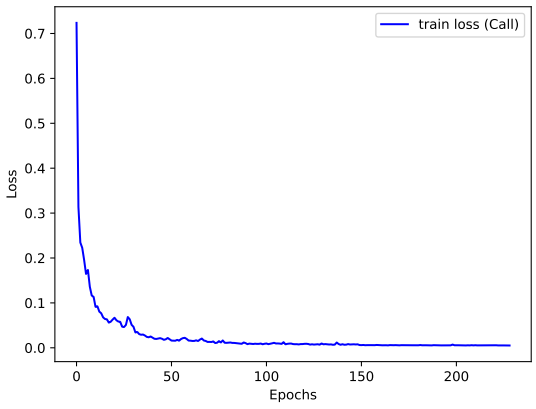


(a)

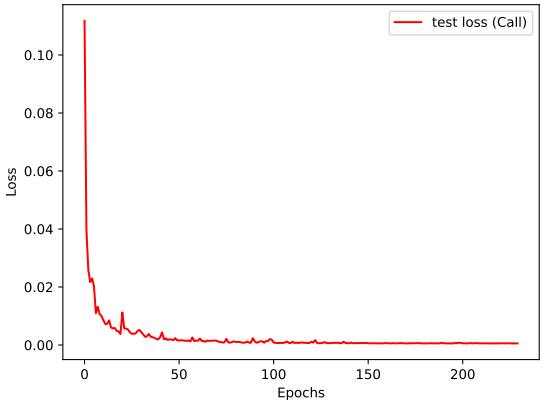


(b)

Figure 14: Loss variation (SMS).



(a)



(b)

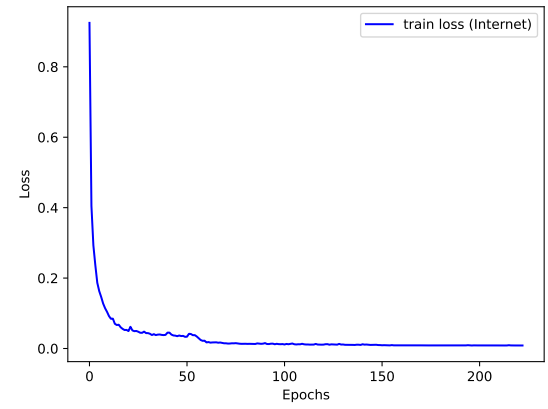
Figure 15: Loss variation (Call).

5.3. Overall Performance

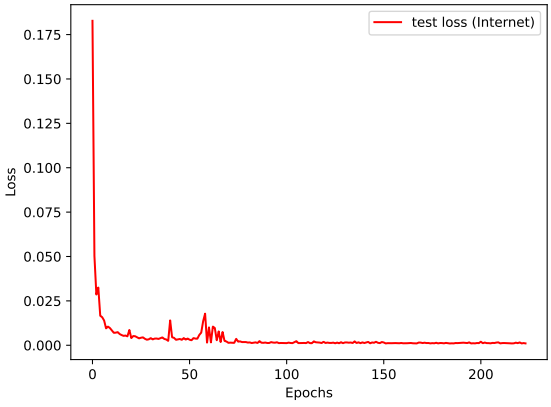
Before exploring the performance of the model we analyze the change in the value of the error loss between the predicted and true values of the model with successive meta-iterations as a way of verifying whether the model converges or not. We train and test the model on SMS, Call, and Intenet datasets respectively, and the results are shown in Figure 14, Figure 15 and Figure 16. It can be seen from the figure that on the SMS dataset the training loss plateaus around 50 epochs and the training loss converges around 25 epochs. However, on the Call dataset the convergence is significantly slower, while on the Internet dataset the training loss oscillates slightly around 60 epochs, but converges quickly. This shows that the model is able to reach the convergence state very quickly, thus effectively verifying the reliability of the model.

In order to verify the prediction performance of the model proposed in this paper, we conduct comparative experiments on the models. The datasets used for the experiments are SMS, Call and Internet, and the evaluation indicators are RMSE and MAE. The experiments are conducted on the lightweight hybrid attention model proposed in this paper as well as some of the existing network traffic prediction models including HA, ARIMA, RNN, LSTM [32], STDenseNet [18] and HSTNet [19]. The results of the experiments are shown in Table 2, Table 3 and Table 4.

From the comparison of experimental results, our model outperforms the existing prediction models on all three datasets. The historical average method (HA) only calculates the average of the existing historical data and lacks the ability to extract more feature correlations from the data itself. ARIMA only considers the linear relationships in the data, which leads to poor results. The above statistical methods are unable to achieve the desired prediction effect due to their own limitations. Deep learning methods such as RNN and LSTM are able to deal with temporal data sequences better, but it is difficult to deal with the spatial correlation of the 5G traffic data. STDenseNet and HSTNet are able to combine temporal correlation and spatial correlation, But they cannot capture the key features well in the feature learning phase. Our method better captures the key features in the spatio-temporal domain through an efficient hybrid attention mechanism (EHA), so it can get better results.



(a)



(b)

Figure 16: Loss variation (Internet).

Table 2
Results of model performance comparison (SMS).

Dataset	approach	RMSE	MAE
SMS	HA	49.2654	27.0625
	ARIMA	38.1265	24.7264
	RNN	33.2232	20.3322
	LSTM	30.2165	17.9926
	STDenseNet	24.9297	15.2868
	HSTNet	24.6212	14.9012
	Ours	**23.9793**	**14.7603**

Table 3
Results of model performance comparison (Call).

Dataset	approach	RMSE	MAE
Call	HA	37.8628	21.6432
	ARIMA	32.2148	19.0067
	RNN	25.0098	15.6689
	LSTM	21.6203	11.2540
	STDenseNet	15.1990	10.6756
	HSTNet	14.0011	9.7205
	Ours	**12.7006**	**8.5299**

Table 4
Results of model performance comparison (Internet).

Dataset	approach	RMSE	MAE
Internet	HA	378.7994	267.1134
	ARIMA	236.2517	187.2671
	RNN	210.3321	141.0089
	LSTM	197.6521	132.1242
	STDenseNet	184.2934	125.4735
	HSTNet	156.0027	99.9223
	Ours	**147.2199**	**92.3762**

Table 5
Performance comparisons (SMS).

model	RMSE	MAE
STDenseNet	24.9297	15.2868
+DSConv	24.1040	14.8707
+EHA	24.4815	15.1758
+DSConv+EHA	23.9793	14.7603

Table 6
Performance comparisons (Call).

model	RMSE	MAE
STDenseNet	15.1990	10.6756
+DSConv	15.0626	10.6217
+EHA	12.8598	8.7914
+DSConv+EHA	12.7006	8.5299

Meanwhile, in order to explore the key factors affecting the performance of the model, we conduct ablation experiments and the results are shown in Table 5, Table 6, Table 7. where DSConv stands for depthwise separable convolution and EHA stands for efficient Hybrid Attention. On the SMS dataset, DSConv provides a large model boost, while EHA provides a relatively small model boost. There is a more significant improvement in model performance when using both DSConv and EHA. On the Call dataset, the performance improvement of DSConv is not significant, but rather EHA brings a larger performance improvement. The best model performance is achieved when both can be used. On the Internet dataset, the improvement of EHA is more obvious. Similarly, the model's prediction is best when both DSConv and EHA are used. The above experiments show that efficient hybrid attention can effectively perform weight enhancement for key features, thus improving the predictive performance of the model. And depthwise separable convolution can also improve the performance of the model to some extent.

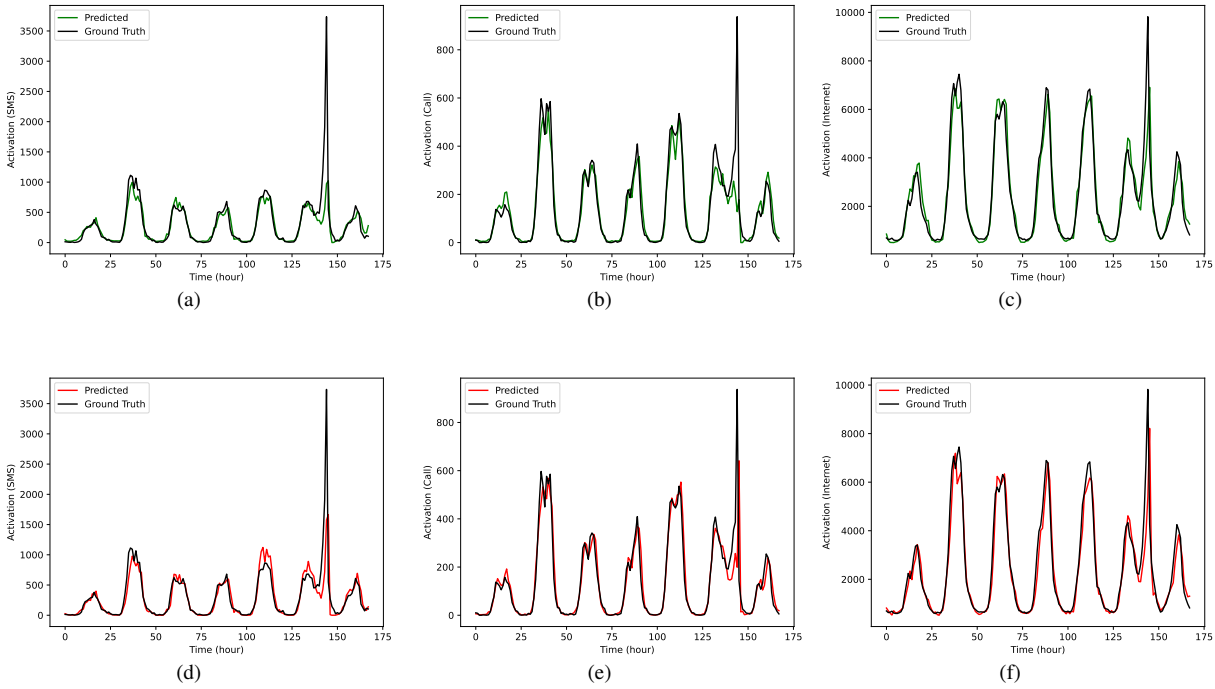


Figure 17: Comparison of prediction results for cells (10, 19).

Table 7
Performance comparisons (Internet).

model	RMSE	MAE
STDenseNet	184.2934	125.4735
+DSConv	170.9484	117.3138
+EHA	148.9835	93.9044
+DSConv+EHA	147.2199	92.3762

We also conduct a comparison experiment between the prediction results of the proposed model and the baseline model. We randomly select cell (10, 19) and cell (14, 8) to conduct experiments on three datasets respectively. The comparison of experimental results is shown in Figure 17 and Figure 18. Where (a), (b) and (c) show the prediction results of the baseline model on three datasets and the prediction curves are represented in green, while (d), (e) and (f) show the prediction results of the model proposed in this paper on three datasets and the prediction curves are represented in red. It can be seen that the method proposed in this paper is more consistent with the ground truth than the baseline model in most cases. This shows that the model can capture key features in 5G data well, so as to obtain good prediction results.

5.4. Parameters and Computational Volume Analysis

We analyze the parameters and computational quantities of the model on three datasets, the results are shown in Table 8, Table 9 and Table 10. The influence of different modules on the lightweight of the model is analyzed in detail. According to the table, when LC layer is used alone to replace the original convolution layer, the model has a significant reduction in the number of parameters and calculation consumption compared with the basic model. It can be seen that the lightweight convolution layer can greatly improve the lightweight property of the model, thereby reducing the prediction cost. When we add the EHA layer alone, the parameters and computational costs of the model increase only slightly compared to the baseline model, with some lightweight optimization of our attention. Combined with the previous ablation experiments, we believe that it is worthwhile to sacrifice a small amount of lightweight properties in exchange for the improved predictive performance of the model. Moreover, our model still has a huge advantage over the baseline model in terms of lightweight. It can be seen that the model proposed in this paper can effectively save computational resources.

6. Conclusion

This paper focuses on network traffic prediction based on deep learning methods for 5G. A lightweight hybrid attention deep learning model is proposed for the prediction performance of the model and the lightweight of the model. In this paper, we use hybrid attention that integrates the channel

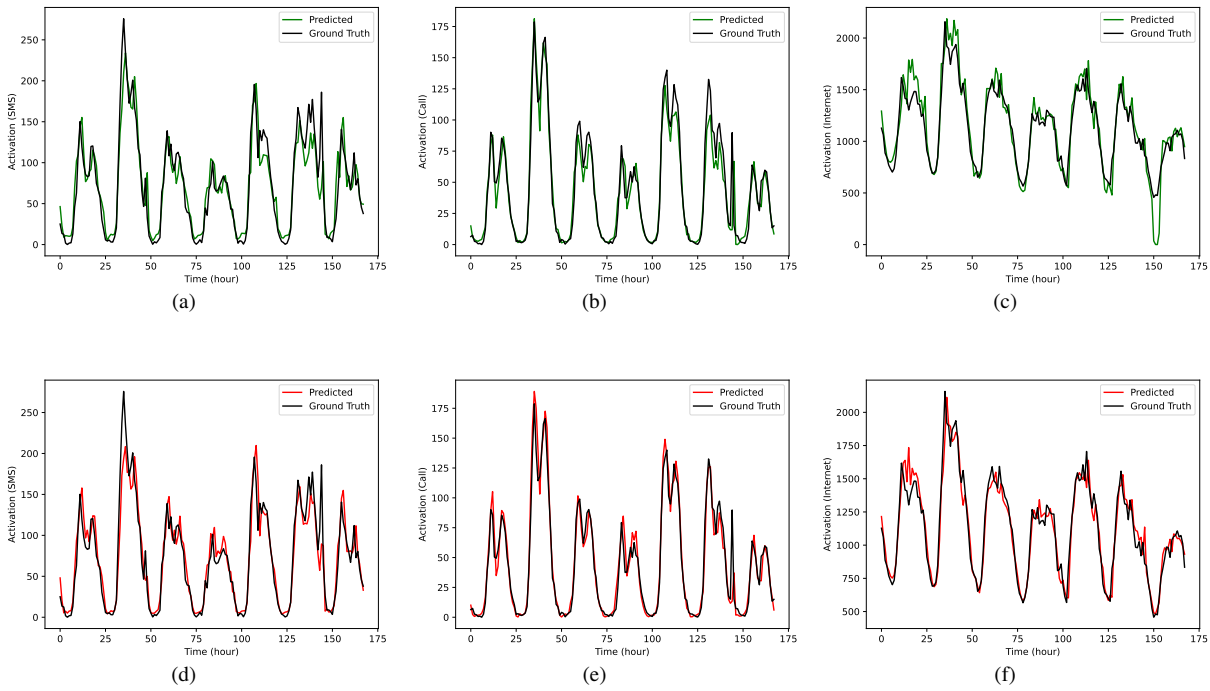


Figure 18: Comparison of prediction results for cells (14, 8).

Table 8
Comparison of Computing Resource Consumption (SMS).

model	Parameters	FLOPs
STDenseNet	0.29 M	1899.11 M
+LC	0.12 M	774.76 M
+EHA	0.30 M	1922.90 M
Ours	0.13 M	798.55 M

Table 9
Comparison of Computing Resource Consumption (Call).

model	Parameters	FLOPs
STDenseNet	1.90 M	12215.91 M
+LC	0.61 M	3974.76 M
+EHA	1.93 M	12265.74 M
Ours	0.65 M	4024.59 M

attention mechanism with the spatial attention mechanism as a way to enhance the feature learning capability of the model. Lightweight operations such as depthwise separable convolution are also used to further reduce the prediction cost of the model. The experimental results show that on three real network traffic prediction datasets, the lightweight hybrid attention deep learning model proposed in this paper has more accurate prediction performance as well as lower prediction cost compared to existing methods. This is

Table 10
Comparison of Computing Resource Consumption (Internet).

model	Parameters	FLOPs
STDenseNet	0.50 M	3228.06 M
+LC	0.18 M	1204.63 M
+EHA	0.51 M	3257.03 M
Ours	0.20 M	1233.61 M

sufficient to demonstrate the value of the prediction model proposed in this paper in 5G network traffic prediction. Relying on the model's accurate network traffic prediction ability to rationalize the allocation of resources, thus solving the communication efficiency and resource consumption problems in the field of integrated sensing, communication and computation (ISCC) to a certain extent.

Our work still needs continuous improvement, and the model's prediction ability is lacking when dealing with sudden 5G network traffic data. In subsequent work, while improving the performance of the model itself, more characteristics of the network traffic data and the factors affecting the prediction can be incorporated. Alternatively, the model can be targeted to optimize a certain aspect of the model, thus making it easier to meet the needs of specific scenarios.

References

[1] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6g wireless systems: Applications, trends, technologies, and open research

problems. *IEEE network*, 34(3):134–142, 2019.

[2] Xiaoyang Li, Yi Gong, Kaibin Huang, and Zhisheng Niu. Over-the-air integrated sensing, communication, and computation in iot networks. *IEEE Wireless Communications*, 30(1):32–38, 2023.

[3] Fan Liu, Yuanhao Cui, Christos Masouros, Jie Xu, Tony Xiao Han, Yonina C Eldar, and Stefano Buzzi. Integrated sensing and communications: Toward dual-functional wireless networks for 6g and beyond. *IEEE journal on selected areas in communications*, 40(6):1728–1767, 2022.

[4] Guangxu Zhu, Zhonghao Lyu, Xiang Jiao, Peixi Liu, Mingzhe Chen, Jie Xu, Shuguang Cui, and Ping Zhang. Pushing ai to wireless network edge: An overview on integrated sensing, communication, and computation towards 6g. *Science China Information Sciences*, 66(3):130301, 2023.

[5] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] H Zare Moayedi and MA Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *2008 international symposium on information technology*, volume 4, pages 1–6. IEEE, 2008.

[7] Bo Zhou, Dan He, Zhili Sun, and Wee Hock Ng. Network traffic modeling and prediction with arima/garch. In *Proc. of HET-NETs Conference*, pages 1–10, 2005.

[8] Yanhua Yu, Jun Wang, Meina Song, and Junde Song. Network traffic prediction and result analysis based on seasonal arima and correlation coefficient. In *2010 International Conference on Intelligent System Design and Engineering Application*, volume 1, pages 980–983. IEEE, 2010.

[9] Rongpeng Li, Zhifeng Zhao, Jianchao Zheng, Chengli Mei, Yueming Cai, and Honggang Zhang. The learning and prediction of application-level traffic data in cellular networks. *IEEE Transactions on Wireless Communications*, 16(6):3899–3912, 2017.

[10] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

[11] Balaji Krithikaivasan, Yong Zeng, Kaushik Deka, and Deep Medhi. Arch-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic. *IEEE/ACM Transactions on networking*, 15(3):683–696, 2007.

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[16] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[17] Shan Jaffry and Syed Faraz Hasan. Cellular traffic prediction using recurrent neural networks. In *2020 IEEE 5th international symposium on telecommunication technologies (ISTT)*, pages 94–98. IEEE, 2020.

[18] Chuanting Zhang, Haixia Zhang, Dongfeng Yuan, and Minggao Zhang. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Communications Letters*, 22(8):1656–1659, 2018.

[19] Dehai Zhang, Linan Liu, Cheng Xie, Bing Yang, and Qing Liu. Citywide cellular traffic prediction based on a hybrid spatiotemporal network. *Algorithms*, 13(1):20, 2020.

[20] Maryam Mohseni, Soodeh Nikan, and Abdallah Shami. Ai-based traffic forecasting in 5g network. In *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 188–192. IEEE, 2022.

[21] Zheheng Rao, Yanyan Xu, Shaoming Pan, Jiabao Guo, Yuejing Yan, and Zhiheng Wang. Cellular traffic prediction: A deep learning method considering dynamic nonlocal spatial correlation, self-attention, and correlation of spatiotemporal feature fusion. *IEEE Transactions on Network and Service Management*, 20(1):426–440, 2022.

[22] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2(1):1–15, 2015.

[23] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020.

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[29] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Chen Qiu, Yanyan Zhang, Zhiyong Feng, Ping Zhang, and Shuguang Cui. Spatio-temporal wireless traffic prediction with recurrent neural network. *IEEE Wireless Communications Letters*, 7(4):554–557, 2018.