A Time Series Forecasting Model Selection Framework Using CNN and Data Augmentation for Small Sample Data

Wentao Jiang (Jiangwt2@163.com)
South China Agricultural University
Liwen Ling
South China Agricultural University
Dabin Zhang
South China Agricultural University
Ruibin Lin
South China Agricultural University
Liling Zeng
South China Agricultural University

Research Article

Keywords: CNN-based Forecast-model Selection, Data Augmentation , Time series image(TSI), automatic feature extraction, meta-learning

Posted Date: February 17th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1094384/v1

A time series forecasting model selection framework using CNN and data augmentation for small sample data

Wentao Jiang · Liwen Ling · Dabin zhang* · Ruibin Lin

Wentao Jiang College of Mathematics and Informatics, South China Agricultural University
Tel.: +86-18469187774
E-mail: Jiangwt2@163.com
Liwen Ling College of Mathematics and Informatics, South China Agricultural University
E-mail:linglw@scau.edu.cn
*Dabin Zhang * College of Mathematics and Informatics, South China Agricultural University*
E-mail: zdbff@aliyun.com
Ruibin Lin College of Mathematics and Informatics, South China Agricultural University
E-mail:450169049@qq.com
Liling Zeng College of Mathematics and Informatics, South China Agricultural University
E-mail:1538023416@qq.com

Abstract

The key to the accuracy of time series forecasting is to find the most appropriate forecasting method. Therefore, the forecasting model selection of time series has become a new research hotspot in the data analysis field. However, most of the existing forecasting model selection methods reduce the forecasting efficiency for relying on subjective manual selection of features. In this paper, an automatic time series feature extraction framework is proposed for forecasting model selection based on the idea of meta learning. Inspired by computer vision, we transform one -dimensional time series into two-dimensional images, and use convolution neural network (CNN) to train and classify time series images (model selection). Moreover, in order to deal with the over fitting problem caused by small sample datasets, the sliding window data augmentation method is used to improve the accuracy of small datasets model selection. A large-scale empirical study on M3 datasets shows that the proposed framework has better model selection accuracy and smaller forecasting error(MAPE) than Support vector machine(SVM) and traditional time series image algorithms. Moreover, the classification rate(model selection accuracy) of the proposed algorithm are increased by 6.5% and 4.4% compare with the traditional time series image method and Support vector machine respectively in average.

Keywords: CNN-based Forecast-model Selection, Data Augmentation , Time series image(TSI), automatic feature extraction, meta-learning

1. Introduction

Time series forecasting analysis is an important role of financial industry, from forecasting economic phenomena to forecasting product sales(Morwitz et al., 2007). In recent decades, a great quantity of time series prediction methods have been applied, aiming to improve the forecast accuracy. However, from the ”no free lunch” theorem(Macready) everyone know that no method is applicable in any time series.
In the past period of time, time series forecasting model selection is chiefly relied on feature selection. Scholars have

made a lot of attempts on the feature-based single variable time series forecasting model selection method. For example, based on 26 time series features, Shah constructed multiple individual selection rules by discriminant analysis Shah(Shah, 1997); Wang, Smith miles and Hyndman(Kang et al., 2017) according to the meta characteristics of time series, supervised and unsupervised learning methods are applied to generate rules for selecting the optimal forecasting model; Lemke and Gabrys(Lemke & Gabrys, 2010) provided a new set of time series feature element learning method for NN3 and NN5 data sets, and analyzes the results; Widodo and Budi(Widodo & Budi, 2013)used principal component analysis to reduce the feature dimension, so as to optimize the forecasting model selection method; Petropoulos et al.(F Petropoulos, 2014) proposed the "horse for course" in M3 dataset(Makridakis et al., 2001), and counted the effects of 7 different time series features on the performance of 14 good forecasting methods. Recently, Talagala et al.(Talagala et al., 2018) proposed a novel framework using random forests as classifiers and meta learning for forecasting model. Therefore, the selection of features plays an important role in the selection of time series forecasting model.

At present, most feature-based time series forecasting model selection algorithms rely on manual feature selection. However, with the advent of the era of automation, manual feature selection seem to be outdated. And it is true that manual feature selection is too cumbersome and consumes a lot of manpower and computing resources. Therefore, the deep learning framework for automatically extracting time series features gives scholars great inspiration.

In recent years, deep learning is widely used in time series forecasting. Connor and Martin(Connor et al., 2002) proposed recurrent neural network (RNN) for the first time, and taken the original characteristics of time series as input to predict the subsequent trend; On this basis, Gers proposed an improved RNN network called long and short term memory neural network (LSTM) to do some prediction(F.A. Gers, 2001). Naduvil-Vadukootu et al.(Naduvilvadukootu et al., 2017) proposed a pipeline framework, which combined the mainstream time series forecasting methods with deep neural network (DNN), so as to improve the forecasting accuracy of time series.

Deep learning needs a lot of data to train the time series. However, in many real-world datasets (such as agricultural product price, sales volume, etc.) the small sample training set problem remains. So the insufficient of data can also be a problem for time series analysis, which leads to over fitting and the low performance.

Data augmentation has been proved to be an effective way to reduce the over fitting of neural network model (Shorten & Khoshgoftaar, 2019). In real life, the amount of data in many fields hard to meet the requirements of deep learning model training, so the use of data augmentation can help the network overcome the problem of too small datasets or class imbalance(Hasibi et al., 2019). Although generalization and regularization methods can be used to reduce over fitting, data augmentation solves the problem from the data preprocessing point without changing the structure of neural network models.

This paper aims to propose an improved meta learning framework to overcome the problem of time series feature selection manually and over fitting of small sample datasets. Get inspired from the achievement of Hatami et al.(Hatami et al., 2019) and Wang and Oates(Wang & Oates, 2015) in image processing, this paper combines the idea of time series imaging and meta learning framework using convolution neural network (CNN) to select best time series forecasting model. This framework can automatically extract time series features and avoid the problem of different standards caused by subjective factors in feature selection. At the same time, window slicing data augmentation is used to solve the problem of over fitting of small datasets in the process of deep learning training, which can improve the accuracy of model selection. And the proposed algorithm in this paper achieve better result in forecasting model selection compare with traditional time series imaging algorithm and support vector machine(SVM).

2. Forecasting Model Selection

2.1. Meta-learning forecasting model selection

John Rice is the first proposer of meta learning in 1976, which called algorithm selection problem(Rice, 1976). The selection structure of Rice algorithm mainly consists of four parts. The problem space P represents the data set involved in the experiment. Feature space F is the set of all features in problem space P. Algorithm space A is a group of excellent candidate algorithms to solve problem space P problem. Performance metric Y is a measure of algorithm performance such as classification accuracy and running speed. Smith Miles(Smith-Miles., 2009) put forward a clear definition of algorithm selection in 2009.

Algorithm selection problem(ASP). For a given problem instance $x \in P$, with features $f(x) \in F$, find the selection mapping $S(f(x))$ into algorithm space A, such that the selected algorithm $\alpha \in A$ maximizes the performance

mapping $y(\alpha(x)) \in Y$.
The main bottleneck of ASP is to recognize the selection mapping from feature space to algorithm space. Although Rice's framework shows the concept of ASP, it does not specify how to obtain S, so it introduces the meta learning method.
With the wide application of machine learning, the term meta learning also appears in the time series literature. PrudeNcio and ludermir(Prudencio & Ludermir, 2004) was the ancestor of applying meta learning to time series, and discussed the influence of meta learning methods on model selection. Wang, Smith-Miles and Hyndman(Smith-Miles., 2009) introduces a new model selection method based on meta learning framework, called simple percentage better (SPB), whose model selection accuracy changes with the forecasting accuracy error of random walk model. Later, Widodo and Budi (Widodo & Budi, 2013) proposed a novel meta learning framework for prediction model selection, which is based on a set of features proposed by Wang, Smith-Miles and Hyndman. Recently, KüCK, Crone and Freitag(Kuck et al., 2016) combined neural network with meta learning to select the forecasting model, constructed a set of new features based on forecasting error, and used the mean absolute forecasting error as the evaluation standard to determine the best forecasting model of each time series.

2.2. CNN-based Forecast-model Selection(CFMS)

The proposed CFMS framework is presented in Figure 1. The frame diagram shows the original time series set stage and the new time series set stage respectively. The model selection (meta learner) is trained in the original time series stage, and the trained algorithm is used to select the appropriate forecasting model for the new time series. For making our trained classification algorithm perform better, large number of time series which consistent with the type we are going to forecasting are necessary.We suppose that there are a large number of time series populations, and they are used as samples to train the classification algorithm. Therefore, the newly input time series in this framework is regarded as additional data similar to the data type of training set, which can be called the "target type" of time series. In fact, the classification accuracy can be improved by simulating and expanding the data similar to the time series of the training set (we will discuss it in detail in Section 2.2 below). In this paper, the total set of time series used to train classifiers is expressed as "reference set". Each time series in the reference set is divided into training period and testing period. Each training cycle in the convolution neural network will fit a candidate forecasting model, that is a classification algorithm. We apply the model trained in the training set to the forecasting error calculated (such as MAPE) in the test set, and determine the "best" model of each time series according to the forecasting error. These models considered the "best" form the output label of the classification algorithm. Algorithm 1 below gives the pseudo code of our proposed framework. In the following section, we briefly discuss some aspects of training the original time series phase. The duty of model selection algorithm is to assign the "best" forecasting method to a given time series. It is impossible to train classifiers for all possible model classes, so in this paper, we choose six most popular time series forecasting models. The selected candidate models will depend on the type of time series. For example, if the time series only have non seasonal instead chaotic feature, the candidate forecasting model are limited by white noise, random walk, ARIMA and ETS processes. So even in this simple scenario, the number of matching models can be quite large.
Since each candidate model must be calculated and compared in each time series in the reference set, this step is the most computationally intensive and time-consuming step. The more candidate models are, the longer the calculation time is and the return may be a significant improvement in forecasting accuracy. This step is the most computationally intensive and time-consuming because each candidate model must be applied to each time series in the reference set. The more candidate models are, the longer the calculation time is and the return may be a significant improvement in forecasting accuracy.
The pseudo code for our proposed framework is presented in Algorithm 1 below.

Algorithm 1

Train the classification
Given:
O=$x_1, x_2, ..., x_n$: the classification of n observed time series;

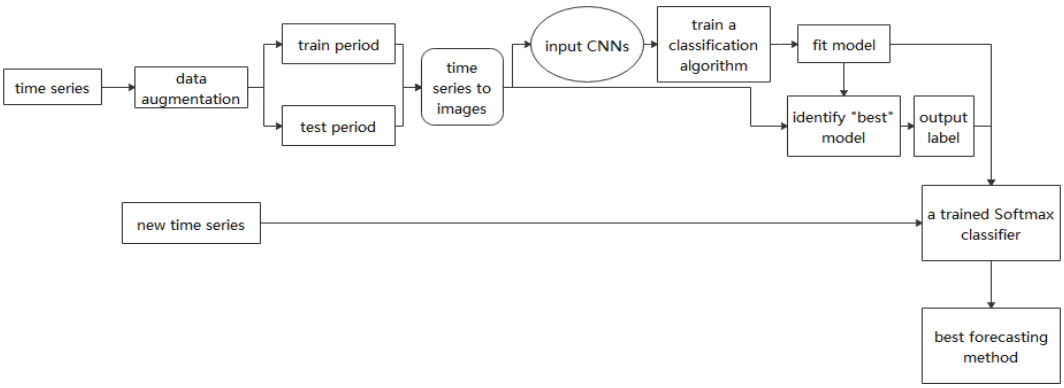

Figure 1: CFMS(CNN-based Forecast-model Selection) framework

C: the set of CNNs(e.g. Resnet-18,VGG-19,Densenet-121);
L: the set of class labels(e.g. ARIMA,ETS,THETA, and so on);
I: the set of time series image(e.g. MTF, RP, GAFs).
Output:
A trained SoftMax classifier.
Data preprocessing:
For i=1 to N
1. Split x_i into a training period and test period;
2. The training set is processed with sliding window length of 2 and step size of 1;
3. Transform x_i into four kinds of images: MTF, RP, and GAFs;
4. Fit L models to the training period;
5. Calculate forecasts for the test period from each model;
6. Calculate forecast error measure over the test period for all models in L;
7. Select the model with the minimum forecast error.
Prepare the meta framework based CNN:
8. Input time series images into CNNs;
9. Train the SoftMax classifier.
Forecast a new time series
Given:
the trained classifier from step 9.
Output:
Class labels from new time series x_{new}.
10. x_{new} repeat the step 2 and step 3;
11. Let C(x_{new}) be the class forecasting of our framework. Then calculate forecasting error according to label.

2.3. The candidate prediction model

The six candidate forecasting models (it is also called labels in supervised learning) are used in this article: (a) White noise (WN);(b) ARIMA;(c) Random walk with drift (RWD); (d) Random walk (RW); (e) Theta; (f) Exponential Smoothing Model (ETS). The several time series forecasting models used in this paper are as follows:

2.3.1. Exponential Smoothing

This model has been developed for several decades and was first proposed by Brown(Brown, 1977). It is the basis of many popular time series prediction algorithms. In exponential smoothing method, time series usually models the four parts of time series such as seasonality and damping by multiplication or addition(Winters, 1976). The ”ZZZ” model used in this paper refers to the best ETS model automatically selected by R software package according to AIC criteria ([Aut & [Aut, 2018).

2.3.2. Theta Method

Theta method is a single variable method for non-seasonal time series forecasting. Theta method is based on decomposing the original time series into"theta line" and solve the second-order difference equation to obtain a new time series. Each decomposed line is calculated by the forecasting algorithm, and the prediction results are reorganized to obtain the prediction results of the original time series(Assimakopoulos & Nikolopoulos, 2000). The algorithm is implemented by R package forcetheta(Fiorucci, 2016).

2.3.3. Random Walk and Random Walk with draft

Random walk (RW) is often used in financial data statistical models because of its effectiveness. The model assumes that adjacent observation points provide guidance for the next predicted value [39]. The mathematical expression of RW model is as follows:

$$y_t - y_{t-1} = \epsilon_t \tag{1}$$

where y_{t-1} and y_t are the observed value of time series, and ϵ_t is a white noise. The white noise term obeys the normal distribution, its mean value is zero and has constant variance σ^2.

2.3.4. ARIMA

In ARIMA model, the forecasting value of a variable is related to several known observations and linear functions of random errors. In terms of mathematical expression, the basic process of generating time series has the following characteristics:

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - ... - \theta_q \epsilon_{t-q} \tag{2}$$

y_t and ϵ_t is the actual value and random error of time period t respectively; $\phi_i (i = 1, 2, ..., p)$ and $\phi_j (j = 0, 1, 2, ..., q)$ is the model parameter. Integers p and q are usually called the order of the model. The mean value of random error ϵ_t is zero and the variance is constant σ^2. Several special cases of $ARIMA$ are included in equation (2). If $q = 0$, then (2) becomes a p-order AR model. When $p = 0$, the model is simplified to a q-order MA model. One of the core tasks of $ARIMA$ model construction is to determine the appropriate model order (p, q)**?**.

2.4. Data augmentation

In order to test whether the proposed framework can identify best forecasting models, M3 datasets are applied in this paper. Table 1 shows the types of M3 datasets. However, there are only 3003 time series in M3 competition. So a data augmentation method is considered to extend the datasets.

In the research of time series forecasting, data augmentation is also widely used, which can be regarded as the prior knowledge about data invariance injection for some transformations. The enhanced data can expand the training set, prevent over fitting and improve the robustness of the deep learning model(Adhikari & Agrawal, 2014). Perm, for example, is a simple method to disturb the time position of events in a random window. In order to disturb the position of data in a single window, the time series are divided into N segments with the same length, and then randomly arrange N fragments to generate a new sequence. Time warping (TimeW) is also a way to interfere with time position. By smoothing the time interval between the samples, the time position of the samples is changed. Whether the original label is retained depends on the magnitude of the distortion change. Scaling changes the length of data in the window by multiplying the scaling factor, while amplitude distortion changes the size of each sample by convoluting the data window, so that the smooth curve changes around a sample. In addition, jitter is also a method of data enhancement by increasing noise. These data enhancement methods can improve the robustness and generalization ability of the training model and improve the performance. Finally, crop is similar to image clipping in(Heaton & Jeff, 2017) to reduce the dependence on event location. In addition, using random position for clipping in different periods will get an optimal sliding window step. It is worth noting that clips may retain non information areas, resulting in label changes. Compared with image recognition, small changes caused by dithering, zooming, clipping, twisting and rotation may not change the data label.

In this paper, window slicing method of Le Guennec et al.(Guennec et al., 2016) are applied to extract multiple small

windows from a single window, and shorten part of the data window to augment the data. One of the advantages of our framework is to expand the smaller datasets to meet the experimental conditions. At present, most public datasets or real life datasets, such as agricultural product prices, financial series, are often limited in size. In order to solve this problem, this paper proposes a data augmentation technology based on the original datasets to avoid over fitting and improve the generalization ability. For massive datasets with rich training data, data expansion may not be necessary. The proposed data augmentation of window slicing as follow: For time series $T = t_1, t_2, .., t_n$, window slice is the fragment of original time series, defined as $S_{i:j} = t_i, t_{i+1}, ..., t_j$, $1 \leq i \leq j \leq n$. Assuming that the length of a given time series is n and the slice length is s, our slicing operation will generate a set of $n - s + 1$ slice time series:

$$Slicing(T, s) = S_{1:s}, S_{2:s+1}, ..., S_{n-s+1:n} \tag{3}$$

where all the time series in $Slicing(T, s)$ have the same label as their original time series T does. In this paper, because the length of each time series in $M3$ datasets is different, the value of S is variable. We choose $s = n - 2$. Therefore, the enhanced time series becomes three times the original one. The reason why we choose the multiple of data augmentation $m = 3$ is that the best window slice length in(Guennec et al., 2016) is 90% of the original.
Later, Mooseop Kim et al.(Kim & Chi, 2020) compared the effect of data augmentation methods with different window slicing ratio using sensor data, and show this by figures. The results show that when the scaling factor is 0.1, that is, the length of time series slice window is 90% of the original length, the classification accuracy is the highest. On the basis of(Kim & Chi, 2020), a mathematical expression is fitted according to the known data and orange line in figure 2

$$y = a \cdot e^{\frac{-(x-0.1)^2}{b}} \qquad 0 \ll x \ll 1 \tag{4}$$

Where y is the classification rate, x is scaling factor, a and b are constants. It can be concluded that the local Gaussian function reaches the maximum when scaling factor $x = 0.1$.
Because in this case, the sliced time series not only meets the data augmentation, but also retains the periodicity, trend and other features of the original series. The length of M3 time series selected in this paper is mostly about 20, so when the augmentation factor m = 3, the length of original time series is 18, which satisfies the condition in(Guennec et al., 2016). The data augmentation method used in this paper is shown in Figure 3.
All time series are sliced in a given training datasets and these sliced time series are regarded as independent training

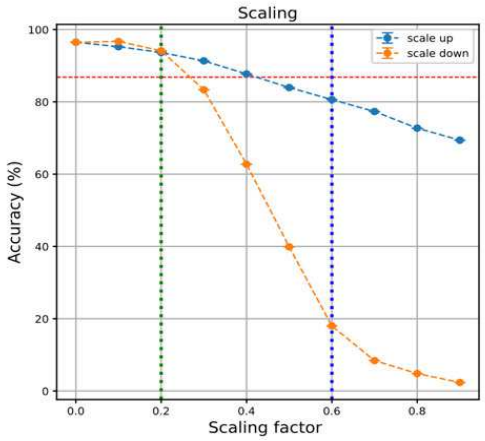

Figure 2: The relationship between classification accuracy and scaling factor in reference(Kim & Chi, 2020)

data. The experimental results show that the sliced label (best forecasting model) may not consistent with the original one.

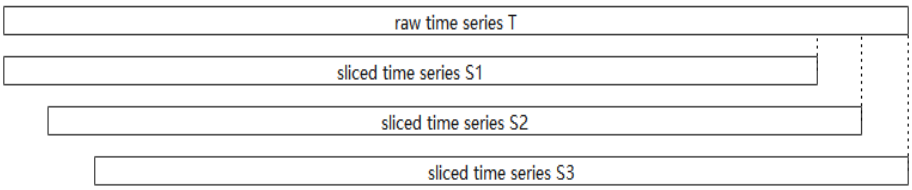

Figure 3: Data augmentation diagram when slice window is 3

3. Time series image and Convolution neural network

3.1. Time series image

This paper uses four algorithms to transform time series into images. They are Gramian Angle Summation Field(GASF), Gramian Angle Difference Field(GADF)(Wang & Oates, 2015), Markov Transition Field(MTF)(Campanharo et al., 2011) and Recurrence Plot(RP)(Eckmann, 1987).

3.1.1. Gramian Angular Field

Gramian Angular Field(GAF) can be divided into Gramian Angular summation field (GASF) and Gramian Angular difference field (GADF). In GAF(Wang & Oates, 2015), the polar coordinate system is used to represent the time series but not traditional Cartesian coordinate system. In the Gramian matrix, each element is actually the cosine or sine of the sum of angles. Given a time-series $X = x_1, x_2, ..., x_n$ with length n, normalize X so that all values are scaled at $[-1, 1]$ or $[0, 1]$ by:

$$\tilde{x}^i_{-1} = \frac{(x_i - max(X)) + x_i - min(X)}{max(x) - min(x)} \tag{5}$$

or

$$\tilde{x}^i_0 = \frac{(x_i - -min(X)}{max(x) - min(x)} \tag{6}$$

Therefore, by encoding the value of the time series $\tilde{X}$ as angular cosine and the time point as radius, the normalized time series can be expressed in polar coordinates, and the formula is as follows:

$$\begin{cases} \phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X}^i_{-1} \\ r = \frac{t_i}{N}, t_i \in N \end{cases} \tag{7}$$

In the equation above,t_i is the time point and N is a constant parameter used to regularize the radius of the polar coordinate system, which is a novel time series visualization method. In the Cartesian coordinate system, the area formula is expressed as:

$$S_{i,j} = \int_{x(i)}^{x(j)} f(x(t))\, dx(t), \tag{8}$$

among that

$$S_{i,i+k} = S_{j,j+k} \tag{9}$$

If $f(x(t))$ has the same values on $[i, i + k]$ and $[j, j + k]$. However, in polar coordinates, if the area is defined as

$$S'_{i,j} = \int_{\phi(i)}^{\phi(j)} r[\phi(t)]^2\, d\phi(t), \tag{10}$$

Then $S'_{i,i+k} \neq S'_{j,j+k}$. That is, the area formed in the polar coordinate system from time point i to time point j depends not only on the time interval $|i - j|$, but also on the absolute values of i and j.

Rescaled data in different intervals have different angular bounds. [0, 1] corresponds to the cosine function in $[0, \frac{\pi}{2}]$, while cosine values in the interval [-1,1] fall into the angular bounds $[0, \pi]$. The formula of GAF is as follows:

$$G = \begin{bmatrix} cos(\phi 1 + \phi 1) & cos(\phi 1 + \phi 2) & \dots & cos(\phi 1 + \phi n) \\ cos(\phi 2 + \phi) & cos(\phi 2 + \phi 2) & \dots & cos(\phi 2 + \phi n) \\ \vdots & \vdots & \dots & \vdots \\ cos(\phi n + \phi 1) & cos(\phi n + \phi 2) & \dots & cos(\phi n + \phi n) \end{bmatrix} \tag{11}$$

The Gramian Angular Summation Field (GASF) and Gramian Angular Difference Field (GADF) are defined as follows:

$$\begin{aligned} GASF &= [cos(\phi_i + \phi_j)] \\ &= \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^{2'}} \cdot \sqrt{I - \tilde{X}^2} \end{aligned} \tag{12}$$

$$\begin{aligned} GADF &= [sin(\phi_i + \phi_j)] \\ &= \sqrt{I - \tilde{X}^{2'}} \cdot \tilde{X} - \sqrt{I - \tilde{X}^2} \cdot \tilde{X}' \end{aligned} \tag{13}$$

I is the unit row vector [1, 1, ..., 1]. After transforming to the polar coordinate system, we take time-series at each time step as a 1-D metric space. By defining the inner product $< x, y >= x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ and $< x, y >= \sqrt{1 - x^2} \cdot y - \sqrt{1 - y^2} \cdot x$ two types of Gramian Angular Fields(GAFs) are actually quasi-Gramian matrices $[\langle x, y\langle]$:

$$G = \begin{bmatrix} [\langle \tilde{x}_1, \tilde{x}_1 \langle] & \dots & [\langle \tilde{x}_1, \tilde{x}_n \langle] \\ \vdots & \dots & \vdots \\ [\langle \tilde{x}_n, \tilde{x}_1 \langle] & \dots & [\langle \tilde{x}_n, \tilde{x}_n \langle] \end{bmatrix} \tag{14}$$

The GAFs have several advantages. First, they provide a way to preserve temporal dependency, since time increases as the position moves from top-left to bottom-right. The GAFs contain temporal correlations because $G_{|i-j|=k}$ represents the correlation when the time interval is $G_{i,i}$. when $k = 0$, it is a special case. It represents the angle containing only the main diagonal element and the original value, the GAFs are large because the size of the Gramian matrix is $n \times n$ when the length of the raw time-series is n.

The transformation maintains the time dependence between the values, and provides time correlation due to the superposition in the direction relative to the time interval, athe bijection matrix is formed. Therefore, the inverse function of the original data is an absolute reconstruction. As shown in Figure 4.

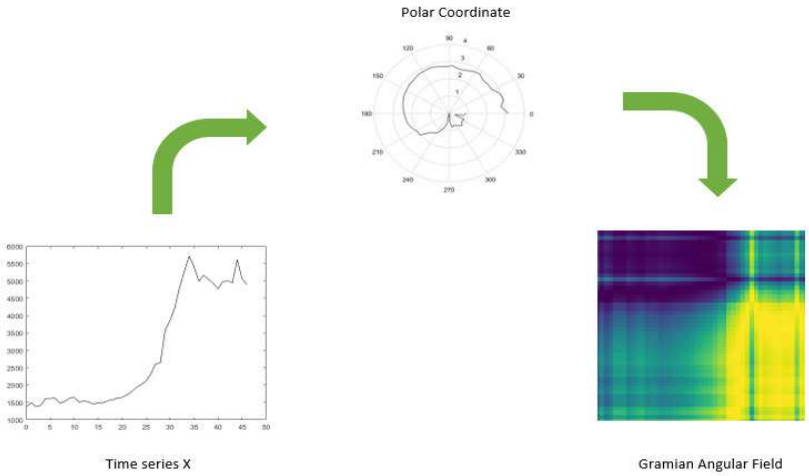

Figure 4: Illustration of the proposed encoding map of Gramian Angular Fields. Taking GADF as an example, the formation of GASF is similar. X is a sequence of rescaled time-series in the M3 datasets and transform X into a polar coordinate system by eq. (7) and finally calculate its GASF images with eqs. (12)

3.1.2. Markov Transition Field

We get inspiration from Campanharo et al.(Campanharo et al., 2011). time series X determine Q quantile bins, and assign each x_i to the corresponding storage unit $q_i(j \in [1, Q])$. Thus we construct a $Q \times Q$ weighted adjacency matrix W by counting transitions among quantile bins in the manner of a first-order Markov chain based on the time axis. $w_{i,j}$ is given by the transition probability of a point in quantile q_j is followed by a point in quantile q_i.After normalization by $\sum_j \omega_{i,j} = 1$ W is the Markov transition matrix. It is irrelevant to the distribution of X and temporal dependency on time steps t_i.However, our experimental results on W demonstrate that getting rid of the temporal dependency results in too much information loss in matrix W. In order to overcome this disadvantage, the mathematical formula of Markov transfer field (MTF) is as follows:

$$M = \begin{bmatrix} \omega_{ij|x_1 \in q_i, x_1 \in q_j} & \omega_{ij|x_1 \in q_i, x_2 \in q_j} & \cdots & \omega_{ij|x_1 \in q_i, x_n \in q_j} \\ \omega_{ij|x_2 \in q_i, x_1 \in q_j} & \omega_{ij|x_2 \in q_i, x_2 \in q_j} & \cdots & \omega_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{ij|x_n \in q_i, x_1 \in q_j} & \omega_{ij|x_n \in q_i, x_2 \in q_j} & \cdots & \omega_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix} \tag{15}$$

A $Q \times Q$ Markov transition matrix is established by dividing the data into Q quantile bins. $M_{i,j}$ in the MTF denotes the transition probability of $q_i \rightarrow q_j$.That is, by considering the time and location, the matrix W is extended to an MTF matrix containing the transition probability on the magnitude axis. By forming the probability of quantiles from time step i to time step j at each pixel M_{ij} , the essence of MTF is the multi span transition probability of coded time series. $M_{i,j|i-j|=k}$ represents the transition probability of two points with time interval k. Figure 5 shows the procedure to encode time-series to MTF.

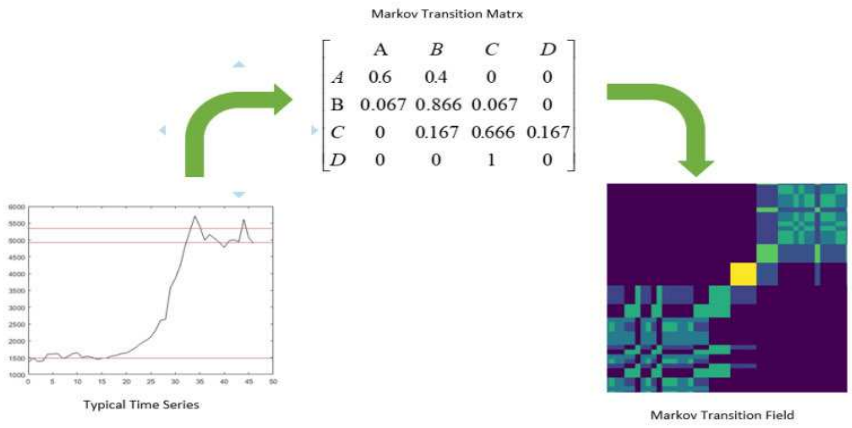

Figure 5: Illustration of the proposed encoding map of Markov Transition Fields. X is a sequence of time-series in the M3 dataset . X is first discretized into Q quantile bins. In this image, we take Q = 4. Then we calculate its Markov Transition Matrix W and finally build its MTF with eq. (15)

3.1.3. Recurrence plot

In this part,recurrence plots (RP) is applied to transform time series into images. The recurrence plots provides a method for visualizing the periodicity of trajectories through phase space(Eckmann, 1987), and it can contain most of the relevant dynamic features in the time series.The recurrence plots of time series x can be expressed as:

$$R(i, j) = \Theta(\epsilon \| x_i - x_j \|) \tag{16}$$

where R(i,j) is the element of recurrence matrix R; i indexes time on the x-axis of the recurrence plot, j indexes time on the y-axis. ϵ is a predefined threshold, and $\Theta(\cdot)$ is the Heaviside function. In short, a black spot will appear when the distance from x_i and x_j are smaller than ϵ. The following modified RP is used to balance binary output with thresholdless RP (Thiel et al., 2004).

$$R(i, j) = \begin{cases} \epsilon, & \| x_i - x_j > \epsilon \| \\ \| x_i - x_j \|, & otherwise \end{cases} \tag{17}$$

Compared with the binary method, it generates more dense points and can generate color images. As we can see in Fig 6.

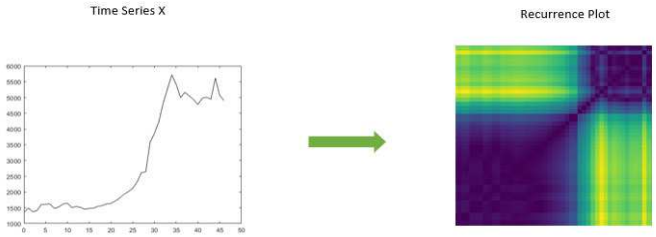

Figure 6: Illustration of the proposed encoding map of recurrence plots. X is a sequence of time-series in the M3 dataset. We finally build its RP with eq. (17).

3.2. Data augmentation time series image(DA-TSI)

This paper combines data augmentation method with time series image transformation, and proposes an integrated and innovative data augmentation time series imaging(DA-TSI) algorithm. The time series image augmentation algorithm proposed in this paper is different from the traditional image augmentation technology. In the field of image

recognition, data augmentation has become a convention. Most of the most advanced convolution neural network (CNN)(Y. Lecun & Haffner, 2013) structures use some form of data expansion. For example, Alexnet(Krizhevsky, 2012) is one of the first deep CNN that created a record benchmark on Imagenet large-scale visual recognition challenge (ILSVRC) datasets(Russakovsky et al., 2015), uses clipping, mirroring, and color augmentation to optimize the network. Other examples include the original proposal for the VGG network(Simonyan & Zisserman, 2014), which uses scaling and clipping, the Resnet work(He et al., 2016) using scaling, clipping, and color augmentation, Densenet(Huang et al., 2016) using translation and mirroring, and perception network using clipping and mirroring. The DA-TSI algorithm proposed in this paper preprocesses the time series (see Section 2.4 for details), and then uses the time series image conversion algorithm to generate images.The advantage of this is that it can better protect the features of the original time series from being lost after image augmentation. Because the image generated by time series is close to mosaic, if image augmentation is carried out on the basis of mosaic, there will be a lot of time series feature loss. Therefore, DA-TSI algorithm has advantages in theory.

3.3. Convolution neural networks

Convolution neural networks(CNNs) have made remarkable achievements in image classification(Technicolor et al., 2012), natural language processing(Devlin et al., 2018) and reinforcement learning(Silver D, 2016). For time series forecasting, CNNs can reflect the subtle differences of underlying datasets and customize the corresponding architecture(Baxter, 2000) and complex data representation(Bengio et al., 2012) to reduce the work of manual feature engineering and model design.
In this paper, Three deep learning frameworks are applied to test the generalization performance of the proposed algorithm. The three deep convolution neural network models have different network depth and network structure, so if the algorithm can perform well in the three convolution neural network models, it can be applied to other deep learning models.

3.3.1. Basic idea of residual learning

He et al.(He et al., 2016) put forward an improved CNN model for image classification, which is called deep residual network. The main difference between residual network and traditional CNN is that they have different network structures and information transmission modes, as shown in Figure 7. For the traditional CNN model, the input layer, convolution layer, pooling layer and output layer are combined in a cascade manner. But for the rest of the network, it has a shortcut that connects input and output directly together. Mathematically speaking, different from the direct approximation of basic function H (x), residual learning emphasizes the fitting of residual mapping f (x)

$$F(x) = H(x) - x \tag{18}$$

The special mapping of residual network block is $F(x) + x$, which is the output of a traditional CNN, namely $H(x)$. However, as He et al. pointed, compared with the original mapping $H(x)$, the fitting residual mapping $F(x)$ is more effective, especially when $H(x)$ is an identity or approximate identity mapping. The characteristics of the residual network will increase the depth greatly, but will not reduce the classification accuracy of the network.

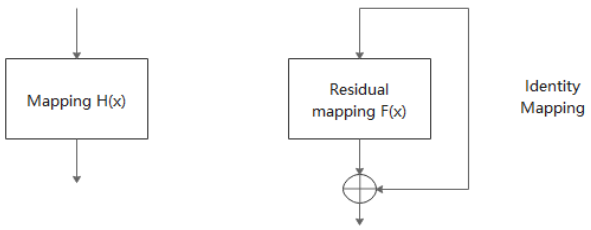

Figure 7: Basic building blocks in different CNN models. Left: a basic building block in a typical CNN model. Right: a basic building block in a residual network

3.3.2. Basic idea of Visual Geometry Group network

Visual Geometry Group network (VGGnet) is a multi-layer neural network. VGGnet is very useful because it will 3×3 size convolution layer is installed on the top, which increases the depth of the network. In order to reduce the size of convolution kernel, max pool layer is used in VGGnet. There are 4096 neurons in two FC layers. As shown in Figure 8.

In the training stage, convolution layer is used to extract features, and maximum pool layer and partial convolution layer are used to reduce feature dimension. In the first convolution layer, 64 kernels ($3 \cdot 3$ filter size) convolution kernel. All connected layers are used to construct eigenvectors. Finally, in the test phase, Softmax activation function is used to classify the images.

VGGnet systematically studies the influence of network depth on classification performance, and constructs a deeper structure on the basis of shallow layer (Jaworek-Korjakowska J, 2019).

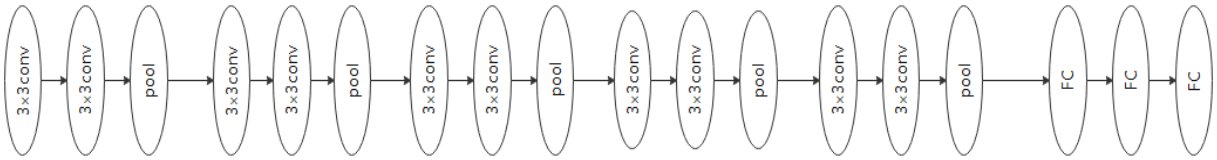

Figure 8: The architecture of Vggnet.

3.3.3. Brief introduction of Densenet

Densenet(Zhu & Newsam, 2017) is a CNN architecture proposed in recent years. It has a new connection mode: dense block connection. In dense blocks, each layer is connected to all other layers. In this case, all layers can access the output features of the previous layer, which enhances the correlation of features. The effect of this framework makes the model is more dense to prevent over fitting. All these excellent features make Densenet more suitable for image recognition, which not only achieves the most advanced performance, but also does not need pre training or additional post-processing.

Traditional CNNs, such as FlowNets, calculate the output of the l^{th} layer by applying a nonlinear transformation H to the previous layer's output x_{l-1}

$$x_l = H_l(x_{l-1}) \tag{19}$$

After the convolution layer and pooling layer processing, the traditional convolution neural network can obtain semantic features at the top, but these features are too rough, and the fine image details often disappear in the network. In order to improve the information exchange between layers, Densnet an improved connection mode: the first layer takes the feature maps of other layers as input:

$$x_l = H_l(x_0, x_1, \ldots, x_{l-1}) \tag{20}$$

where $[x_0, x_1, \ldots, x_{l-1}]$ is a single tensor formed by concatenating the output feature maps of the previous layer. It is a single tensor formed by concatenating the output eigenvectors of the previous layer. In this way, even the last layer network can share features with the first layer. The loss function directly supervises each layer through quick connection. As we can see in Fig 9.

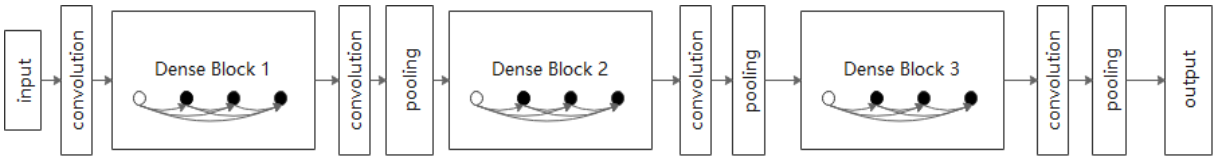

Figure 9: The architecture of Densenet.

4. Experiment and result analysis

In this section, the experimental effect of data augmentation time series image(DA-TSI) algorithm are applied in a small number of datasets, and make theoretical analysis combined with the experimental results. Finally, DA-TSI algorithm is compared with time series image(TSI) baseline algorithm(GAFs, MTF, RP) and traditional machine learning classification algorithm support vector machine (SVM).

4.1. Baseline algorithm

The TSI of deep CNN without data augmentation and traditional machine learning classifier SVM are applied as the base comparison algorithm in this paper. In this paper, control variables are controlled in order to achieve scientific experimental results. Three different network structures of Resnet-18, VGG-11 and Densenet-121 are used as the general training models of deep learning method and applied to the benchmark deep CNN time series image method. In machine learning method, SVM is used as a classifier to train the time series after data augmentation.

4.2. Datasets

Time series of M3 datasets is applied in this paper. M3 datasets includes more complex Micro-economic and Industrial data, and is conducive to verify the generalization ability of the proposed method. The specific classification is shown in Table 1.

Table 1: Category of 3003 datasets of M3 competition

Types	Yearly	Quarterly	Monthly	Other	Total
Micro	146	204	474	4	828
Industry	102	83	334		519
Macro	83	336	312		731
Finance	58	76	145	29	308
Demographic	245	57	111		413
Other	11		52	141	204

4.3. Model evaluation

The evaluation criteria to verify the correctness of the model selection are the classification accuracy rate obtained by comparing the label of the test set with the optimal label, and the Mean Absolute Percentage Error (MAPE) obtained according to the model selection results. Therefore, this paper has two standards. One is classification accuracy, the other is forecasting error.
The classification accuracy can be expressed as

$$accuracy = \frac{(TP + TN)}{All} \tag{21}$$

Where True positives(TP) is the number of positive examples correctly divided, and True negatives(TN) is the number of negative cases correctly divided.
The forecasting error used in this paper is the Mean absolute percentage error (MAPE). The benchmark model in this paper is four different single image generation methods and six econometric model methods.

$$MAPE = \sum_{t=1}^{n} |\frac{Y_t - \hat{Y}_t}{Y_t} \times \frac{100}{n} \tag{22}$$

where Y_t is the real value of the time-series at point t, $\hat{Y}_t$ is the forecast, n is the forecasting horizon.

4.4. Parameter setting

In our experiment, we used Python 3.7 and R. The size of four kinds of single pictures is 359×359, and after resizing the size of combined image(GADF-GASF-MTF-RP) is also 359×359. The parameters for pre-trained CNN models are set as follows:
Dimension of the output of the pre trained VGG-11bn: 1000.
Dimension of the output of the pre trained resnet-18: 512.
Dimension of the output of the pre trained densenet-121: 1000.
The iteration rate of CNNs is 0.001, and the batch size is 16.

4.5. Result analysis

Three deep learning models Resnet-18, Densenet-121 and VGG-19 with different network structures are applied to M3 datasets. M3 datasets is divided into training set, verification set and test set according to the ratio of 8:1:1. Then four kinds of TSIs generated from M3 datasets and four DA-TSIs are input into three convolution neural networks to compare the classification accuracy (model selection accuracy). In order to save space, this algorithm did not show the complete results on the single channel image.
Through the analysis of the experimental results, some conclusions come to us:
1. As we can see Table 2, 3 and 4, the average classification rates of GADF and GASF under three depth CNN are 6.6% and 5.5% higher than MTF respectively. And after data augmentation, the average classification rates of DA-GADF and DA-GASF under three depth CNNs were 2.8% and 3.6% higher than that of DA-MTF respectively. In addition to the potential risk of over fitting, we find that after three different CNNs training, the classification rate of MTF in the test set is generally slightly lower than that of GAFs under the same algorithm. This may be due to the uncertainty of the inverse mapping of MTF relative to GAF. Although both GAF and MTF time series image maps after time series standardization are epimorphic, on the $[0, 1]$ standardized time series, the mapping function of GAF is bijective, while MTF is not bijective. The original time series can be reconstructed from the diagonal of GAFs, but it is very difficult to roughly recover the signal from MTF.
2. Experimental results show that the DA-TSI algorithm proposed in this paper is basically suitable for all deep learning models, different time series visualization methods and different step sizes, and can improve the classification rate of the original TSI algorithm. As we can see the Table 2, 3 and 4, When the steps are 1, 3 and 6, the classification rates of time series images after data enhancement in CNN classifier are improved by 2.0%, 5.7% and 11.0% respectively compared with traditional time series images. On the whole, with the increase of forecasting step, the classification accuracy of DA-TSI algorithm proposed in this paper will be improved. The reason may be that with the increase of forecasting step, the fluctuation of time series increases, so the discrimination of time series images is more obvious, which is conducive to the higher classification rate of CNNs. It shows that this algorithm has better effect on the medium and long-term forecasting of small datasets.
3. From tables 2 and 6, the classification rate of DA-TSI-MTF algorithm is improved compared with the original MTF algorithm after input into Densenet and VGG network, but the forecasting error MAPE increased by 0.14 and 0.06. The reason is that there may be a huge forecasting error in some wrongly selected forecasting models, which will affect the overall average error, resulting in the phenomenon that the classification rate increases but the error also raises. On the contrary, the classification rate of DA-TSI-GASF algorithm input into VGG network is slightly lower than that of TSI-GASF algorithm, but the forecasting error MAPE is reduced 0.27, which also shows that the pros and cons of an algorithm should be judged by multiple criteria.
4. From table 4 and 8, when the forecasting step is 6, the classification effect of various CNNs for DA-TSI image algorithm has best results compared with other steps, but the error is also the largest. Because even if the classification rate increases due to the increase of step size, that is, the accuracy of model selection increases, but the step size is 6, it belongs to medium and long-term forecasting, so even the forecasting result of the optimal forecasting model has a large standard deviation.
5. From the results of classification rate, the highest classification accuracy (model selection accuracy) of DA-TSI algorithm proposed in this paper has increased significantly compared with the traditional TSI algorithm. At the same time, compared with table 9, after data augmentation of the same scale, the DA-TSI-CNN algorithm proposed in this paper is obviously superior to the traditional machine learning algorithm. From the perspective of prediction error MAPE, in order to show the advantages of this algorithm more intuitively, we simply average the errors of all DA-TSI

algorithms and TSI algorithms combined with different CNNs, as shown in Figure 10. In different steps, the error of the algorithm is the lowest, which can fully demonstrate the superiority of the algorithm proposed in this paper.
6. In order to show the advantages of this algorithm more intuitively, we combine tables 2, 3 and 4, as shown in table 10. It can be seen from the table that when the step size is 1, the advantage of the proposed data augmentation imaging (DA-TSI) algorithm is not obvious compared with the traditional time series imaging (TSI) algorithm, and the maximum improvement of3.6% is reflected in the Densenet network. On the one hand, the Densenet network is more suitable for short-term time image classification, On the other hand, its dense network structure has better classification effect. With the increase of step size, the classification effect of DA-TSI algorithm is better on different CNNs models. When the step size is 6, the DA-TSI algorithm achieves 54.2% classification rate when combined with VGG network, which compared with the six classification problem with an average classification rate of 16.7% and TSI-VGG with an average classification rate of 42.4%, this method has a significant improvement. It can be more intuitive in Figure 11.

Table 2: Comparison of classification rates between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=1

Classification rate	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	0.392	**0.395**	0.405	**0.435**	0.349	**0.375**	0.415	**0.429**
Densenet	0.385	**0.435**	0.352	**0.425**	0.346	**0.369**	0.419	**0.419**
VGG	0.392	**0.395**	**0.449**	0.435	0.362	**0.382**	0.412	**0.429**

Table 3: Comparison of classification rates between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=3

Classification rate	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	0.435	**0.468**	0.379	**0.445**	0.309	**0.419**	0.369	**0.415**
Densenet	0.379	**0.445**	0.379	**0.468**	0.362	**0.449**	0.399	**0.462**
VGG	**0.469**	0.419	0.435	**0.438**	0.282	**0.445**	0.401	**0.419**

Table 4: Comparison of classification rates between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=6

classification rate	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	0.449	**0.542**	0.422	**0.508**	0.385	**0.522**	0.412	**0.515**
Densenet	0.429	**0.538**	0.435	**0.545**	0.389	**0.495**	0.442	**0.551**
VGG	0.445	**0.555**	0.422	**0.575**	0.395	**0.495**	0.432	**0.542**

Table 5: The average value of the forecasting error MAPE of the six forecasting models on the test set under different step sizes

Forecasting Model	ARIMA	ETS	WN	RWd	RW	THETA	Average
MAPE(h=1)	**11.65**	12.1	13.72	12.83	12.92	11.98	12.53
MAPE(h=3)	15.45	15.42	18.03	17.03	17.13	**15.22**	16.38
MAPE(h=6)	19.44	20.07	22.56	21.74	21.76	**19.33**	20.82

Table 6: Comparison of test set forecasting error MAPE between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=1

Classification rates	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	11.41	**11.32**	11.54	**11.41**	11.23	**11.21**	11.48	**11.47**
Densenet	11.36	**11.13**	11.49	**11.18**	**11.18**	11.32	11.52	**11.37**
VGG	11.32	**11.10**	11.42	**11.15**	**11.29**	11.35	11.06	**11.03**

Table 7: Comparison of test set forecasting error MAPE between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=3

Classification rates	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	15.33	**13.82**	15.44	**13.99**	15.57	**13.79**	15.33	**14.01**
Densenet	15.08	**14.04**	15.45	**14.08**	15.31	**14.09**	15.57	**13.95**
VGG	15.42	**14.03**	15.27	**14.24**	15.76	**13.67**	15.34	**14.22**

Table 8: Comparison of test set forecasting error MAPE between DA-TSI algorithm and traditional image algorithms by three different CNNs when the step size h=6

Classification rates	GADF	DA-TSI-GADF	GASF	DA-TSI-GASF	MTF	DA-TSI-MTF	RP	DA-TSI-RP
Resnet	16.68	**16.36**	16.68	**16.26**	16.77	**16.61**	16.53	**16.02**
Densenet	16.51	**15.94**	16.68	**16.25**	16.93	**16.22**	16.84	**16.15**
VGG	16.38	**15.94**	16.31	**15.92**	17.01	**16.32**	16.41	**16.04**

Table 9: The average classification rate of TSI algorithm and DA-TSI algorithm in different CNNs

Rates	TSI-Res	DA-TSI-Res	Improve	TSI-Dense	DA-TSI-Dense	Improve	TSI-VGG	DA-TSI-VGG	Improve
h=1	0.390	0.409	0.019	0.376	0.412	0.036	0.404	0.410	0.006
h=3	0.373	0.437	0.064	0.380	0.456	0.076	0.397	0.448	0.051
h=6	0.417	0.522	0.105	0.424	0.532	0.108	0.424	0.542	0.118

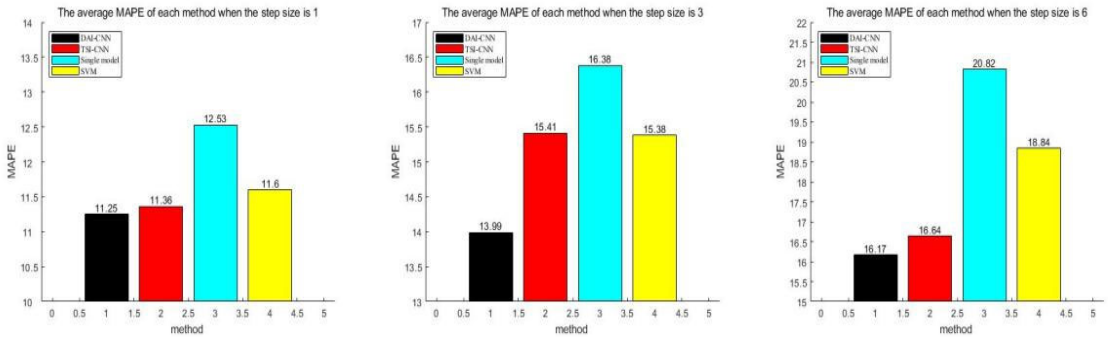

Figure 10: The average MAPE of DA-TSI-CNN, TSI-CNN, Single model and SVM in different step size

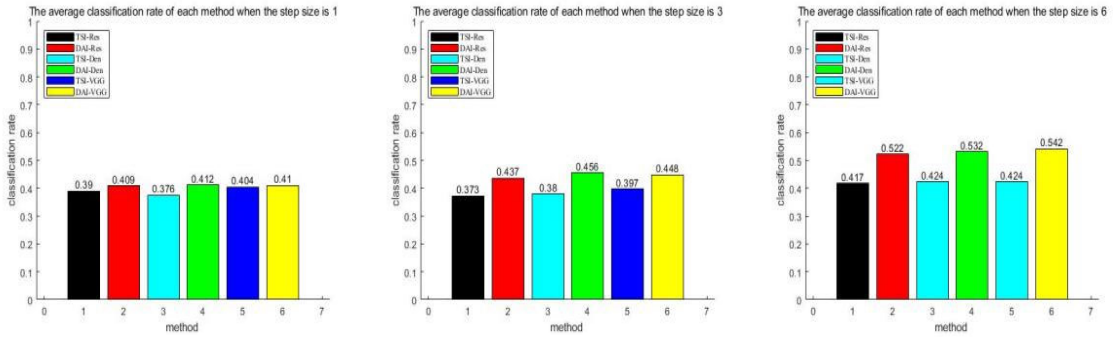

Figure 11: The average classification rate of different method in different step size

5. Conclusion and future work

This work investigated meta-learning based CNNs time series image for time series prediction with the aim to link problem-specific knowledge to well performing forecasting methods and apply them in similar situations. In the improved meta learning framework proposed in this paper, we use computer vision algorithm instead of Feature Engineering, and use convolution neural network to automatically extract features from time series images, so as to reduce the workload. In addition, in order to deal with the over fitting problem of small datasets in deep convolution network, we propose data augmentation imaging (DA-TSI) algorithm, which can effectively solve the problem of over fitting caused by insufficient data in real life. M3 datasets is applied to this algorithm, the experimental results show that this algorithm can automatically extract time series features, and has stronger advantages than the original time series image algorithm and machine learning algorithm.
In the future work, we will continue to explore the significance of Gaussian function between data augmentation sliding window cut length (multiple of data increase) and time series classification. And try other classification methods to enrich the meta learning framework proposed in this paper.

Ethical approval

This artical does not contain any studies with human participants or animals performed by any of the authors.

Funding

This work was supported by the National Natural Science Foundation of China (71971089,72001083).

Conflict of interest

All Authors declar that they have no conflict of interest.

Informed Consent

All submitted manuscripts must include a statement that informed consent, if necessary.

Author information

Affiliations
South China Agricultural University,School of mathematics and information,Guangzhou,510630.
Authors
Wentao Jiang
Liwen Ling
Dabin Zhang
Ruibin Lin
Liling Zeng
Corresponding author
Correspondence to Dabin Zhang

Authorship Contributions

Wentao Jiang Software, Data curation, Writing, Methodology. **Dabin Zhang** Writing-review, Supervision. **Liwen Ling** Conceptualization, Methodology, Writing - review, Supervision. **Ruibin Lin** Software, Data curation.

Additional information

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Adhikari, R., & Agrawal, R. K. (2014). A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing and Applications*, *24*, 1441–1449.

Assimakopoulos, V., & Nikolopoulos, K. (2000). The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, *16*, 521–530.

[Aut, N. K., & [Aut, F. P. (2018). Mapa: Multiple aggregation prediction algorithm, .

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, *12*.

Bengio, Y., Courville, A., & Vincent, P. (2012). Representation learning: A review and new perspectives, .

Brown, R. G. (1977). Forecasting: Issues and challenges for marketing management. *Journal of Marketing*, *41*, 24–38.

Campanharo, A., Sirer, M. I., Malmgren, R. D., Ramos, F. M., Amaral, L., & Perc, M. (2011). Duality between time series and networks. *Plos One*, *6*, e23378.

Chris, & Chatfield (1993). "rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations": Fred collopy and j. scott armstrong, management science, 38 (1992) 1394–1414. *International Journal of Forecasting*, .

Connor, J. T., Martin, R. D., & Atlas, L. E. (2002). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, *5*, 240–254.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, .

Eckmann, J. P. (1987). Kamphorst, s.o., ruelle, d.: Recurrence plots of dynamical systems. europhys. lett. (epl) 4, 973-977, .

F Petropoulos, A. V., Makridakis S (2014). Horses for courses' in demand forecasting. *European Journal of Operational Research*, *237*, 152–163.

F.A. Gers, J., D. Eck (2001). Applying lstm to time series predictable through time-window approaches. In *International Conference on Artificial Neural Networks*.

Fiorucci, L. F. Y. B. F. M. J. A., J. A. (2016). forectheta r package manual, .

Guennec, A. L., Malinowski, S., & Tavenard, R. (2016). Data augmentation for time series classification using convolutional neural networks, .

Hasibi, R., Shokri, M., & Dehghan, M. (2019). Augmentation scheme for dealing with imbalanced network traffic classification using deep learning, .

Hatami, N., Gavet, Y., & Debayle, J. (2019). Bag of recurrence patterns representation for time-series classification. *Pattern Analysis and Applications*, *22*, 877–887.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heaton, & Jeff (2017). Ian goodfellow, yoshua bengio, and aaron courville: Deep learning. *Genetic Programming and Evolvable Machines*, (pp. 1–3).

Huang, G., Liu, Z., Laurens, V., & Weinberger, K. Q. (2016). Densely connected convolutional networks. In *IEEE Computer Society*.

Jaworek-Korjakowska J, G. M., Kleczek P (2019). Melanoma thickness prediction based on convolutional neural network with vgg-19 model transfer learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, *33*, 345–358.

Kim, M., & Chi, Y. J. (2020). Label-preserving data augmentation for mobile sensor data. *Multidimensional Systems and Signal Processing*, .

Krizhevsky, A. (2012). Learning multiple layers of features from tiny images, .

Kuck, M., Crone, S. F., & Freitag, M. (2016). Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In *2016 International Joint Conference on Neural Networks (IJCNN 2016)*.

Lemke, C., & Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, *73*, 2006–2016.

Macready (). Macready, w.g.: No free lunch theorems for optimization. ieee transactions on evolutionary computation 1(1), 67-82, .

Makridakis, Spyros, Hibon, & Michèle (2001). Response to the commentaries on 'the m3-competition: results, conclusions and implications'. *International Journal of Forecasting*, *17*, 581–584.

Meade, N. (2000). Evidence for the selection of forecasting methods. *Journal of Forecasting*, *19*, 515–535.

Mittelman, R. (2015). Time-series modeling with undecimated fully convolutional neural networks. *Computer Science*, .

Morwitz, V. G., Steckel, J. H., & Gupta, A. (2007). When do purchase intentions predict sales? *International Journal of Forecasting*, *23*, 347–364.

Naduvilvadukootu, S., Angryk, R. A., & Riley, P. (2017). Evaluating preprocessing strategies for time series prediction using deep learning architectures, .

Prudencio, R., & Ludermir, T. B. (2004). Meta-learning approaches to selecting time series models. *Neurocomputing*, *61*, 121–137.

Rachana, W., Suvarna, M., & Sonal, G. (2010). Use of arima model for forecasting pigeon pea production in india, .

Rice, J. (1976). The algorithm selection problem4, .

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Shah, C. (1997). Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting*, *13*, 489–500.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*.

Silver D, M. C. G. A. S. L. v. d. D. G., Huang A (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*, 484–503.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Computer Science*, .

Smith-Miles. (2009). *Cross-disciplinary perspectives on meta-learning for algorithm selection*. Cross-disciplinary perspectives on meta-learning for algorithm selection.

S. Dasgupta, T. (2017). Nonlinear dynamic boltzmann machines for time-series prediction. *Proceedings of the AAAI*, (pp. 1833–1839).

Talagala, T. S., Hyndman, R. J., & Athanasopoulos, G. (2018). Meta-learning how to forecast time series. *Monash Econometrics and Business Statistics Working Papers*, .

Technicolor, T., Related, S., Technicolor, T., & Related, S. (2012). Imagenet classification with deep convolutional neural networks, . (p. 1097–1105).

Thiel, M., Romano, M. C., & Kurths, J. (2004). How much information is contained in a recurrence plot? *Physics Letters A*, *330*, 343–349.

Timmermann, A., & Granger, C. (2002). Efficient market hypothesis and forecasting. *International Journal of Forecasting*, *20*, 15–27.

Wang, Z., & Oates, T. (2015). Imaging time-series to improve classification and imputation. *AAAI Press*, (p. 3939–3945).

Widodo, A., & Budi, I. (2013). Model selection using dimensionality reduction of time series characteristics. *Paper presented at the International Symposium on Forecasting, Seoul, South Korea.*, .

Winters, P. R. (1976). Forecasting sales by exponentially weighted moving averages. *Management ence*, *6*, 324–342.

Y. Lecun, Y. B., L. Bottou, & Haffner, P. (2013). Gradient-based learning applied to document recognition, . (pp. 2278–2324).

Zhu, Y., & Newsam, S. (2017). Densenet for dense flow. In *2017 IEEE International Conference on Image Processing (ICIP)*.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- latex.rar