Trajectory time series classification algorithm based on convolutional self-attention mechanism

Hang Yang1*, Xiaohui Li1, Zhonglin Liu1, Licai Wang1, Qibin Luo1

1 Smart Platform R&D Department, North China Institute of Computing Technology, Beijing, 100000, China
*Corresponding author’s e-mail: cetc15@vip.163.com, yh1813725727@163.com

Abstract. As a kind of time series data in the field of transportation, trajectory has complex multi-dimensional information. How to fully excavate the multi-dimensional information of the trajectory and analyse its complex dynamic pattern is an important problem to be solved in the classification of trajectory time series. In response to the above problems, this paper improves the existing self-attention mechanism, and proposes a trajectory time series classification algorithm based on convolutional self-attention mechanism. According to the characteristics of the trajectory data, the convolutional self-attention mechanism first extracts the characteristics of the trajectory fragments in the trajectory to obtain the flight status characteristics, and then performs the self-attention operation on these characteristics, so as to achieve a better feature extraction effect. The experimental results show that the trajectory time series classification algorithm based on the convolutional self-attention mechanism has better feature extraction ability and classification accuracy than other existing algorithms.

1. Introduction
Time Series Classification (TSC, Time Series Classification) is an important research content of data mining. It allows us to investigate this complex, huge, and rapidly updated world from the perspective of time dimension. In recent years, there are many algorithms for researching time series data in different fields, such as transportation[1], clothing sales [2] and medical care[3]. Flight trajectory data is a kind of multi-dimensional time series data in the transportation field. Whether it can achieve a more accurate classification has important practical significance for rational planning of routes and identification of counterfeit aircraft.

Methods focusing on TSC tasks can generally be divided into three categories: 1) feature-based classification, 2) distance-based classification, and 3) model-based classification.

Algorithms based on feature classification [4][5] extract feature vectors from time series, and then apply traditional methods or neural network models for classification. Traditional methods are represented by Support Vector Machine (SVM) [6] and K-Nearest (KNN)[7]; neural network models can strongly fit nonlinear mapping and extract complex time features for classification [8], such as Maria[9]The proposed reservoir calculation based on recurrent neural network is used to learn the representation of multivariate TSC. However, support vector machines in traditional methods have difficulties in large-scale sample training and multi-classification problems. The K-Nearest proximity algorithm has problems of low efficiency and poor multi-dimensional data processing. The method of using neural network models has not been in the trajectory time. It is widely used in sequence.

The method based on distance classification aims to design a distance function to measure the similarity of a pair of time series. As long as a reasonable distance metric is obtained, we can apply
conventional algorithms to further classify. For example, DTW[10] is a typical distance-based algorithm, suitable for time series with different lengths. There are also semi-supervised techniques for constructing time series classifiers for TSC[11] and so on. However, the classification method based on sequence distance is not suitable for trajectory time series data that contains multiple types of information.

The method based on model classification assumes that every time series belonging to each category are generated by the latent model. In the training phase, the corresponding parameters of the potential model are learned, and the test samples are classified according to the possibility. For example, Hidden Markov Model (HMM) [12] is widely used in speech recognition. The defect of this model is that it cannot pay attention to the connection between the current state and the context and the whole world. The naïve Bayes sequence classifier [13] is another typical model-based method, which follows the assumptions that are independent of features, but the model cannot learn the interaction between features.

In this paper, a model-based method is selected, and a network model with convolutional self-attention mechanism as the core is used to classify the trajectory time series. The self-attention mechanism [14] has received widespread attention following the proposal of the transformer [15]. This architecture completely eliminates sequential processing and cyclic connections, and only relies on the self-attention mechanism to capture the global dependency between input and output. In the past three years, many variants of the self-attention mechanism have been applied to solve various problems in different fields. For example, OpenAI's GPT and GPT-2 (decoder-only converters for language model) [16], using BERT [17] bidirectional encoder for language representation learning, and image converters for image generation [18], General transformer [19] and reinforcement learning[20] for problem solving, etc. However, these self-attention improvements are mainly aimed at various specific scenarios of image and natural language processing, and are not suitable for special multi-dimensional time series data such as trajectory data. This article adopts improved methods based on the characteristics of trajectory time series data. Convolutional self-attention mechanism.

Aircraft trajectory data is a kind of multi-dimensional time series data, and each trajectory point has multi-dimensional information such as time, longitude, latitude, altitude, flight speed, course, call sign, flight number, etc. The goal of the thesis is to fully explore the correlation of each dimension in this multi-dimensional time series data and the complex dynamic pattern of the trajectory time series, so as to realize the accurate classification of the trajectory. The work of the paper is as follows:

1) Extraction of trajectory features.

According to the information characteristics and data characteristics of each dimension of the trajectory data, the paper divides the different dimension information of the trajectory data into data of different views for feature extraction, and fusion of the output features of each view as the input of the model, so as to make full use of the navigation Information implicit in the trace data.

2) Self-attention structure

For trajectory time series, the trajectory segment composed of several consecutive trajectory points contains important flight characteristics and intention information of the aircraft, which is extremely important for trajectory classification. The self-attention structure designed in this article is composed of a normal self-attention module and a convolutional self-attention module. The normal self-attention module is responsible for the connection between trajectory points and the whole world, and the convolutional self-attention module is responsible for trajectory fragments and global contact. This makes the self-attention structure have the advantages of two different modules, so as to better mine the characteristics of the trajectory time series.

3) Trajectory time series classification algorithm

We implemented trajectory time series classification algorithm based on trajectory feature extraction, self-attention structure and convolution module. We conducted comparative experiments on the model itself and comparative experiments with other models. The experimental results verify the effectiveness of the convolutional self-attention mechanism proposed in this paper, and the effective extraction ability and excellent recognition accuracy of the trajectory time series classification algorithm proposed in this paper.
2. Algorithm design

Assuming that the trajectory time series data can be expressed as $X = [x_1, x_2 \ldots x_C] \in \mathbb{R}^{D \times C}$, where $D$ is the dimension of the input trajectory point, and $C$ is the number of trajectory points. The goal of the trajectory time series classification algorithm in this paper is to use the algorithm $\gamma$ based on the convolutional self-attention model, so that $l = \gamma(e)$, where $l$ is the classification label, and $e$ is the flight path after feature extraction of the trajectory sequence $X$. The algorithm is composed of three parts: trajectory feature extraction, self-attention structure, and convolutional structure. The self-attention structure is composed of multiple normal self-attention modules (NSAM) and convolutional self-attention modules (CSAM). The convolutional self-attention module is designed with an improved convolutional self-attention mechanism as the core. The frame diagram of the trajectory time series classification algorithm is shown in Figure 1. The whole algorithm process is as follows:

- After feature extraction of the trajectory, the trajectory feature matrix is obtained.
- The generated trajectory feature matrix is used as the input of the self-attention structure. In the self-attention structure, the ordinary self-attention module is responsible for paying attention to the dependency relationship between individual trajectory points and global trajectory points, and the convolutional self-attention module is responsible for paying attention to the dependency relationship between trajectory fragments, so as to carry out the trajectory feature effective extraction.
- The convolution structure performs further feature extraction on the processing results of the self-attention structure, and finally obtains the classification results of the trajectory time series $X$ through the linear layer and the Softmax function.

![Figure 1. Framework diagram of trajectory time series classification algorithm](image)

2.1. Trajectory feature extraction

Because the trajectory time series data has multi-dimensional information, in order to better represent and learn it, we divide the different dimensional information of the trajectory data into different types of groups according to their characteristics, and independently perform feature extraction and fusion to obtain the flight path. The trajectory representation learning process diagram is shown in Figure 2. This article divides the dimensional information of the trajectory data into three categories:

- Time, latitude and longitude, and altitude, which are the most important aircraft time and position information for the trajectory. For this kind of dimensional information, this paper adopts the global coordinate grid method to obtain the feature matrix.
- Heading speed and other information. For this kind of dimensional information, the article standardizes the course and speed to obtain a feature matrix.
- Other supplementary information such as flight number, call sign, and destination. For this kind of dimensional information, this article uses a similar natural language processing method to build a corresponding dictionary for flight numbers, call signs, and destinations, and then Embedding generates its feature matrix.

The independent feature extraction of the three categories is shown in formulas (1), (2), (3), and the fusion of the results is shown in formula (4):
Among them, \( \tau_i, i \in 1,2,3 \) represent the representation learning results of the three types of dimensional information of the trajectory time series \( X \). \( E_i, i \in 1,2,3 \) are representation learning operations for different types of dimensions, \( X^a, X^b, X^c \) are the dimensional information of the three categories of sequence \( X \), \( \varphi \) is the fusion function of the three categories of learning results, \( C_i, i \in 1,2,3 \) are the fusion functions of the learning results of different categories. The weight parameter in \( \varphi \), the weight parameter \( C_i \) is continuously optimized during the model training process, \( e \in \mathbb{R}^{L \times H} \) is the trajectory embedding obtained after fusion, where \( L \) is the number of trajectory points. \( H \) is the dimension after embedding. After the trajectory feature extraction, we can get a better trajectory feature matrix representing the sequence \( X \) as the input of the self-attention structure.

\[
\begin{align*}
\tau_1 &= E_1(X^a) \\
\tau_2 &= E_2(X^b) \\
\tau_3 &= E_3(X^c) \\
e &= \varphi(\left[ \tau_1 \ast C_1, \tau_2 \ast C_2, \tau_3 \ast C_3 \right])
\end{align*}
\]

2.2. Self-attention structure
The trajectory time series diagram is shown in Figure 3. The yellow dots represent the trajectory points, and each trajectory point stores multi-dimensional information; the green line represents the aircraft trajectory composed of the trajectory points; the red dashed line represents the trajectory segment, the trajectory. The trajectory segment consists of several trajectory points. Our research found that the trajectory segment contains more abundant trajectory feature information than the trajectory point. From the segment, a more comprehensive flight status of the aircraft can be extracted, such as the acceleration and angular velocity of the flight. Therefore, the trajectory segment can effectively improve the feature extraction ability of the aircraft, which makes us not only use the existing self-attention network to learn the dependency between trajectory points and the global sequence, but also need to pay attention to the deep layer of the trajectory segment and the global sequence. For this reason, this paper designs a self-attention structure consisting of a normal self-attention module and a convolutional self-attention module.
The self-attention structure is shown in Figure 1. The paper uses the trajectory feature matrix and the position embedding in the transformer as the input vector of the structure. The structure consists of 3 ordinary self-attention modules and 3 convolutional self-attention modules. The modules are connected in series.

The structure of the normal self-attention module follows the Encoder Layer of the existing transformer model to mine the relationship between individual trajectory points and the entire sequence. The content of this module will not be repeated.

The structure of the convolutional self-attention module is shown in Figure 4. The convolutional self-attention module has two sub-layers, namely the multi-head convolution self-attention mechanism and Feed-forward network, each sub-layer has residual connection and layer normalization, normalization and residual connection are mechanisms to help the model train faster and more accurately. The representation of each layer is shown in formula (5):

\[
\text{sub\_layer\_output} = \text{LayerNorm}(x + \text{SubLayer}(x))
\]

Where \(x\) is the sub-layer input, Sublayer is the sub-layer operation, and LayerNorm is the normalization operation. Next, we will describe the multi-head convolutional self-attention mechanism and the feed-forward network separately.

Figure 3. Schematic diagram of the trajectory

Figure 4. Convolutional self-attention module structure diagram
The convolutional self-attention mechanism is shown in Figure 4, where $Q$, $K$, and $V$ are the trajectory feature matrix multiplied by the matrices $W_Q$, $W_K$ and $W_V$. Among them, $W_Q$, $W_K$ and $W_V$ are different weight matrices, which perform different linear transformation on the trajectory feature matrix and map them to different subspaces.

The convolutional layer in the convolutional self-attention mechanism is shown in Figure 5. The feature matrix before convolution is $Q$, $K$, $V$, and the feature matrix after convolution is set to $Q'$, $K'$, $V'$, and the red box on the left is the volume of the vector before the product, the red box on the right is the vector after the convolution, the height of the convolution kernel used is 3, and the width is 1, so the width of the feature vector before and after the convolution does not change. After convolution and linear transformation, the information of the trajectory segment composed of multiple points is stored in a low-dimensional vector space, and the dimension $H$ is equivalent to the feature vector of a single trajectory point. Convolution of the trajectory segment allows the attention mechanism to learn the dependence of the trajectory segment on the global trajectory time series.

$$Q' = \text{Con}(Q) + \text{bias}_q'$$

$$K' = \text{Con}(K) + \text{bias}_k'$$

$$V' = \text{Con}(V) + \text{bias}_v'$$

The Con function represents the convolution operation performed on $Q$, $K$, and $V$. They share the same convolution weight to reduce the use of parameters, $\text{bias}_q'$, $\text{bias}_k'$, $\text{bias}_v'$ is the unique error vector of $Q'$, $K'$, and $V'$.

As shown in Figure 4, the zoom and click self-attention layer is mainly a self-attention operation, which can be described as a mapping from a query ($Q'$) to a series of ($K'$, $V'$) pairs. The self-attention operation is mainly divided into three steps. The first step is to calculate the similarity between $Q'$ and each $K'$ to obtain the weights. This article uses the dot product method to obtain the weights; then the second step is generally to use the Softmax function to calculate these weights; finally, the weight and the corresponding key value are weighted and summed to obtain the final self-attention vector. The formula for self-attention operation is shown in formula (9):

$$\text{Attention}(Q', K', V') = \text{softmax}\left(\frac{Q'R'}{\sqrt{d_k}}\right)V'$$

Among them, $d_k$ is the dimension of the vector $Q'$, and scaling with $\sqrt{d_k}$ plays a distracting role, so that the model has better generalization ability. The convolutional attention mechanism has a multi-head structure because several attention layers use the same input to perform different linear transformations and stack them, and this stacking helps the model capture various dimensional information of the input and improve the feature extraction ability of the model. Figure 4 The $A$ represents the number of attention heads, and the value of $A$ is 8 in this article. The merge layer is to merge the results of multiple attention heads. The formulas are shown in (10) and (11):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_A)W^0$$

$$\text{head}_i = \text{Attention}(QW^0_i, KW^K_i, VW^K_i)$$
The Concat function is the merging layer in Figure 4, which merges the generated results of each attention head. In addition to the general linear transformation, the linear layer in Figure 4 also uses a fully connected layer to perform dimensional transformation on the result after merging, so that the output dimension is consistent with the input dimension, so as to ensure that the convolutional self-attention module can be reused in series.

The feedforward network layer in Figure 4 contains linear transformations and activation functions. In the feedforward network, the dimension H of the input vector is expanded by linear transformation. This paper uses the dimension of 2H. In the selection of the activation function, we use the GELU activation function of the Bert model. The approximate formula used in this paper is shown in formula (12):

$$GELU(x) = 0.5x(1 + \tanh \left( \frac{x}{\sqrt{2\pi}(x + 0.044715x^3)} \right)$$

2.3. Convolutional structure
The convolution structure is shown in Figure 6. This structure performs convolution pooling on the output result of the self-attention structure, so that the model extracts trajectory features in one step, so as to achieve a better classification effect. We compare the feature matrix output from the self-attention structure to an image, and use a convolutional neural network to extract features. As shown in Figure 1, the difference from the traditional convolution method is that the adjacent trajectory points in the trajectory feature are highly correlated, so this paper only performs the convolution in the vertical direction of the feature matrix. The width of the convolution kernel is fixed to the dimension H of the input vector, and the height of the convolution kernel is a hyperparameter. After experimentation, we select (3, 5, 7) these three different height combinations for feature extraction, and through different height volumes the product core extracts the features of different scales of the feature matrix, connects the extraction results as the input of the classification function, and finally obtains the specific classification category through the fully connected layer and the Softmax function.

3. Experiment
3.1. Experimental data
The experiment uses ADS-B data, and the ADS-B data is derived from Automatic Dependent Surveillance-Broadcast System. ADS-B data acquisition is easier, and the rich message information is more conducive to the monitoring of complex and busy traffic airspace. Some field information of the trajectory points is shown in Table 1. The experimental ADS-B data in this article has a total of 1 billion trajectory points. The trajectory points of 20 types of aircraft are first extracted from them, and then the trajectory points of each type are determined according to ICAO. The (International Civil Aviation Organization) number and call sign are separated, and finally the time interval of the trajectory point is processed according to the flight law of the aircraft to obtain the aircraft trajectory time sequence. Since the number of trajectories obtained by each type of aircraft is different, in order to balance the experiment, this article selects 10 aircraft with 2000 trajectories for the experiment, including 6 general models (A343, AC95, AT72, C550, DC10, IL76) and specific models in 4.
3.2. Experimental environment settings
Experimental environment: operating system: CentOS7.6 operating system; Programming language: Python3.7; Framework: Open source deep learning framework Pytorch, version 1.4.1; Graphics Processing Unit (GPU): RTX TITAN. This experiment uses the Pytorch framework and uses CUDA and cuDNN to train on the GPU to improve the training speed of the model.

3.3. Convolutional self-attention mechanism comparison experiment
In order to verify the effectiveness of the convolutional self-attention mechanism, an experimental study was conducted based on the ADS-B data set. In addition to the difference in the number of ordinary self-attention modules and convolutional attention modules, the experimental settings are consistent with other parameters. In the experiment, the initial learning rate is $10^{-6}$, the optimization algorithm selects the Adam algorithm, the training batch size $BATCH$ is set to 32, and the height of a set of convolutional kernels in the convolution module is $(3, 5, 7)$. The width of the convolution kernel is the dimension 256 of the trajectory feature matrix, and the value of the number of self-attention heads $A$ is 8. In the experiment, three types of aircraft trajectories are selected for three classifications, and the ADS-B trajectory is split according to the ratio of training set, validation set, and test set to 0.8, 0.1, and 0.1. During the experiment, there are 6 layers of normal self-attention module and convolutional self-attention module, which are divided into 7 groups of experiments according to the different ratios of the two. The experimental results are shown in Table 2:

| MODULE QUANTITY RATIO | LOSS | ACC (%) | RECALL (%) | F1 (%) |
|-----------------------|------|---------|------------|--------|
| 0:6                   | 0.3  | 91.47   | 91.45      | 91.44  |
| 1:5                   | 0.18 | 94.03   | 94.02      | 93.08  |
| 2:4                   | 0.19 | 95.05   | 95.04      | 95.02  |
| 3:3                   | **0.17** | **95.05** | **95.04** | **95.03** |
| 4:2                   | 0.23 | 93.52   | 93.53      | 93.44  |
| 5:1                   | 0.18 | 94.71   | 94.68      | 94.58  |
| 6:0                   | 0.18 | 94.71   | 94.68      | 94.68  |

The following conclusions can be drawn from Table 2:
- A module number ratio of 0:6 means that only convolutional self-attention modules are used, and a module number ratio of 6:0 means that only ordinary self-attention modules are used, neither of which achieves the best experimental results.
- Compared with only using the ordinary self-attention module, the combination of the two has achieved better experimental results. This is because the ordinary self-attention module can extract the features of the trajectory points, and the convolutional self-attention module can also extract the unique features of the trajectory segment to verify the effectiveness of the convolutional self-attention mechanism.
- From the experimental results, it can be seen that when the number of ordinary self-attention modules and convolutional self-attention modules are equal (the number ratio in the experiment is 3:3), the model achieves the best experimental results, loss value, accuracy rate, and recall rate, F1 values are 0.17, 5.05%, 95.04%, 95.03% respectively.

3.4. Self-attention structure module layer number comparison experiment
In order to explore the influence of the number of module layers in the self-attention structure on the classification effect, an experimental study is carried out based on the ADS-B data set. Three sets of comparative experiments are set up in the experiment, and the number of convolutional self-attention modules is 6 layers, 12 layers, and 18 layers respectively. In the experiment, the ratio of ordinary self-
attention module and convolutional self-attention module is 1:1, and other parameters are consistent with experiment 3.3. The experimental results are shown in Table 3:

| MODULE LAYERS | LOSS (%) | ACC (%) | RECALL (%) | F1 (%) |
|---------------|----------|---------|------------|--------|
| 6 layers      | 0.17     | 95.05   | 95.04      | 95.03  |
| 12 layers     | 0.16     | 94.03   | 94.02      | 93.99  |
| 18 layers     | 0.21     | 93.86   | 93.80      | 93.82  |

It can be seen from Table 3 that the increase in the number of layers of the modular convolutional self-attention module did not improve the accuracy of the model. The accuracy of the model has decreased, except that the loss at 12 layers is reduced by 0.1, and the other results are not as good as the 6-layer algorithm. The analysis is because too many layers and more parameters increase the difficulty of optimization, and it is difficult to achieve the best results. From experiments 3.3 and 3.4, it can be seen that when the number of self-attention structure modules of the model is 6 layers, and the ratio of ordinary self-attention modules and convolutional self-attention modules is 1:1, the effect of trajectory time series classification It is best to conduct comparative experiments with other models with this parameter in the follow-up article.

3.5. Comparison experiment of trajectory time series classification algorithms

In order to verify the effectiveness of the trajectory time series classification algorithm based on the convolutional self-attention mechanism, this paper compares with a variety of algorithms on the ADS-B data set, including SVM algorithm, Transformer algorithm, BiLSTM algorithm, CNN algorithm, BiLSTM-CNN algorithm (combined with BiLSTM and CNN algorithm), Transformer-CNN (combined with Transformer and CNN algorithm). The proportions of the experimental training set, validation set, and test set are still 0.8, 0.1, and 0.1. The three-classification experiment results of different models are shown in Table 4:

| MODEL                   | LOSS (%) | ACC (%) | RECALL (%) | F1 (%) |
|-------------------------|----------|---------|------------|--------|
| SVM[6]                  | 1.05     | 56.31   | 55.72      | 54.65  |
| Transformer[15]         | 0.2      | 94.71   | 94.70      | 94.69  |
| BiLSTM                  | 0.24     | 90.69   | 90.89      | 91.00  |
| CNN                     | 0.16     | 94.03   | 94.02      | 93.99  |
| BiLSTM-CNN              | 0.15     | 93.86   | 93.85      | 93.83  |
| Transformer-CNN         | 0.17     | 94.37   | 94.37      | 94.33  |
| **Our Algorithm**       | **0.17** | **95.05** | **95.04** | **95.03** |

From Table 4, the following conclusions can be drawn:

- The SVM model is poorly classified in the trajectory time series. The analysis is due to the complex characteristics of the trajectory time series and the multi-classification problem that SVM is difficult to handle.
- The convolution module has greatly improved the effect of algorithm classification, and the model fused with the convolution module has a higher classification accuracy overall.
- The trajectory time series classification algorithm used in this article has achieved the best classification results, except that the loss value is slightly higher than the BiLSTM-CNN model, and the accuracy, recall and F1 value are higher than other models.

3.5.1. PR curve. In addition to the parameters for evaluating the performance of different models in Table 3, this article draws Precision-Recall (PR) curves based on recall and precision, and evaluates the performance of the trajectory time series classification algorithm more intuitively through the PR curve. AP It is the surface of the area under the PR curve. The P-R comparison diagram of different models is shown in Figure 7:
Figure 7. Comparison of P-R curves of different models

It can be seen from Figure 7 that the algorithm in this paper has the largest AP area. When the recall rate $R<0.5$, except for the SVM model, the average accuracy of the other models is almost the same, and when the recall rate $R>0.5$, the algorithm in this paper has a good performance in most cases. The accuracy measured under the high recall rate ($R>0.6$) is more effective than the small recall rate. Overall, the algorithm in this paper is better than other algorithms.

4. Conclusion

In order to effectively improve the feature extraction and classification accuracy of the trajectory time series, this paper proposes a trajectory time series classification algorithm based on the convolutional self-attention mechanism according to the characteristics of the trajectory time series data. (1) A method for extracting trajectory features is designed. The dimensions of multi-dimensional trajectory data are divided into three types according to their own characteristics, and feature extraction is performed independently and then fused to obtain a better trajectory feature matrix; (2) This paper designs a self-attention structure. The self-attention structure adopts the combination of ordinary self-attention module and convolutional attention module, which not only pays attention to the dependency between individual trajectory points and the global, but also pays attention to the dependency between trajectory fragments and the global. The convolutional self-attention module is designed with the convolutional self-attention mechanism as the core. The convolutional self-attention mechanism first extracts the features of the trajectory segment, and then performs the self-attention operation between the segments. This allows the convolutional self-attention module to learn the dependencies between trajectory segments, and some trajectory features of the trajectory segments are not available for individual trajectory points. (3) The paper designs and implements the trajectory time series classification algorithm, and verifies its effectiveness through experiments. In this paper, a comparative experiment on the algorithm itself: Convolutional self-attention mechanism comparison experiment and self-attention structure module layer number comparison experiment, as well as comparison experiments with other models on three categories. (4) The experimental results show that the improved self-attention module effectively improves the ability of feature extraction, and the trajectory time series classification algorithm has a higher trajectory classification accuracy.

This article believes that the trajectory time series classification algorithm has a certain generalization ability when the time series segment contains unique characteristic information, and can be applied to
more types of time series data. Later, we will conduct in-depth research in this direction. It is worth pointing out that the trajectory representation learning method adopted in this article uses a relatively simple fusion mechanism for different types of dimensional information. Follow-up research will combine multi-view learning (MVL, Multi-view learning) for further research. In-depth integration of information of different dimensions improves the representation learning ability of multi-dimensional time series data.

References
[1] Yao H , Fei W , Ke J , et al. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction[J]. 2018.
[2] Vijay Ekambaram, Kushagra Manglik, Sumanta Mukherjee, Surya Shravan Kumar Sajja, Satyam Dwivedi, Vikas Raykar. Attention based Multi-Modal New Product Sales Time-series Forecasting[P]. Knowledge Discovery & Data Mining, 2020.
[3] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, Aram Galstyan. Multitask learning and benchmarking with clinical time series data[J]. Scientific Data, 2019, 6(1).
[4] Mohammed Waleed Kadous, Claude Sammut. Classification of Multivariate Time Series and Structured Data Using Constructive Induction[J]. Machine Learning, 2005, 58(2-3).
[5] Ye L , Keogh E J . Time series shapelets: a new primitive for data mining[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2009.
[6] Cortes C. Support-Vector Networks[J]. Machine Learning, 1995, 20.
[7] Fukunaga K , Narendra P M . A Branch and Bound Algorithm for Computing k-Nearest Neighbors[J]. IEEE Trans Comput, 1975, 24(7):750-753.
[8] Karim F , Majumdar S , H Darabi, et al. Multivariate LSTM-FCNs for time series classification[J]. Neural Networks, 2018, 116.
[9] Bianchi Filippo Maria, Scardapane Simone, Lokse Sigurd, Jenssen Robert. Reservoir Computing Approaches for Representation and Classification of Multivariate Time Series.[J]. IEEE transactions on neural networks and learning systems, 2020, PP.
[10] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction[P]. Machine learning, 2006.
[11] Wei L , Keogh E J . Semi-supervised time series classification[C]// Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006. ACM, 2006.
[12] Rabiner L R . A tutorial on hidden Markov models and selected applications in speech recognition[J]. Proc IEEE, 1989, 77.
[13] Rish I . An empirical study of the naive Bayes classifier[C]// IJCAI-01 Workshop on Empirical Methods in AI. 2001. 2001.
[14] Chaudhari S , Polatkan G , Ramanath R , et al. An Attentive Survey of Attention Models[J]. 2019.
[15] Vaswani A , Shazeer N , Parmar N , et al. Attention Is All You Need[J]. arXiv, 2017.
[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners. (2018).
[17] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.
[18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image Transformer (Proceedings of Machine Learning Research, Vol. 80). PMLR, 4055–4064.
[19] Dehghani M , Gouws S , Vinyals O , et al. Universal Transformers[J]. 2018.
[20] Parisotto E , Song H F , Rae J W , et al. Stabilizing Transformers for Reinforcement Learning[J]. 2019.