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

Wentao Jiang¹,² · Liwen Ling¹,² · Dabin Zhang¹,² · Ruibin Lin¹,² · Liling Zeng¹,²

Accepted: 10 December 2022 / Published online: 13 February 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

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 meta learning forecasting model selection methods rely on manual selection of features, which leads to low efficiency and lack of objectivity. Therefore, this paper proposes an improved meta learning framework for deep learning time series forecasting model selection. Inspired by computer vision, we transform one-dimensional time series into two-dimensional images, and use convolution neural network 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. The large-scale empirical study on M3 data sets shows that the framework has better model selection accuracy and smaller forecasting error than the recurrent neural network (RNN), back propagation neural network and traditional time series image algorithms. In addition, compared with the traditional time series image method, RNN and BP, the classification rate (model selection accuracy) of this algorithm is improved by 6.5%, 4.4% and 3.2%, respectively.

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 [28]. In recent decades, a great quantity of time series forecasting methods have been applied, aiming to improve the forecast accuracy. However, from the "no free lunch" theorem [26] 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 [33]; Wang, Smith miles and Hyndman [21] 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 [25] provided a new set of time series feature element learning method for NN3 and NN5 data sets, and analyzes the results; Widodo and Budi [42]used principal component analysis to reduce the feature dimension, so as to optimize the forecasting model selection method; Petropoulos et al. [11] proposed the "horse for course" in M3 dataset [27], and counted the effects of 7 different time series features on the performance of 14 good forecasting methods. Recently, Talagala et al. [38] 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 [8] 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 [12]. Naduvil-Vadukootu et al. [29] 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 [34]. 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 [15]. 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. [16] and Wang and Oates [41] 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 Compared with the traditional time series imaging algorithm, RNN and BPNN.

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 [31]. 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 [37] 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 [30] 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 [37] 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 [42] 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 [24] 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 Fig. 1. The CFMS framework proposed in this paper can be divided into four parts, which are time series preprocessing, time series visualization, CNN classification and model selection, and evaluation criterions. First, in the preprocessing stage, time series are divided into training set, verification set and test set. Then perform data augmentation(SWDA) on the training set data, and assign labels to the enhanced time series. The label set used in this paper is six forecasting models (Introduced...
Each time series will calculate the label with the smallest forecasting error among the six alternative models in R. In the time series visualization stage, each time series is transformed into four kinds of two-dimensional images (GASF, GADF, MTF and RP) (Introduced in Sect. 3.1), which is conducive to automatic feature extraction during CNN model training and fully reflects the generalization ability of this algorithm. In the model selection stage, the time-series images are input into three CNN with different depths and structures for training, and the trained CNN classifier is obtained. Then, the new time series in the test set are input into the trained CNN classification, and the labels are automatically output for the new sequences, so as to obtain the best forecasting model of the new time series. Finally, the classification rate and average absolute percentage error (MAPE) are used to evaluate the model. 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=x₁, x₂, ..., xₙ: the classification of n observed time series;
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 xi 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 xi 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[6]. 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[43]. The "ZZZ" model used in this paper refers to the best ETS model automatically selected by R software package according to AIC criteria [3].

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[2]. The algorithm is implemented by R package forctheta[13].

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:

\[ y_t - y_{t-1} = \epsilon_t \]  

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

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:

\[ y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \ldots - \theta_q \epsilon_{t-q} \]  

where \( y_t \) and \( \epsilon_t \) 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 \( \epsilon_t \) is zero and the variance is constant \( \sigma^2 \). 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
A Time Series Forecasting Model...

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   |

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[1]. 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[18] 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.[14] 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 \( T = t_1, t_2, \ldots, t_n \), window slice is the fragment of original time series, defined as \( S_{i:j} = t_i, t_{i+1}, \ldots, t_j, 1 \leq i \leq j \leq n \). 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:

\[
\text{Slicing}(T, s) = S_{1:s}, S_{2:s+1}, \ldots, S_{n-s+1:n}
\]

where all the time series in \( \text{Slicing}(T, s) \) 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 [14] is 90% of the original.

Later, Mooseop Kim et al. [22] 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 [22], a mathematical expression is fitted according to the known data and orange line in Fig. 2

$$y = a \cdot e^{-(x-0.1)^2/b} \quad 0 \ll x \ll 1$$

(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 [14]. The data augmentation method used in this paper is shown in Fig. 3.

All time series are sliced in a given training datasets and these sliced time series are regarded as independent training data. The experimental results show that the sliced label (best forecasting model) may not consistent with the original one.
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) [41], Markov Transition Field(MTF) [7] and Recurrence Plot(RP) [10].

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 [41], 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:

\[
\tilde{x}_{i-1} = \frac{(x_i - \max(X)) + x_i - \min(X)}{\max(x) - \min(x)}
\]

or

\[
\tilde{x}_0 = \frac{(x_i - \min(X))}{\max(x) - \min(x)}
\]

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:

\[
\begin{cases}
\phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X}_{i-1} \\
r = \frac{t_i}{N}, t_i \in N
\end{cases}
\]

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:

\[
S_{i,j} = \int_{x(i)}^{x(j)} f(x(t)) \, dx(t),
\]

among that

\[
S_{i,i+k} = S_{j,j+k}
\]

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

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

$$G = \begin{bmatrix}
cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \ldots & \cos(\phi_1 + \phi_n) \\
cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \ldots & \cos(\phi_2 + \phi_n) \\
\vdots & \vdots & \ddots & \vdots \\
cos(\phi_n + \phi_1) & \cos(\phi_n + \phi_2) & \ldots & \cos(\phi_n + \phi_n)
\end{bmatrix}$$ (11)

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

$$GASF = [\cos(\phi_i + \phi_j)] = \tilde{X} \cdot \tilde{X} - \sqrt{1 - \tilde{X}^2} \cdot \sqrt{1 - \tilde{X}^2}$$ (12)

$$GADF = [\sin(\phi_i + \phi_j)] = \sqrt{1 - \tilde{X}^2} \cdot \tilde{X} - \sqrt{1 - \tilde{X}^2} \cdot \tilde{X}'$$ (13)

$I$ is the unit row vector $[1, 1, \ldots, 1]$. 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 $\langle x, y \rangle = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ and $\langle x, y \rangle = \sqrt{1 - x^2} \cdot y - \sqrt{1 - y^2} \cdot x$ two types of Gramian Angular Fields (GAFs) are actually quasi-Gramian matrices $\langle \langle x, y \rangle \rangle$:

$$G = \begin{bmatrix}
\langle \tilde{x}_1, \tilde{x}_1 \rangle & \ldots & \langle \tilde{x}_1, \tilde{x}_n \rangle \\
\vdots & \ddots & \vdots \\
\langle \tilde{x}_n, \tilde{x}_1 \rangle & \ldots & \langle \tilde{x}_n, \tilde{x}_n \rangle
\end{bmatrix}$$ (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 $G_{i-i}=k$ 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, at the bijection matrix is formed. Therefore, the inverse function of the original data is an absolute reconstruction. As shown in Fig. 4.

3.1.2 Markov Transition Field

We get inspiration from Campanharo et al. [7]. 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 $\sum_j \omega_{i,j} = 1$ 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:

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

(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.

### 3.1.3 Recurrence Plot

In this part, recurrence plots (RP) is applied to transform time series into images. The recurrence plots provide a method for visualizing the periodicity of trajectories through phase space [10], 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:

$$R(i, j) = \Theta(\epsilon | |x_i - x_j||)$$

(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. $\epsilon$ is a predefined threshold, and $\Theta(\cdot)$ is the
Fig. 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. Eq. (15).

Fig. 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. Eq. (17).

Heaviside function. In short, a black spot will appear when the distance from $x_i$ and $x_j$ are smaller than $\epsilon$. The following modified RP is used to balance binary output with thresholdless RP [40].

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

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

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
A Time Series Forecasting Model...

Fig. 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

(CNN) [44] structures use some form of data expansion. For example, Alexnet [23] is one of the first deep CNN that created a record benchmark on Imagenet large-scale visual recognition challenge (ILSVRC) datasets [32], uses clipping, mirroring, and color augmentation to optimize the network. Other examples include the original proposal for the VGG network [36], which uses scaling and clipping, the Resnet work [17] using scaling, clipping, and color augmentation, Densenet [19] using translation and mirroring, and perception network using clipping and mirroring. The DA-TSI algorithm proposed in this paper preprocesses the time series (see Sect. 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 [39], natural language processing [9] and reinforcement learning [35]. For time series forecasting, CNNs can reflect the subtle differences of underlying datasets and customize the corresponding architecture [4] and complex data representation [5] 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. [17] 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 Fig. 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 \( f(x) \)

\[
F(x) = H(x) - x
\]  

(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.

### 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 \( 3 \times 3 \) 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 Fig. 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 (\( 3 \times 3 \) 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 [20].

### 3.3.3 Brief Introduction of Densenet

Densenet [45] 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 \( x_{l-1} \)

\[
x_l = H_l(x_{l-1})
\]  

(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:

\[ x_l = H_l(x_0, x_1, \ldots, x_{l-1}) \]  

(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.

### 4 Experiment and Result Analysis

In this section, the experimental effect of data enhanced time series image (da-tsi) algorithm is applied to a small number of data sets, and theoretical analysis is carried out in combination with the experimental results. Finally, the da-tsi algorithm is compared with the time series image (TSI) baseline algorithm, RNN and BPNN.

#### 4.1 Baseline Algorithm

In this paper, tsi-cnn algorithm without data enhancement and RNN and BPNN algorithm before and after data augmentation are used as the benchmark model. In this paper, controlling variables is to obtain scientific experimental results. Three different network structures, resnet-18, vgg-11 and denset-121, are used as the general training model of the deep learning method and applied to the reference depth CNN time series image method.

#### 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.

#### 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

\[
\text{accuracy} = \frac{(TP + TN)}{All}
\]  

(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.

\[
\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100
\]  

(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 \(359 \times 359\), and after resizing the size of combined image (GADF-GASF-MTF-RP) is also \(359 \times 359\). 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 Tables 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 (Table 5).
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** |

Bold values indicate the best experimental result in each experiment.
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** |

Bold values indicate the best experimental result in each experiment
| 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** |

Bold values indicate the best experimental result in each experiment.
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   |

Bold values indicate the best experimental result in each experiment.

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 Tables 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 (Table 7).

4. From Tables 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. From the perspective of forecasting 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. 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. 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 of 3.6% is reflected in the Densenet network. On the one hand, the Densenet network is more
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    |

Bold values indicate the best experimental result in each experiment.
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** |

Bold values indicate the best experimental result in each experiment
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** |

*Bold values indicate the best experimental result in each experiment*
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 Fig. 10 (Table 9).

7. In this paper, RNN and BPNN are added to the comparative experiment. From Tables 10 and 11, it can be seen that RNN has a deeper and more complex network structure than BPNN, and can better extract time series features for classification. Therefore, RNN has the best classification effect and forecasting error before and after data enhancement. Through comparison with TSI-CNN, it is found that although the classification rate of RNN is slightly higher than that of MTF DesNet, the main reason may be overfitting caused by insufficient data volume. After data augmentation, DA-TSI-CNN is superior to DA-RNN in both classification rate and forecasting error MAPE, thus more comprehensively proving the superiority of this algorithm.

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.
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   |
Table 10 The classification rate of baseline models

| Classification rates | h=1  | h=3  | h=6  |
|----------------------|------|------|------|
| RNN                  | 0.387| 0.421| 0.481|
| DA-RNN               | 0.396| 0.440| 0.502|
| BP                   | 0.370| 0.418| 0.464|
| DA-BP                | 0.388| 0.432| 0.482|

Table 11 The MAPE of baseline models

| Classification rates | h=1  | h=3  | h=6  |
|----------------------|------|------|------|
| RNN                  | 12.08| 14.87| 16.96|
| DA-RNN               | 11.50| 14.46| 16.78|
| BP                   | 12.81| 15.53| 17.45|
| DA-BP                | 12.04| 15.17| 17.06|

Author Contributions WJ Software, Data curation, Writing, Methodology. DZ Writing-review, Supervision. LL Conceptualization, Methodology, Writing - review, Supervision. RL Software, Data curation. LZ Data curation.

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

Declarations

Conflict of interest All Authors declar that they have no conflict of interest.

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

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

References

1. Adhikari R, Agrawal RK (2014) A combination of artificial neural network and random walk models for financial time series forecasting. Neural Comput Appl 24:1441–1449
2. Assimakopoulos V, Nikolopoulos K (2000) The theta model: a decomposition approach to forecasting. Int J Forecast 16:521–530
3. Aut NK, Aut FP (2018) Mapa: multiple aggregation prediction algorithm
4. Baxter J (2000) A model of inductive bias learning. J Artif Intell Res 12:149–98
5. Bengio Y, Courville A, Vincent P (2012) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–828
6. Brown RG (1977) Forecasting: issues and challenges for marketing management. J Mark 41:24–38
7. Campanharo A, Sirer MI, Malmgren RD, Ramos FM, Amaral L, Perc M (2011) Duality between time series and networks. PLoS One 6:e23378
8. Connor JT, Martin RD, Atlas LE (2002) Recurrent neural networks and robust time series prediction. IEEE Trans Neural Netw 5:240–254
9. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding
10. Eckmann JP, Kamphorst SO, Ruelle D (1987) Recurrence plots of dynamical systems. Europhys Lett (epl) 4:973–977
11. Petropoulos F, Makridakis AVS (2014) Horses for courses’ in demand forecasting. Eur J Oper Res 237:152–163
12. Gers FAJ, Eck D (2001) Applying lstm to time series predictable through time-window approaches. In: International conference on artificial neural networks
13. Fiorucci L, F. Y. B. F. M. J. A., J. A (2016) Forechtera r package manual
14. Guennec AL., Malinowski S, Tavenard R (2016) Data augmentation for time series classification using convolutional neural networks
15. Hasibi R, Shokri M, Dehghan M (2019) Augmentation scheme for dealing with imbalanced network traffic classification using deep learning
16. Hatami N, Gavet Y, Debayle J (2019) Bag of recurrence patterns representation for time-series classification. Pattern Anal Appl 22:877–887
17. 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)
18. Heaton Y, Jeff IG, Bengio Y, Courville A (2017) Deep learning, genetic programming and evolvable machines. Nature 19:1–137
19. Huang G, Liu Z, Laurens V, Weinberger KQ (2016) Densely connected convolutional networks. In: IEEE computer society
20. Jaworek-Korjakowska JGM, 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)
21. Kang Y, Hyndman RJ, Smith-Miles K (2017) Visualising forecasting algorithm performance using time series instance spaces. Int J Forecast 33:345–358
22. Kim M, Chi YJ (2020) Label-preserving data augmentation for mobile sensor data. Multidimens Syst Signal Process 32(1):115–29
23. Krizhevsky A (2012) Learning multiple layers of features from tiny images
24. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
25. Lemke C, Gabrys B (2010) Meta-learning for time series forecasting and forecast combination. Neurocomputing 73:2006–2016
26. Macready No free lunch theorems for optimization. IEEE Trans Evolut Comput 1(1):67-82
27. Spyros M, Hibon M (2001) Response to the commentaries on ‘the m3-competition: results, conclusions and implications’. Int J Forecast 17:581–584
28. Morwitz VG, Steckel JH, Gupta A (2007) When do purchase intentions predict sales? Int J Forecast 23:347–364
29. Naduvilvadukootu S, Aniry RA, Riley P (2017) Evaluating preprocessing strategies for time series prediction using deep learning architectures
30. Prudencio R, Ludermir TB (2004) Meta-learning approaches to selecting time series models. Neurocomputing 61:121–137
31. Rice J (1976) The algorithm selection problem4
32. 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. Int J Comput Vision 115:211–252
33. Shah C (1997) Model selection in univariate time series forecasting using discriminant analysis. Int J Forecast 13:489–500
34. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:1–48
35. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S (2016) Mastering the game of go with deep neural networks and tree search. Nature 529:484–503
36. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Sci
37. Smith-Miles (2009) Cross-disciplinary perspectives on meta-learning for algorithm selection. Cross-disciplinary perspectives on meta-learning for algorithm selection
38. Talagala TS, Hyndman RJ, Athanasopoulos G (2018) Meta-learning how to forecast time series. Monash Econometrics and Business Statistics Working Papers
39. Technicolor T, Related S, Technicolor T, Related S (2012) Imagenet classification with deep convolutional neural networks, 1097–1105
40. Thiel M, Romano MC, Kurths J (2004) How much information is contained in a recurrence plot? Phys Lett A 330:343–349
41. Wang Z, Oates T (2015) Imaging time-series to improve classification and imputation. AAAI Press, USA, pp 3939–3945
42. 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
43. Winters PR (1976) Forecasting sales by exponentially weighted moving averages. Manag Ence 6:324–342
44. Lecun Y, YB, Bottou L, Haffner P (2013) Gradient-based learning applied to document recognition. pp 2278–2324
45. Zhu Y, Newsam S (2017) Densenet for dense flow. In: 2017 IEEE international conference on image processing (ICIP)

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.