A Multi-head ConvLSTM for Healthcare Time Series Classification

Le Sun，Hui Liu

Abstract: Healthcare Time Series Classification is a learning classification of collected human physiological information based on artificial intelligence deep learning. The main purpose is to use deep learning to enable machines to learn and analyze the same characteristics as humans, analyze various physiological information of the human body, make correct judgments, and reduce medical costs. Healthcare time series classifications are specifically used, for example, to treat electrocardiograms (ECGs) to diagnose cardiovascular disease, to treat brain waves (EEG) to diagnose brain diseases, and to process human motion signals to alert to abnormal and dangerous physical actions. For these purposes, we propose the Multiple-head Convolutional LSTM (MCL) model for Healthcare Time Series Classification. MCL is a ConvLSTM model with Multiple_heads. It can extract both time and spatial features of healthcare data and increase the number of features to achieve more accurate classification.

Keywords: A Multi-head ConvLSTM deep learning model, healthcare Time Serie Classification; ECG classification; Sleep-EDF classification; Human Activity Recognition classification

1.Introduction

The field of medicine is one of the main directions of artificial intelligence deep learning application. We can speculate on the probability of people getting sick based on known information, trace the source of the disease through big data, learn deep to determine if they are ill, and give warning to people to go to the hospital for detailed testing.

Much has been done to analyze the Healthcare timing based on artificial intelligence technology. M Irfan,IA Hameed et al. proposed implementing Deep Summary Neural Networks (DCNN) to solve the problem of classifying image pattern data extracted from electrocardiograms (ECGs) [1]. Morghan Hartmann; Hasan Farooq; Ali Imran proposes combining 5G and AI and using machine learning to apply medical applications such as medicine [4], and so on. Those work all promote the application of artificial intelligence deep learning in the field of medicine, and effectively promote the development of modern medicine.

Through these workers' research, we find that the improvement of the accuracy of Healthcare Timing Classification has always been a focus and difficulty in academic research. Fortunately, a great deal of work has been done on this issue, such as M. Irfan's Deep Learning for Healthcare data system [1]. A Guo et al. use deep learning LSTM and 5x cross-validation models to predict heart disease [11].

One of the main problems that makes Healthcare Timing Classification accuracy difficult to improve is category imbalance. Category imbalance means that the number of training samples for different categories varies too much. There is a widespread imbalance in medical data. We need to target it. TensorFlow Core suggests that category imbalances can be addressed with data enhancement and standardization. This paper also puts forward a method to solve the category imbalance, which includes normalizing the data, trimming the normalized data and adjusting the weight of the sample label [9]. In addition, we have used L2 regularization in training to address overfitting issues caused by category imbalances.

In order to improve the accuracy of Healthcare Timing Classification, an MCL model is proposed to implement Healthcare Timing Classification.

The main purposes of this article are as follows:

(1) The method of normalization, pruning and adjusting the weight of sample labels is proposed to solve the problem of category imbalance.

(2) MCL model for high-accuracy Healthcare Timing Classification

(3) Based on Human Activity Har, MIT-BIH, Sleep-EDF.A great deal of experiments have been done to verify the effectiveness of our method.

The structure of this article is as follows:

The second chapter introduces the relevant work.

Chapter 3 describes the proposed Multiple-head Convolutional LSTM (MCL) model structure.

The fourth chapter introduces the experimental process and results.

Chapter 5 summarizes the article and puts forward the future research direction.

2.Related work

2015 Y Bar et al. used CNN to train neural networks for chest X-ray images to classify different types of pathology [16]. 2017 M Rezaei et al. use deep learning for end-to-end training of medical images [17]. In 2017, M Irfan et al. proposed a deep summary neural network (DCNN) to classify electrocardiograms [1]. 2018 DS Kermany et al. use transfer learning to conduct training analyses of scanned images for medical diagnosis and treatment of diseases [15]. Using residual networks and end-to-end learning methods, such as DBian2020 use PPG signals to estimate respiratory rate (RP) [20]。 2020 DH Kisa et al. proposed the use of a 101-layer converged neural network based on the Residual Network (ResNet) architecture to identify movements and gestures for purpose [2]. 2021 A Guo et al. demonstrated the advantages of LSTM by using the deep learning lstm model and predicting heart disease through 5x cross-validation, and comparing the model with other deep neural networks [11]. In 2020, N Sridhar et al. used deep learning models to analyze instantaneous heart rate extracted from electrocardiograms and classify the sleep phase [13]. In 2021, GQ Ali et al. used a deep, aggregated neural network architecture to identify human activity data sets (HAR) and achieve 99.8% overall accuracy [12]. In 2021, A Alharbi et al. used RNN, LSTM, GRU, BL_LSTM techniques to build models to compare heart rate time series in MIMIC-II and conclude that a three-tierED GRU can achieve the best experiment results [18]. 2021 D Suryani et al. used Musk R-CNN to construct a model to automatically detect objects in X-ray images to improve the accuracy of radiation detection [19].

In 2019, J Goschenhofer et al. used a grade-weighted approach in the motion status testing model for people with Parkinson's disease, successfully alleviating the problem of high imbalance in patient movement data [14]. In 2020, AM Shaker et al. used the Build-Adversarial Network to balance the data set in proposing two deep learning methods and a deep summary neural network (CNN) hierarchical approach to electrocardiogram classification [5]. 2020 TFR, etc. proposes to use focus loss to address data imbalances [6]. TensorFlow Core suggests that data imbalances can be addressed, for the most part, such as data enhancement and standardization [9].

In this article, a Multiple-head Convolutional LSTM (MCL) model is proposed, which combines the characteristics of CNN and LSTM. The problem of information loss encountered in LSTM processing is effectively solved. And in a Multi_head way, the model can operate simultaneously using convolution cores of different sizes, which can provide better performance. In addition, we have tried to solve the problem of data imbalance by using normalization processing and trimming data. This method effectively increasesthe accuracy of the MCL model. In this article, mcL performance is demonstrated by using three datasets: Human Activity Identity (HAR), MIT-BIH, Sleep-EDF, and comparing them to general deep learning neural networks.

3 Multiple-head Convolutional LSTM

The Multiple-head Convolutional LSTM (MCL) model is ConvLSTM with multiple_heads Model, each One_head ConvLSTM of the model can read input time steps using convolution cores of different sizes. For example, in Lab 1, we used the MCL model of Three_head, and for three different One_head ConvLSTM, we used three Different convolution cores (1,3), (1,5), (1,11). This allows the model to read and interpret input data in three different filter sizes [8]. The three One_head then run the results of ConvLSTM in the merge layer (merge) connection. The full-connection layer (FC) performs the prediction operation.

The MCL architecture uses convolutional neural networks (Convolutional LSTM) in conjunction with LSTM to process input data, perform functional extraction to support sequence prediction, and further expansion, andMCL will cnn Summarized as part of LSTM. Unlike LSTM, which reads data directly to calculate internal state and state transitions, ConvLSTM uses convolutions directly as part of the LSTM's own input [7].

3.1 The proposed Multiple-head Convolutional LSTM

3.1.1. Data pre-processing

The initial data of the experiment had problems with the imbalance of sample classification, and we normalized the data and trimmed the normalized data (for example, set the data greater than 5 to 5 and the value less than -5 to -5), making it easy for us to perform experimental calculations. Normalization processing de-means and scales variances for each feature dimension, so that the processed data conforms to the standard normal distribution. We divide the data into training sets and test sets. The training set is normalized and trimmed, and the test set is normalized and trimmed using the mean and variance obtained by the training set, so as not to interfere with the test set and make false experimental results. The formula for normalization processing is (1). Where X is the initial input and X^* is the data after normalization, μ is the mean of the training set sample data, σ is the standard deviation of the training set sample data.

$$X^* = \frac{X-\mu}{\sigma} \quad (1)$$

Data that is balanced is a three-dimensional data $X = [x_1, x_2, \cdots x_n\ ]^T$ shaped (sampleNum, timesteps, channel). The first sample, the second sample, and the nth sample are thefirst, and each sample has a shape of$x_1, x_2, \cdots x_n$(1, timesteps, channel), and sampleNum is the sample number Timesteps is the time series for each sample, andchannel is the number of channels that the sample data has. In our proposed MCL model, ConvLSTM has convolution structures from incoming to state and state-to-state transitions, which requires us to expand the dimensions. For example, we took ConvLSTM2D as the ConvLSTM of MCL in the processing of a pair of Activity Activity(HAR) datasets layer, then we need to reshape 3D data (sampleNum, timesteps, channel) into 5D data (sampleNum, time, row, cols, channel). In MCL we keep the sample and channelof the input data, reshape of the timesteps 1D data for 3D data (time, row, col), time is timing input, row and col are rows and columns, respectively [7]. When we process data, we make timesteps of all samples the same shape to avoid losing chronological order during the reshape process. The data preprocessing steps are shown in Figure 1.

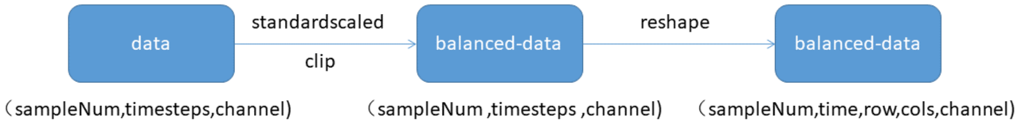

data
standardscaled
clip
balanced-data
reshape
balanced-data
（sampleNum,timesteps,channel)
（sampleNum ,timesteps ,channel)
（sampleNum,time,row,cols,channel)

Fig 1 data preprocessing

1

3.2 Build MCL

3.2.1. ConvLSTM

CNN uses convolutional operations to extract spatial features, LSTM uses memory units and doors to extract time characteristics, and ConvLSTM combines the characteristics of CNN and LSTM to take advantage of both space-time characteristics. At the heart of ConvLSTM is convolution of memory units and doors. Memory unit. C_t ConvLSTM acts as an accumulator for status information. The door is controlled by adaptive training parameters. Whenever there is a new input, the information that the input door i_t controls it accumulates in the memory unit C_{t-1}. Forget the door f_t control the cell state of the past C_{t-1} will it be "forgotten" during this procession. The output gate o_t controls whether the final unit output will be entered to the final state [10]. In general LSTM, it (f_t, o_t, i_t) is a 1D vector that loses information when processing space-time data. In ConvLSTM, which is a 3Dvector [10]. With convolution operations, ConvLSTM not only gets time-to-time features, but also extracts spatial features like convolution layers. Datacollected by devices in the H ealth care timing classification in time intervals has a single time series with no spatial properties. We change the shape of the experimental data by preprocessing, giving the data spatial properties. Time and spatial features are extracted at the ConvLSTM layer of the MCL model, after which both time and spatial features are interpreted and classified at the full connection layer. Figure 2 shows the internal structure of ConvLSTM. H is a hidden state, a cell state, $C_t\chi_t$ is input data. [10]

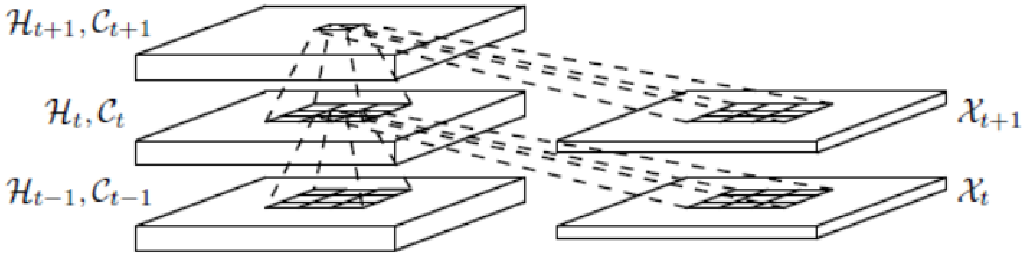

$\mathcal{H}_{t+1}, \mathcal{C}_{t+1}$
$\mathcal{H}_t, \mathcal{C}_t$
$\mathcal{H}_{t-1}, \mathcal{C}_{t-1}$
$\mathcal{X}_{t+1}$
$\mathcal{X}_t$

Fig 2: inner structure [10]

The key equation of ConvLSTM (2). H is a hidden state, W is a filter, i_t indicates an input door, χ_t is the input, f_t indicates a forgotten door, C_t is a cell state, o_t is an output door, and b is a bias. Where '*' means convolution operator, '∘' represents Hadamard product [10].

$$i_t = \sigma(W_i * \chi_t + W_i * H_{t-1} + W_{ci} \circ C_{t-1} + bi)$$

$$f_t = \sigma\left(W_f * \chi_t + W_f * H_{t-1} + W_{cf} \circ C_{t-1} + bf\right)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_c * X_t + W_c * H_{t-1} + b_c) \qquad （2）[10]$$

$$o_t = \sigma(W_o * \chi_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + bf)$$

$$H_t = o_t \circ \tanh(C_t)$$

3.2.2 One_head ConvLSTM

A standard One_head ConvLSTM model consists of four layers: Input, ConvLSTM, Dropout, and Flatten (in order). We first enter the data input layer, which preprocesses the data, and records our data for next training. The ConvLSTM layer is the core layer of our model, which determines the future state of a cell in the grid by its input and past state, and extracts spatial and timeal features through its convolution core and cell state, gates. We use the Correction Linear Unit $f(x) = \max(0, x)$(ReLU) as the ConvLSTM layer activation function. The Dropout layer is randomly inactivated, and we take a random inactivated 50% in this layer to reduce overfitting. The final Flatten layer "flattens" the input data, which is the one-dimensional input of the multidimensional layer for the transition from the convolution layer to the full connection layer. To reduce the degree of overfitting, we regularize L2 for each layer that has data processing. The L2 regularization formula (3), which is a regularized L^*loss function, is the original loss function, and $L$is the sum of the W_i^2 squares of the weight parameters. $\lambda$is a regularized parameter.

$$L^* = L + \lambda \sum_{i=1}^{n} W_i^2 \qquad (3)$$

3.2.3 MCL

In our experiment, we chose Three_head to build the MCL model, Three_head that the actual effect of the model is not too expensive. MCL uses One_head ConvLSTM of different convolution core sizes to extract different space-time features at the same time, and then connects space-time features extracted by different convolution cores at the merge layer. This equates to a sample with more space-time characteristics and more accurate predictions. Therefore, generally speaking, the more head, the higher the accuracy of the model. In an MCL model, each One_head ConvLSTM model can select different superparameters, such as the number of filters, the size of the filters, activation functions, L2 regularization parameters, and so on. The merged layer then passes the resulting data to the full-connected layer (FC). The full connection layer is processed once, fused with the space-time characteristics learned, and then classified by $S_i = \frac{e^i}{\sum_j e^j}$ (softmax). The loss function is a multiclassified loss function(categorical_crossentropy). Figure 3 shows the architecture used by the MCL model in Activity Recogniz (HAR).

Formula 4 Multi-classification loss function (categorical_crossentropy), Where y^{pred} is the predicted quantity and y is the actual quantity.

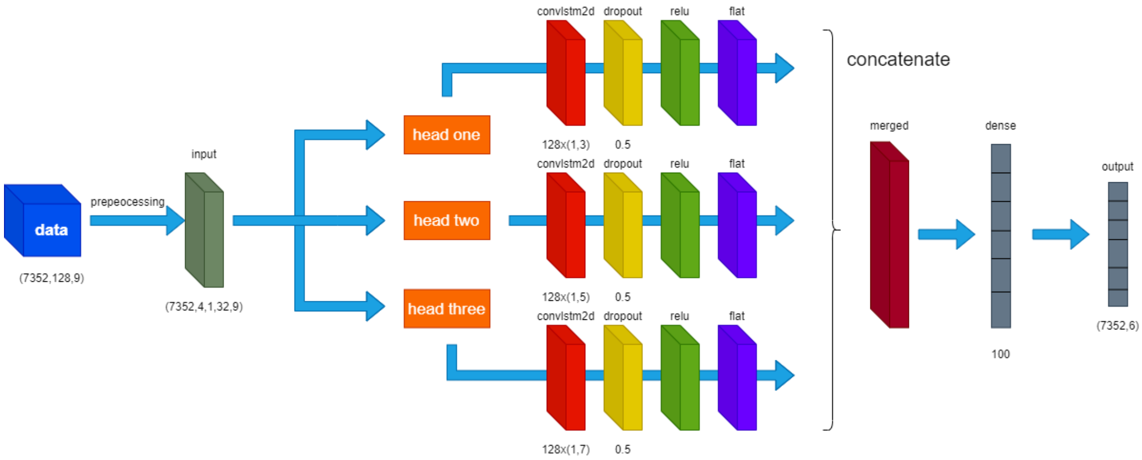
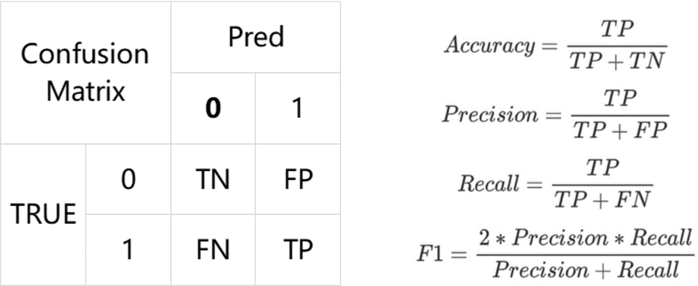


Figure 3：MCL on HAR

$$Loss = -\sum_{i=1}^{size} y_i * log\ y_i^{pred} \qquad (4)$$

3.3 Evaluate the model

Due to the imbalance in the number of sample classifications, in addition to the correct rate, we need more indicators to evaluate the model, so as not to fall into the trap of high accuracy and low practicality. We chose Confusion Matrix to visualize the performance of the model, and in Confusion Matrix we used accracy, precision, recall, F1_score to accurately reflect the performance of the model. Figure 4 shows the mathematical formulas of Confusion Matrix, precision, recall, F1_score.



Confusion Matrix		Pred	
		0	1
TRUE	0	TN	FP
	1	FN	TP

$$Accuracy = \frac{TP}{TP + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Figure 4 Confusion Matrix [22]

3.4 Superparameter

The number of filters (filters) is an important superparameter of ConvLSTM, and in order to extract enough features, the convolution layer should start wide enough. The more complex the problem, the wider the convolution layer, that is, the larger the number of filters. However, sometimes in order to reduce overfitting problems, you can choose a small number of filters for training. The number of filters can be adjusted dynamically. Select the most appropriate quantity after multiple comparisons. The number of filters is generally increased and decreased by multiples. In training we select 16 and 32 as the number of filters.

The size of the filter (kernel_size) is another important superparameter of ConvLSTM, where the kernel size controls the number of time steps considered in each Read in the input sequence and then projected onto the function diagram (via a summary process). Large size means that the reading of data is less rigorous, but generally has a faster training speed. We used (1,3), (1,5), (1,7) and so on as filter sizes in our experiments.

Batch Size is the number of samples selected for a training session. The size of Batch Size affects how optimized and faster the model is. It also has a direct impact on GPU usage, and in general, choosing a larger Batch Size can reduce over-proposed merges and speed up training. After many comparative experiments, we finally chose 16 to train.

The number of iterations (epochs) is the number of times training is optimized for the same batch of data. Theoretically, the more training you have in general, the smaller the loss function and the higher the accuracy. But generally, Batch Size is larger, and to reach the same accuracy, you have to increase the epoch, but too high an epichs can cause serious overfitting. We chose 32 to train.

Regularization parameters, L2 regularization can be seen as a penalty for loss functions. Reduce overfitting by adjusting the size of the parameters of L2. We used 0.005 as the L2 parameter in our experiment.

4 Experiment data

To evaluate the MCL model, we used three data sets: Human Activity Recognition (HAR), MIT-BIH, Sleep-EDF. The experimental equipment we use during training is a personal notebook: CPU: Intel(R) Core (TM) i5-10400 CPU s 2.90GHz 2.90 GHz. GPU:RTX2070 RAM:16.0GB. System type:64-bit operating system, x64-based processor (Window 10).

4.1 **Lab 1:** Human Activity Recognition (HAR)

The Activity Set dataset we used was the 2012 "A public domain dataset for human activity recognition using smartphones"[25].

The data was collected from 30 subjects between the ages of 19 and 48, who wore a waist smartphone that recorded motion data and performed one of six standard activities. Videos of each theme performing activities are recorded, and motion data is manually marked from those videos. The six activities performed are as follows:1 walk 2 go upstairs,3 go downstairs, 4 sit, 5 stations, 6 lie down. Smartphones record accelerometer and gyroscope data for x, y, and z, with observations recorded at 50Hz [8].

The original dataset was not available, and we used preprocessed data. Pretreatment includes: Pre-processing accelerometers and gyroscopes using noise filters. Split the data into 2.56 seconds (128 data points) each, overlapping by 50%. Split the accelerometer data into gravity (total) and body movement components [7].

The dataset can be loaded free of charge from the UCI Machine Learning repository. https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

Each experimental data is one of the six activities, with 128 time series and 9 channels.

The training sample was 2947 out of 7,352 test samples. No validation sets are divided.

Figure 3 is the architecture used by the MCL model in Activity HAR.

TABLE I: Super parameter

ConvLSTM2D	Head one	Head two	Head three
Filters	128	128	128
Kernel_size	(1,3)	(1,5)	(1,7)
dropout	0.5	0.5	0.5
L2	0.005	0.005	0.005

TABLE II: experimental result

	Accuracy	F1_socre	recall	precision
MCL	93.553%	93.503%	93.571%	93.823%
Multi_CNN	91.076%	91.153%	91.459%	91.651%
ConvLSTM	91.805%	91.766%	91.873%	91.893%

We can see that theMCLmodel has a higher accuracy and better performance on Activity Aware (HAR) than the general ConvLSTM model. MCL doesn't have much computational power in terms of computational costs.

4.2 Experiment 2: MIT-BIH

We chose MIT-BIH Arrhythmia Database data, which is the most commonly used database in the ECG space. The database is available for free with physionet. The website is as follows.

https://www.physionet.org/cgi-bin/atm/ATM

The MIT-BIH Arrhythmia Database contains 48 double-lead ECG records, of which, with the exception of a few records, the first lead for each record is an II lead, each recording 30 minutes long with a sample rate of 360Hz. Each record has 650,000 messages. We make small wave denoising of the data and perform heart beat interception on the basis of R peak, using each beat as a piece of data. At the same time, we divide all heart beats into four categories:"Normal (N)," "Left Beam Block (LBBB)," "Right Beam Block (RBBB)," and "Room Early Fight (PVC) for training [23].

A training sample is a heart beat with2,56 time series, a single channel, and each sample belongs to one of four types of heart beats.

The training sample is 80538. The test sample is 20134.

TABLE III:Super parameter

ConvLSTM2D	Head one	Head two	Head three
Filters	32	32	32

Kernel_size	(1,3)	(1,5)	(1,7)
dropout	0.5	0.5	0.5
L2	0.005	0.005	0.005

TABLE IV: experimental result

	Accuracy	F1_socre	recall	precision
MCL	99.257%	96.519%	95.232%	97.913%
Multi_CNN	99.237%	96.589%	95.807%	97.462%
ConvLSTM	99.036%	95.731%	94.834%	96.771%

MCL and Multi_CNN perform little differently in data classification training for MIT-BIH Arrhythmia Database. For data sets large enough, MCLs can't go any further when the average CNN model can reach a theoretical limit close to human performance.

4.3 Lab 3: Sleep-EDF

The Sleep-EDF database contains 197 multisonic graphical sleep records, including EEG, EOG, EMG, and event tags. Some records also include breathing and body temperature. We make sleep state classification predictions. The input ECE signals are classified into one of five categories: Wake(W), Non-Fast Eye Movement (N1, N2, N3), and Fast Eye Movement (REM). Sleep-EDF Public Data Database Download Link [24].

https://www.physionet.org/content/Sleep-EDFx/1.0.0/

EOG and EEG signals are sampled at 100 Hz, respectively. The EMG signal is filtered through electron high pass filtering, correction, and low pass, and then samples the EMG envelope in root squares at1Hz. We divided the data into4,2308, each with 3,000 time series, a single channel, and each sample representing a brainwave signal.

The data were 8285,2804,17799,5703,7717 samples per class, and we treated the data unbalanced and adjusted the sample weights (2,6,1,3,2) to reduce overfitting.

After pre-processing the data, put it into theMCL model for training. The results of the experiment are as follows:

The total data sample is 42308. We divide the data into training sets (60%), validation sets (20%) and test sets (20%).

TABLE V: Super parameter

ConvLSTM1D	Head one	Head two	Head three
Filters	128	128	128
Kernel_size	7	11	13
dropout	0.5	0.5	0.5
L2	0.001	0.001	0.001

TABLE VI: experimental result

	Accuracy	F1_socre	recall	precision
MCL	83.016%	75.555%	75.500%	75.742%
Multi_CNN	76.752%	68.302%	68.631%	68.917%
ConvLSTM	82.165%	74.436%	74.819%	74.634%

MCL performs better in situations where data is difficult to classify.

4.4 Discussion

In most cases the MCL model has better performance, both in Accuracy and on F1_socre, recall, and precision with greater accuracy. Especially for sample data with a small number of timing characteristics, such as Activity Recogniz (HAR)in Lab 1, this is because the Multi_head structure in the MCL can extract more features and make more accurate judgments. The MCL model does not have a particular advantage over data sets where general model training results are close to human performance.

5 Conclusion

In this article, we present the Multiple-head Convolutional LSTM (MCL), a Healthcare timing analysis model based on artificial intelligence technology. We use regularization, data normalization, trimming, and changing sample weights to reduce overfitting and process data imbalances. In the experiment, we also tried to use L1 regularization, increase filters, add more timing inputs, and add more ConvLSTM layers to One_head ConvLSTM, but found that these increased training time and calculation costs, but did not significantly increase mcCs L performance, so we put only one

ConvLSTM layer in One_head ConvLSTM to represent MCL performance. In future experiments, we hope to build more complex and in-depth models to better classify the health care timing.

Reference

[1] M. Irfan and I. A. Hameed, "Deep learning based classification for healthcare data analysis system," 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), 2017, pp. 1-6, doi: 10.1109/BESC.2017.8256396.

[2] TY - BOOKAU - Kisa, Deniz Hande, AU - Özdemir, Mehmet, AU - Güren, Onan,AU - Akan, Aydin, PY - 2020/11/19SP - T1 - EMG based Hand Gesture Classification using Empirical Mode Decomposition Time-Series and Deep Learning doi:10.1109/TIPTEKNO50054.2020.9299282ER

[3] Rajan Jeyaraj P, Nadar ERS. Atrial fibrillation classification using deep learning algorithm in Internet of Things-based smart healthcare system. Health Informatics J. 2020 Sep;26[3]:1827-1840. doi: 10.1177/1460458219891384. Epub 2019 Dec 16. Retraction in: Health Informatics J. 2020 Nov 3;:1460458220968756. PMID: 31840559.

[4] M. Hartmann, H. Farooq and A. Imran, "Distilled Deep Learning based Classification of Abnormal Heartbeat Using ECG Data through a Low Cost Edge Device," 2019 IEEE Symposium on Computers and Communications (ISCC), 2019, pp. 1068-1071, doi: 10.1109/ISCC47284.2019.8969657.

[5] A. M. Shaker, M. Tantawi, H. A. Shedeed and M. F. Tolba, "Generalization of Convolutional Neural Networks for ECG Classification Using Generative Adversarial Networks,"in IEEE Access, vol. 8, pp. 35592-35605, 2020, doi: 10.1109/ACCESS.2020.2974712.

[6] Taissir Fekih Romdhane, Haikel Alhichri, Ridha Ouni, Mohamed Atri, Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss, Computers in Biology and Medicine, Volume 123,2020,103866,ISSN 0010-4825,Doi:10.1016/j.compbiomed.2020.103866.

[7] https://machinelearningmastery.com/how-to-develop-rnn-models-for-human-activity-recognition-time-series-classification

[8] https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/

[9] https://www.tensorflow.org/tutorials/structured_data/imbalanced_data

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, Wang-chun Woo Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

[11] ixia Guo & Sakima Smith & Yosef M Khan & James R Langabeer II & Randi E Foraker, 2021. "Application of a time-series deep learning model to predict cardiac dysrhythmias in electronic health records," DOI: 10.1371/journal.pone.0239007

[12] Ali, Ghada & Al-Libawy, Hilal. (2021). Time-Series Deep-Learning Classifier for Human Activity Recognition Based On Smartphone Built-in Sensors. doi:10.1088/1742-6596/1973/1/012127

[13] Sridhar, N., Shoeb, A., Stephens, P. et al. Deep learning for automated sleep staging using instantaneous heart rate. npj Digit. Med. 3, 106 (2020). doi:/10.1038/s41746-020-0291-x

[14] Goschenhofer J., Pfister F.M.J., Yuksel K.A., Bischl B., Fietzek U., Thomas J. (2020) Wearable-Based Parkinson’s Disease Severity Monitoring Using Deep Learning. In: Brefeld U., Fromont E., Hotho A., Knobbe A., Maathuis M., Robardet C. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2019. Lecture Notes in Computer Science, vol 11908. Springer, Cham. Doi：/10.1007/978-3-030-46133-1_24

[15]Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, Kang Zhang,Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,Cell,Volume 172, Issue 5,2018,Pages 1122-1131.e9,ISSN 0092-8674,doi:/10.1016/j.cell.2018.02.010.

[16] Bar, Yaniv & Diamant, Idit & Wolf, Lior & Greenspan, Heather. (2015). Deep learning with non-medical training used for chest pathology identification. 94140V. doi:/10.1117/12.2083124.

[17] Rezaei, M., Yang, H., & Meinel, C. (2017). Deep Learning for Medical Image Analysis. ArXiv, abs/1708.08987.

[18] Abdullah Alharbi & Wael Alosaimi & Radhya Sahal & Hager Saleh, 2021. "Real-Time System Prediction for Heart Rate Using Deep Learning and Stream Processing Platforms," Complexity, Hindawi, vol. 2021, pages 1-9, February. doi: 10.1155/2021/5535734

[19] Suryani, D & Shoumi, M & Wakhidah, Rokhimatul. (2021). Object detection on dental x-ray images using deep learning method. IOP Conference Series: Materials Science and Engineering. 1073. 012058. doi:/10.10881757-899X/1073/1/012058.

[20] D. Bian, P. Mehta and N. Selvaraj, "Respiratory Rate Estimation using PPG: A Deep Learning Approach," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 5948-5952, doi: 10.1109/EMBC44109.2020.9176231.

[21] https://blog.csdn.net/qq_40661327/article/details/107034575

[22] https://www.jianshu.com/p/54213477fba7

[23] https://blog.csdn.net/u011538734/article/details/73477421

[24] https://www.physionet.org/content/Sleep-EDFx/1.0.0/

[25] Anguita, D., Ghio, A., ... Reyes-Ortiz, J.L., 2013. A public domain dataset for human activity recognition using smartphones, in: ESANN 2013 Proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. pp. 437–442. URI http://hdl.handle.net/2117/20897 **ISBN** 978-2-87419-081-0