entropy

MDPI

Article

BBS Posts Time Series Analysis based on Sample Entropy and Deep Neural Networks

Jindong Chen [1,2], Yuxuan Du [1,2], Linlin Liu [1,2], Pinyi Zhang [1,2] and Wen Zhang [3,4,*]

1 School of Economics and Management, Beijing Information Science & Technology University, Beijing 100192, China; j.chen@amss.ac.cn (J.C.); rainia.dyx@gmail.com (Y.D.); lll291165517@gmail.com (L.L.); zhangpinyi@bjut.edu.cn (P.Z.)
2 Beijing Key Lab of Green Development Decision Based on Big Data, Beijing 100192, China
3 School of Economics and Management, Beijing University of Technology, Beijing 100124, China
4 School of Information Engineering, Xi'an University, Xi'an 710065, China
* Correspondence: zhangwen@mail.buct.edu.cn; Tel.: +86-10-6419-7040

Received: 10 December 2018; Accepted: 8 January 2019; Published: 12 January 2019

Abstract: The modeling and forecasting of BBS (Bulletin Board System) posts time series is crucial for government agencies, corporations and website operators to monitor public opinion. Accurate prediction of the number of BBS posts will assist government agencies or corporations in making timely decisions and estimating the future number of BBS posts will help website operators to allocate resources to deal with the possible hot events pressure. By combining sample entropy (SampEn) and deep neural networks (DNN), an approach (SampEn-DNN) is proposed for BBS posts time series modeling and forecasting. The main idea of SampEn-DNN is to utilize SampEn to decide the input vectors of DNN with smallest complexity, and DNN to enhance the prediction performance of time series. Selecting Tianya Zatan new posts as the data source, the performances of SampEn-DNN were compared with auto-regressive integrated moving average (ARIMA), seasonal ARIMA, polynomial regression, neural networks, etc. approaches for prediction of the daily number of new posts. From the experimental results, it can be found that the proposed approach SampEn-DNN outperforms the state-of-the-art approaches for BBS posts time series modeling and forecasting.

Keywords: sample entropy; deep neural networks; BBS posts; time series

1. Introduction

The Internet has become a major public opinion formation and diffusion platform [1]. In the Web 2.0 era, Bulletin Board System (BBS), Micro-blog, WeChat, etc. are main sources for public information dissemination, and become the core areas of public opinion monitoring [2]. The numbers of posts, blogs or micro-blogs represent the hot-degree and trend of public opinion, so time series analysis of the numbers of posts, blogs or micro-blogs, such as prediction of the number of new posts, blogs or micro-blogs in a time interval ahead, can be considered as an important signal for government agencies, enterprises and website operators to make decisions. Based on time series analysis of the numbers of posts, blogs or micro-blogs, convincing results are obtained for the election result prediction [3], crisis management [4] and stock market forecasting [5]. Therefore, time series analysis of the numbers of posts, blogs or micro-blogs enable them to monitor the tendency of public opinion, and further support them to make rational planning and actions for public opinion management and guidance [6].

For government agencies and enterprises, the prediction of the numbers of posts, blogs or micro-blogs can help them make decision for at least three reasons [7,8]. First, the prediction of the numbers of posts, blogs or micro-blogs provides government agencies and enterprises with a measure of the trend, scope and duration time of public opinion on their related topics. Second, it is useful to

estimate the effort involved in public opinion management and guidance. To different trend types of events or topics, different operation methods and effort investments are required. Generally, the effort involved in public opinion management and guidance is proportional to the number of posts, blogs or micro-blogs. Third, based on the numbers of posts, blogs or micro-blogs, the effectiveness of guidance strategies can be evaluated. According to the positive or negative feedback from public opinion, the guidance strategies can be improved immediately. For website operators, the prediction of the numbers of posts, blogs or micro-blogs can help them allocate resource or strategies on hot topics [9]. Without sufficient resource allocation for hot events, it will make their system delay or crash. Conversely, too many resources allocation will increase their operational cost.

Hence, for effective management and guidance of public opinion, the prediction of the numbers of posts, blogs or micro-blogs is a critical issue. Various approaches are proposed to solve this problem and we can divide the approaches into two kinds: diffusion model and time series model. For diffusion model, the classic mathematical models of diffusion are adopted to establish public opinion, such as Logistic distribution [10], epidemic model [11] and Michaelis–Menten model [12]. The information diffusion process of public opinion is modeled through the classic diffusion model. Based on the identified model, the trend, peak and duration at different stages of public opinion are predicted. For time series model, ignoring the diffusion mechanism of public opinion, the diffusion model of public opinion is constructed only based on time series data. Auto-regressive integrated moving average (ARIMA) [13], artificial neural network [14] and support vector machines [15,16] are the frequently used models for time series data. Due to effectiveness of time series model for public opinion prediction, the time series model is applied for the prediction of the numbers of posts, blogs or micro-blogs.

In Chinese online community, BBS serves as an important social media; the extensive interests and contents are distinct characters of the Chinese BBS sites [17]. As an emerging media and electronic information center, the Chinese BBS sites fulfill the requirements of netizens to be informed and exchange opinions [18]. With the appearance of blogs, twitter, WeChat, etc., the influence of BBS is declining, but, due to the Internet regulations in China, the Chinese BBS sites can offer another channel to express information and spread opinions, and are still an important platform in China. Tianya Club (http://bbs.tianya.cn/), one of the most influential Chinese BBS sites, provides BBS, blogs, etc. services for netizen and consists of many sub-boards for different content/topic discussions, such as Tianya Zatan and Baixing Shengyin [19]. Tianya Zatan board is an important board within Tianya Club, the content of which covers the daily news of current society and personal life. Daily new posts published in Tianya Zatan board are nearly 1000, and millions of clicks and replies are created by netizens [20]. The daily new posts data of Tianya Zatan were selected as the data source for public opinion monitoring.

Takens' [21] embedding theorem and Sauer et al.'s [22] embedology theorem provide a theoretical foundation for nonlinear dynamical system reconstruction based on its generated time series sequence. Consequently, for the modeling and forecasting of BBS posts time series, two main problems are: (1) Determination of the embedding dimension. A time series can be represented in the so-called "phase space" by a set of delay vectors (DVs), and the embedding dimension defines the size of the DVs. (2) Which model is fit for BBS post number prediction. Approximate entropy is an effective model for the embedding dimension analysis of time series; sample entropy (SampEn) is a modification of approximate entropy [23,24]. With multiple layers and more neurons, deep neural networks (DNN) can detect the features of data, and are more effective for time series prediction [25].

Therefore, by combining SampEn and DNN, an approach SampEn-DNN is proposed to predict BBS new post number time series. SampEn-DNN applies sample entropy to measure the predictability of DVs with different dimensions, selects the dimension of DVs with smallest complexity, and feeds the DVs as the input of DNN to improve the predictive performance of time series. However, in some cases, the single-scale sample entropy cannot really reflect the complexity of a time series, thus, to avoid this issue, multi-scale sample entropy is adopted in this paper. The skipping parameter and the dimension

of DVs are tuned by multi-scale sample entropy. The predictive performance of SampEn-DNN for Tianya Zatan new posts time series was investigated. For the combination of sample entropy and DNN for time series modeling and forecasting, both the proposed method and the application area are attempted for the first time. To illustrate the improvements of SampEn-DNN, the performances of ARIMA, seasonal ARIMA, polynomial regression and artificial neural network (ANN) on Tianya Zatan new posts time series analysis were compared.

The rest of the paper is organized as follows. Section 2 presents the related methods for time series analysis. The proposed approach SampEn-DNN is shown in Section 3. Section 4 presents the experimental results of SampEn-DNN and the state-of-the-art approaches for time series analysis of Tianya Zatan new posts. Finally, concluding remarks and future work are given in Section 5.

2. Methodology

Time series $X_t(t = 1, 2, 3, \ldots, N)$ is an ordering set of observations of a variable over successive periods of time. Time series modeling and forecasting has fundamental importance to various practical domains [26]. Stock exchange data, wind speed, global temperature, etc. are typical examples of time series. The natural temporal ordering feature of time series makes time series analysis different from other data analysis problems, in which there is no natural ordering of the observations. The studies of time series data can be divided into two parts: one is to extract and understand the meaningful statistics and other characteristics of the data, and the other is to predict future values based on previously observed values.

Parametric approaches are frequently used for time series modeling and forecasting [26]. The state-of-the-art parametric approaches include ARIMA model, seasonal ARIMA model, polynomial regression and ANN. The parametric approaches of time series analysis assume that underlying process is stationary. Generally, a time series $X_t(t = 1, 2, 3, \ldots, N)$ is stationary if $E(X_t^2) < \infty$, $E(X_t) = \alpha$, $\forall t \in T = \{1, 2, \ldots, N\}$ and $E(X_{t+r} - \alpha)(X_t - \alpha) = \gamma_X(r)$, $\forall t, t + r \in T = \{1, 2, \ldots, N\}$, and $\gamma_X(r)$ is the auto-covariance function. In other words, for a stationary time series, the variation is finite, the expected values at any time points equals the same value α, and the auto-covariance is merely dependent on their time lag r and not dependent on time t or $t + r$. For example, the simplest stationary time series is white noise. For time series modeling and forecasting, the first issue is to know whether a time series is non-stationary or stationary. The common method for stationary test is augmented Dicker–Fuller (ADF) test [27]. The related approaches for time series modeling and forecasting are presented as follows.

2.1. ARIMA Model

ARIMA model is a general class of ARMA model with differencing manipulation on time series data, and ARMA model consists of two parts: autoregressive (AR) model and moving average (MA) model. These models are applied for the fitting of time series data, and aim to describe the autocorrelations in time series.

For a time series $X_t(t = 1, 2, 3, \ldots, N)$, assume the number of autoregressive terms as p, AR model can be abbreviated as AR(p), and expressed as

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \omega_t \tag{1}$$

where x_t is stationary, $\phi_1, \phi_2, \ldots, \phi_p$ are constants and $\phi_p \neq 0$. ω_t is assumed to be Gaussian white noise with variance σ_ω^2 and zero mean.

Assume the moving average order as q, so MA model can be abbreviated as MA(q) and expressed as

$$x_t = \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \cdots + \theta_q \omega_{t-q} \tag{2}$$

where model parameters are $\theta_1, \theta_2, \ldots, \theta_q (\theta_q \neq 0)$, and q lags are in the moving average.

According to Equations (1) and (2), ARMA model with the autoregressive and the moving average order p and q can be abbreviated as ARMA(p, q) and expressed as

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \cdots + \theta_q \omega_{t-q} \quad (3)$$

If the mean μ of x_t is non-zero, then set $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$, and the ARMA model can be rewritten as

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \cdots + \theta_q \omega_{t-q} \quad (4)$$

ARIMA model contains differencing manipulation, which is used to transfer a non-stationary time series to a stationary time series. If L is a differencing operator, $W_t = \nabla^d x_t = (1 - L)^d x_t$ conforms to the process ARMA(p, q). ARIMA model can be denoted as ARIMA(p, d, q). The general expression of ARIMA(p, d, q) is given as Equation (5).

$$W_t = \alpha + \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \theta_2 \omega_{t-2} + \cdots + \theta_q \omega_{t-q} \quad (5)$$

where, through difference by order d, the original time series $X_t (t = 1, 2, 3, \ldots, N)$ is converted from non-stationary into a stationary time series $\boldsymbol{W}_t$.

2.2. Seasonal ARIMA Model

For real issues, most time series show seasonal variation. Seasonal time series mean that there is a similar trend of the observations during the same period (e.g., daily, monthly or yearly) of the time series. Additionally, the observations during the successive periods may also exhibit another seasonal trend.

To address the seasonality and potential seasonal unit root, an extensional ARIMA model called Seasonal ARIMA model is proposed [27]. Assume the periodicity of time series is s, Seasonal ARIMA model is given by

$$(1 - L^s)^{D_s} W_t = \alpha + \phi_1 (1 - L^s)^{D_s} W_{t-1} + \cdots + \phi_p (1 - L^s)^{D_s} W_{t-p} + \omega_t + \theta_1 \omega_{t-1} + \cdots + \theta_q \omega_{t-q} \quad (6)$$

where $(1 - L^s)^{D_s}$ is the seasonal differencing operator, and accounts for non-stationarity in observations made in the same period in successive period, $D_s = 0$ or 1 for $s = 0$ or > 1.

2.3. Polynomial Regression

For a given dataset $(x_i, y_i),\ i = 1, 2, \cdots, N$, x is the independent variable and y is the dependent variable. Polynomial regression is a form of linear regression to model the relationship between x and y as an nth order polynomial. In general, a polynomial regression fits data to a model of the following form,

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n + \varepsilon_i \quad (7)$$

Parameters $a_1, a_2, \cdots, a_n$ of polynomial regression are identified by the method of least squares. According to Taylor's theorem [28], a polynomial regression is the expansion of Taylor series, so it can be used to approximate continuous functions for curve fitting and trend analysis.

2.4. Artificial Neural Networks

ANN is a framework for machine learning inspired by biological neural networks. One of the most widely applied models is back propagation neural network (BPNN) [29]. BPNN is a kind of feed-forward network, the connection weights of which are trained by error back propagation algorithm. For a given dataset $(x_i, y_i),\ i = 1, 2, \cdots, N$, the training of BPNN includes two parts: one is forward propagation, and the other is back propagation. Forward propagation: The input sample x_i

is propagated from the input layer, via the hidden layer, to the output layer. The connection weights of BPNN in forward propagation process are maintained constant. Back propagation: The difference (error) between the real value y_i and expected output $\hat{y}_i$ of BPNN is propagated from the output layer to the input layer. The connection weights of BPNN are updated by the error feedback during the process. The objective of BPNN training is to find a set of network weights that minimize the difference between the real value y_i and the expect output $\hat{y}_i$.

3. SampEn-DNN

By combining of sample entropy (SampEn) and deep neural networks (DNN), a novel time series modeling and forecasting method SampEn-DNN is proposed to predict the daily number of BBS new posts.

3.1. Sample Entropy

For a time series $X_t(t = 1, 2, 3, \ldots, N)$, assume its constant time interval as τ. The constant time interval of BBS new posts time series in this study is one day. The SampEn of time series X_t can be computed as follows.

First, define the dimension of embedding vector as m and tolerance as r, such that embedding vector is given as $X_m(i) = \{x_i, x_{i+1}, \cdots, x_{i+m-1}\}$.

Second, the Chebyshev distance $d[X_m(i), X_m(j)](i \neq j)$ is used as the distance function [30],

$$d[X_m(i), X_m(j)] = \max_{k=0,\cdots,m-1}(|x(i+k) - x(j+k)|) \tag{8}$$

Third, the number of $X_m(j)$ $(1 \leq j \leq N-m, j \neq i)$ that do not exceed the tolerance r $(d[X_m(i), X_m(j)] < r)$ is counted and denoted as $n_i(m)$, and then the proportion $c_i(m) = \frac{n_i(m)}{N-m}$ that any $X_m(j)$ is close to $X_m(i)$ is computed.

Fourth, by averaging over all possible $X_m(i)$, the proportion $c(m) = \frac{1}{N-m+1}\sum_{i=1}^{N-m+1} c_i(m)$ is estimated.

Fifth, the Chebyshev distance and $c(m+1)$ for embedding vector dimension as m + 1 are computed in a similar way.

Sixth, SampEn of $X_t(t = 1, 2, 3, \ldots, N)$ is defined as

$$SampEn(m, r) = -\log\frac{c(m+1)}{c(m)} \tag{9}$$

From the definition, it can be found that $c(m + 1)$ is not bigger than $c(m)$, so the $SampEn(m, r)$ value will be either zero or positive. For time series dataset, a smaller value of $SampEn(m, r)$ means more self-similarity (predictability) or less noise. To overcome the shortcuts of single-scale SampEn in some special case, multi-scale SampEn is adopted. In multi-scale SampEn, a certain interval between its every element is defined for input vector specified by the skipping parameter δ. Hence, input vector is modified as $X_{m,\delta}(i) = \left\{x_i, x_{i+\delta}, \cdots, x_{i+(m-1)\delta}\right\}$, $c(m)_\delta$ is expressed as $c(m)_\delta = d[X_{m,\delta}(i), X_{m,\delta}(j)] < r$, and then SampEn can be given as $SampEn(m, r, \delta) = -log(c(m+1)_\delta / c(m_\delta))$. In this study, the value of tolerance r is set as $0.02std$, where the notation std stands for the standard deviation of time series $X_t(t = 1, 2, 3, \ldots, N)$ [24].

The procedures for the calculation of $SampEn(m, r, \delta)$ is presented in Figure 1. For the given starting positions as i and j in the time series $X_t(t = 1, 2, 3, \ldots, N)$, Lines 4–9 in Figure 1 are applied to decide $d[X_{m,\delta}(i), X_{m,\delta}(j)] < r$ or not, and Lines 11–13 are implemented to decide $d[X_{m+1,\delta}(i), X_{m+1,\delta}(j)] < r$ or not. In Figure 1, for given i and j, according to the definition of Chebyshev distance, if $d[X_{m,\delta}(i), X_{m,\delta}(j)] < r$ and $\text{Max}(\text{Abs}(X(i+k*\delta) - X(j+k*\delta))) \leq r$, $d[X_{m+1,\delta}(i), X_{m+1,\delta}(j)] < r$ can be achieved.

Input:
X_t -a time series with N data points;
m-dimension of input vector;
r - the tolerance of variation of the time series;
δ-the skipping parameter;
Output:
SampEn-the sample entropy of the time series;
Procedure:

```
For i=0;i<N-(m+1)*δ; i++
  For j=i+1;j<N-(m+1)*δ;j++
      flag=true;
      For k =0;k <m; k++
          If Max(Abs (X_m(i+k*δ)-X_m(j+k*δ)))>r
              flag=false;
              Break;
           End if
      End for
      If(flag)cm++; End if
      k=m;
      If(flag&& Max(Abs(X_{m+1}(i+k*δ)-X_{m+1}(j+k*δ)))≤r)
          cml++;
       End if
  End for
End for
If cm>0&&cml>0 SampEn=Log（cm/cml）;
```

Figure 1. Procedures for the calculation of *SampEn*(m, r, δ).

3.2. Deep Neural Network

DNN model consists of deep belief network (DBN) [31] and feedforward neural network (FNN). DBN is developed by stacking of multiple-units of Restricted Boltzmann Machines (RBM) [32]. The structure of DNN is shown in Figure 2. The aim of DBN is to extract the high-level features from input data by the stacked RBMs. The learning process of the stacked RBMs is that the features produced by the hidden layer of one RBM serve as the input to the higher-level RBM. The high-level feature representation learned by DBN is fed as the input of FNN. Meanwhile, BPNN is one of the most used FNN models, and adopted for DNN model.

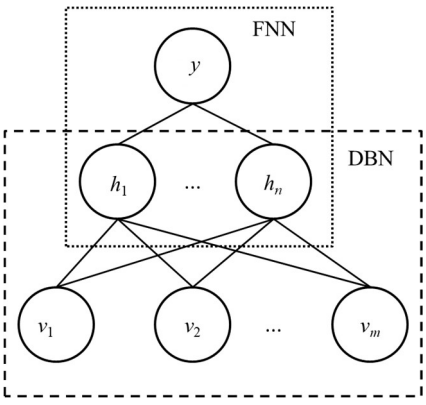

Figure 2. DNN (deep neural networks) model consists of DBN (deep belief network) and FNN (feedforward neural network).

Hence, RBM is the main component of DNN. RBM is an energy-based deep learning model for unsupervised learning, and consists of two kinds of layers: one is the visible layer and the other is the hidden layer. The visible layer is for input data representation, and the hidden layer is to represent a

probability of the distribution of input data. The neurons in the visible layer are only connected to the neurons in the hidden layer.

For RBM, assume the numbers of neurons in the visible layer and the hidden layer as m and n, denoted as $\mathbf{v}= (v_1, \ldots, v_m)$ and $\mathbf{h} = (h_1, \ldots, h_n)$, respectively. Meanwhile, assume the *bias* vectors and the weight matrix of RBM as **a**, **b** and **W**. The energy-based model means an entropy function is applied to define the log-likelihood input data distribution over the parameters **a**, **b**, **W**, **v** and **h**. The energy function for RBM is given by Equations (10) and (11).

$$\mathbf{E}(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T\mathbf{W}\mathbf{v} - \mathbf{a}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} \tag{10}$$

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i,j=1}^{m,n} v_i h_j w_{i,j} - \sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j \tag{11}$$

For each pair of neurons in the visible layer and the hidden layer, the joint probabilistic distribution is defined as

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{v,h} e^{-E(v,h)}} \tag{12}$$

The sum of all probabilities of the hidden vector is the probability that the network assigns to the visible vector, and expressed as

$$p(\mathbf{v}) = \frac{\sum_h e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \tag{13}$$

Since the neurons in visible layer only connect to the neurons in hidden layer, there is no connection between neurons in the same layer. The joint probability of each pair of neurons in different layers can be facilitated by the conditional probabilities,

$$p(\mathbf{h}\,|\mathbf{v}\,) = \prod_j p(h_j|v\,) \tag{14}$$

$$p(\mathbf{v}\,|\mathbf{h}\,) = \prod_i p(v_i|h\,) \tag{15}$$

For binary data, Equations (14) and (15) can be expressed as

$$p(h_j = 1\,|\mathbf{v}\,) = sigm(b_j + \sum_{i=1}^{m} v_i w_{ij}) \tag{16}$$

$$p(v_i = 1\,|\mathbf{h}\,) = sigm(a_i + \sum_{i=1}^{n} h_i w_{ij}) \tag{17}$$

where $sigm(x)$ is the sigmoid function.

3.3. SampEn-DNN Approach

DNN as regression model for time series analysis. The primary problem is to decide the formation of its input vectors, which is the main factor to affect the predictive performance for model. Generally, the input vectors are decided by two parameters: the dimension of input vector m and the skipping parameter δ. Based on SampEn method, the dimension of input vector m and the skipping parameter δ are optimized. For DNN model training, the first $m - 1$ elements are applied to predict the last (m-th) element.

To optimize the parameters m and δ of input vector, the maximum values of the two parameters need to be decided first. For the dimension of input vector, if m is larger than 13, the SampEn of time series cannot be derived in most cases. Because, at this stage, $c(m)$ and $c(m + 1)$ in Equation (9) are zeros, the differences between the elements of m-length and the $(m + 1)$-length input vectors are all bigger than $0.02std$. For skipping parameter, if the value of δ is too big, the intervals between data

points will be large, which means the similarity of the data points to each other of time series will be smaller, and the unpredictability of the input vectors will be larger. Based on this point, the biggest skipping parameter δ is constrained as 12 in the study [16].

The SampEn results of BBS new posts time series based on different dimensions of input vectors m and different skipping parameters δ are presented in Table 1.

Table 1. SampEn (sample entropy) results of BBS (Bulletin Board System) posts time series with different m and δ.

m \ δ	1	2	3	4	5	6	7	8	9	10	11	12
2	1.19	1.22	1.20	1.24	1.31	1.32	1.05	1.32	1.36	1.26	1.30	1.38
3	1.06	1.06	1.13	1.14	1.09	1.18	0.93	1.20	1.14	1.12	1.18	1.20
4	0.96	0.95	1.01	1.05	0.96	1.09	0.85	1.10	1.06	1.04	1.06	1.12
5	0.90	0.80	0.91	0.97	0.92	0.95	0.81	1.07	1.00	0.93	0.97	1.15
6	0.83	0.75	0.87	0.96	0.82	0.93	0.78	1.00	0.89	0.99	0.94	1.09
7	0.81	0.72	0.80	0.90	0.73	0.93	0.76	0.87	0.83	1.13	0.83	0.92
8	0.68	0.72	0.90	0.73	0.70	0.85	0.71	0.88	0.97	1.10	0.76	0.78
9	0.73	0.72	0.92	0.78	0.61	1.30	0.76	1.14	0.98	0.96	1.10	0.89
10	0.63	0.74	0.69	0.98	0.50	1.50	0.74	0.98	1.10	0.92	1.20	1.95
11	0.54	0.85	0.59	1.10	0.41	NaN	0.45	NaN	1.11	NaN	NaN	NaN
12	0.56	1.50	0.92	NaN	0.47	NaN	0.48	NaN	NaN	NaN	NaN	NaN
13	0.98	NaN	NaN	NaN	0.69	NaN	0.62	NaN	NaN	NaN	NaN	NaN

As shown in Table 1, when keeping the skipping parameter δ constant, for the dimension of input vectors m varying from 2 to 13, the SampEn results of BBS post time series decrease at first and increase after a critical threshold. For example, when $\delta = 1$, for m varying from 2 to 11, the SampEn results of the time series decrease from 1.19 to 0.54; however, for m varying from 12 to 13, the SampEn results increase to 0.54. Formally, this phenomenon is known as phase transition. In Table 1, it can also be found that, when m is increasing, the SampEn results will decreases along with the range of $c(m)$ and $c(m+1)$. Similarly, the value of $(c(m)-q)/(c(m+1)-q)$ for a given q is smaller than $c(m)/c(m+1)$ when $c(m)$ is larger than $c(m+1)$. When phase transition appears, the complexity of input vector will increase, which means the unpredictability of time series increase.

Therefore, according to the above analysis, the procedures for the determination of the optimal size of input vector m and the optimal skipping parameter δ are shown in Figures 3 and 4. At first, for the optimal size of input vectors m determination, the average of the SampEn results of different skipping parameter δ under same m is adopted, and m with the smallest average SampEn is selected as the optimal parameter. After that, the skipping parameter δ with the minimum SampEn is selected as the optimal skipping parameter.

The procedure for determination of the optimal size of input vector m^* is presented in Figure 3. *seMatrix* is derived based on the SampEn results of different m and δ in Table 1. According to Lines 1–8, the SampEn results of all δ under the same size m are summarized. Line 9 is to get the average of SampEn results under the same size parameter m. Based on the SampEn values in Table 1, the optimal dimension of input vector is obtained as 11. It can be found that, for different dimension of input vector m, the possible skipping parameters δ with SampEn results not equal to infinity are different from each other.

```
Input:
seMatrix -a matrix recording SampEn under different settings of m and δ;
Output:
m* -the optimal dimension of input vectors;
Procedure:
1. For each m in matrix seMatrix
2.     count=0; sum=0; average[m]=0;
3.     For each  δ in matrix seMatrix
4.         If seMatrix[m][δ]≠-1.0
5.             count++;
6.             sum=sum+seMatrix[m][δ];
7.         End if
8.     End for
9.     average[m]=sum/count;
10.End for
11.m*=0;
12.For each m in array average
13.If average[m*]>average[m]
14.    m*=m;
15.End if
```

Figure 3. Procedure of determination of the optimal dimension m^*.

```
Input:
seMatrix [m*] -an array recording SampEn results under the parameter m*;
Output:
δ* - the optimal skipping parameters;
Procedure:
   δ*=0
1. For each m in array seMatrix[m*]
2.     If seMatrix[m*][δ*]>seMatrix[m*][δ]
3.             δ *= δ
4.     End if
```

Figure 4. Procedure for determination of the optimal skipping parameter δ^*.

Therefore, the procedure for determination of the optimal skipping parameter δ is shown in Figure 4. According to Lines 2–3, when increasing the skipping parameter δ under the dimension of input vector m^*, the optimal skipping parameter δ^* is found. Based on the SampEn values in Table 1, the optimal skipping parameter δ is derived as 5.

4. Experiments and Discussions

4.1. Datasets

To compare the effectiveness of SampEn-DNN with the state-of-the-art approaches for time series modeling and forecasting, the time series of daily new post number of Tianya Zantan board (website: http://bbs.tianya.cn/list.jsp?item=free&order=1) was selected as the data source. The daily new posts on Tianya Zatan broad are published by netizens to discuss the hot and sensitive topics of current society. With a spider system developed by our group, the new posts published on Tianya Zatan board were downloaded and parsed, and the numbers of daily new posts were counted. For this study, the numbers of daily new posts from 1 January 2013 to 31 December 2017 were selected. This dataset contains 1826 data points with 1,782,793 new posts. The whole time series of daily new post number is shown in Figure 5.

In Figure 5, it can be found that the time series of daily new post number published on Tianya Zatan shows a downtrend from 2013 to 2017, which is mainly owing to the popularity of Micro-blog and WeChat. The time series of daily new post number also presents periodic fluctuations, e.g., the number of new posts on the working days is much greater than the number on weekends or holidays.

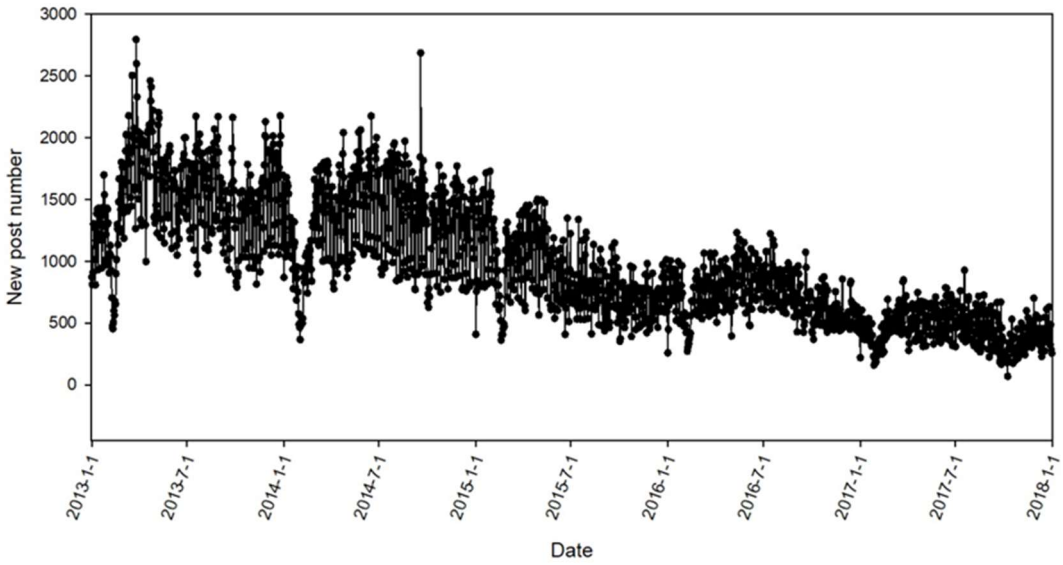

Figure 5. The time series of daily new post number on Tianya Zatan broad.

4.2. Experimental Procedures

For time series modeling and forecasting, the first issue is to verify whether the time series is stationary. Through ADF test on the BBS new posts time series, it can be found that the dataset is stationary ($P < 0.05$). For effectiveness comparison in daily BBS new post number prediction, the whole dataset was split into five subsets with the daily new posts for each year as one subset (2013, 2014, 2015, 2016 and 2017). Through the ADF test on these five subsets, it can be found that the subset of 2014 is non-stationary. The first order difference was conducted on the five subsets, and then the ADF test results show all subsets are stationary ($P < 0.05$).

ARIMA model: The parameters identification of ARIMA model is based on Box and Jenkin's approach [27]. After first order differencing manipulation, the time series became stationary. Hence, only the orders p and q needed to be identified. Traditionally, the charts of autocorrelation faction (ACF) and partial ACF (PACF) [27] are adopted to find several candidate couples p and q. Furthermore, through the Akaiike information criterion, the two parameters p and q of ARIMA model were decided. Seasonal ARIMA model: First, considering the influences of seasonal vacations and holidays, the seasonal factor of the time series S_t was derived, $S_t = 7$. Second, with the seasonal factor S_t, the time series data were filtered through convolution operation. Third, the order of the autoregressive order p and the moving average order q were also decided based on ACF and PACF charts. Polynomial regression: First, BBS new posts time series was converted to a cumulative time series $N_{new} = \{x_1^{new}, x_2^{new}, \cdots, x_n^{new}\}$, where $x_i^{\text{new}} = \sum_{k=0}^{i} x_k$. Second, the least squares method was used for parameters identification; Third, the performances of different orders of polynomial regression ewre compared to select the optimal order. ANN model: According to the skipping parameter and input vectors results of SampEn, the training and test dataset were extracted from the time series; then, the reconstructed time series were fed into ANN model to regulate the weights of networks. DNN model: The procedures were similar with ANN model. Based on the results of sample entropy, the training and test datasets were derived, and then used for DNN modeling.

For each subset, the new post number points in the time series of the first 11 months (January–November) were selected as the training dataset, and the new post number points in the time series of the last month (December) were selected as the test dataset. Based on the reconstructed dataset, the predictive performance of SamEn-DNN for BBS post time series was compared with ARIMA, Seasonal ARIMA, polynomial regression, and ANN models. To evaluate the performance of SamEn-DNN and other mentioned methods, magnitude of relative error (MRE) and mean magnitude

of relative error (MMRE) were selected as the evaluation criterion. MRE and MMRE are expressed in Equations (18) and (19).

$$MRE_i = \frac{|x_i - \hat{x}_i|}{x_i} \tag{18}$$

$$MMRE = \frac{\sum_{i=1}^{n} MRE_i}{n} \tag{19}$$

where x_i is the real post number and $\hat{x}_i$ is the estimated post number of model. The smaller are the MRE and MMRE, the better is the prediction. The range of MRE and MMRE are $[0, 1]$.

4.3. Experimental Results

Based on Equations (18) and (19), the performances of SampEn-DNN and the mentioned methods were evaluated, and the MMRE results are presented in Table 2.

Table 2. MMREs (mean magnitude of relative error) of ARIMA (auto-regressive integrated moving average), seasonal ARIMA, polynomial (polynomial regression), ANN (artificial neural networks) and SampEn-DNN (sample entropy-deep neural networks) on BBS post time series.

Subset	ARIMA	Seasonal ARIMA	Polynomial	ANN	SampEn-DNN
1	0.2355 ± 0.0090	0.1968 ± 0.0119	0.2772 ± 0.0109	0.2003 ± 0.0129	0.1419 ± 0.0078
2	0.1895 ± 0.0103	0.1691 ± 0.0126	0.4735 ± 0.0179	0.1694 ± 0.0107	0.1241 ± 0.0078
3	0.1915 ± 0.0117	0.1704 ± 0.0107	0.3535 ± 0.0135	0.1934 ± 0.0092	0.1748 ± 0.0080
4	0.1325 ± 0.0049	0.1188 ± 0.0066	0.3637 ± 0.0212	0.1450 ± 0.0098	0.0878 ± 0.0036
5	0.1653 ± 0.0083	0.1451 ± 0.0102	0.2401 ± 0.0126	0.1494 ± 0.0101	0.1256 ± 0.0043

Table 2 shows the MMREs of SampEn-DNN and the mentioned methods for BBS daily new post number prediction. In Table 2, it can be found that, among the five models, SampEn-DNN model shows better performances than other methods on four of the five subsets. Meanwhile, Seasonal ARIMA model shows better performance on one of the five subsets. Thus, it may be concluded that the SampEn-DNN outperforms Seasonal ARIMA model. Furthermore, both SampEn-DNN and Seasonal ARIMA model show better performances on the five subsets than ARIMA, polynomial regression and ANN. Moreover, ANN has produced better performance than ARIMA model, and ARIMA model has outperformed polynomial regression. From this point, it can be said that, among the mentioned five methods, polynomial regression has obtained the poorest performance for BBS new posts time series modeling and forecasting.

To directly show the advantage of each method, the pairwise comparisons of the predictive performances for SampEn-DNN and the other mentioned methods were conducted. The results of MREs of each method on the five datasets were used for comparison. As mentioned in Section 4.2, for each subset, the 31 data points of December were selected as the test dataset, thus there wre totally 155 MREs in the five subsets. For pairwise effectiveness comparison, the Wilcoxon signed rank test [33] was employed. The Wilcoxon signed rank test is a non-parametric statistical hypothesis test for the difference assess of the repeated measurements on a single sample. The Wilcoxon signed rank test results of the MREs of the compared methods are shown in Table 3. The codification of the P-value in ranges is defined as follows: "$\sim$" means $P > 0.05$, which indicates that differences in performances of two compared methods are not significant; "$<$" ("$>$") means $0.01 < P \leq 0.05$, which indicates one model slightly outperforms the other model; and "$\gg$" ("$\ll$") means $P \leq 0.01$, which indicates one model significantly outperforms the other model. For example, for effectiveness comparison of ARIMA and SampEn-DNN, the code "$\ll$" means that the P-value of Wilcoxon signed rank test is <0.01, thus SampEn-DNN shows a significantly better performance than ARIMA.

Table 3. Wilcoxon signed rank test on MREs (magnitude of relative error) for the five methods.

Model Pair	Seasonal ARIMA	Polynomial Regression	ANN	SampEn-DNN
ARIMA	$<$	$\gg$	$<$	$\ll$
Seasonal ARIMA	$\sim$	$\gg$	$\sim$	$<$
Polynomial Regression	$\ll$	$\sim$	$\ll$	$\ll$
ANN	$\sim$	$\gg$	$\sim$	$<$

From the results in Table 3, it can be found that the outcome of Table 3 validates the conclusions of Table 2. SampEn-DNN has obtained the best performance for time series forecasting of daily BBS new posts, Seasonal ARIMA and ANN are also effective methods for prediction the number of BBS new posts, and Polynomial regression has produced the poorest performance. The results can be explained from two points: (i) SampEn-DNN applies the SampEn method for the dimension optimization of input vectors. Based on the input vectors with smallest complexity, DNN method easily captures the micro-level patterns of BBS new posts time series; (ii) DNN model is a probabilistic, generative model that can learn to probabilistically reconstruct its inputs, and also takes advantage of more neurons to reach the minimum error, thus DNN generates superior performance to the other four methods. In Figure 5, it can also be found that BBS new posts time series show some periodic fluctuations, thus Seasonal ARIMA method shows better performance for this periodically fluctuating dataset than ARIMA. Polynomial regression is in the macro-level and cannot detect the micro-level patterns of post number time series, thus it has produced the poorest performance for BBS new posts time series.

5. Concluding Remarks

In this paper, based on SampEn and DNN, a novel approach SampEn-DNN is proposed for BBS new post number time series modeling and forecasting. The multi-scale sample entropy is adopted to optimize the skipping parameter δ and the dimension of input vector m, and DNN is applied for time series modeling and forecasting based on the optimal parameters. Tianya Zatan board daily new post number was selected as the data source, and extensive experiments based on SampEn-DNN and the state-of-the-art approaches were carried out. From the experimental results, it can be found that, due to the parameter optimization of multi-scale sample entropy, DNN easily learns the micro-level patterns from BBS new posts time series and SampEn-DNN has produced better performance than ARIMA, Seasonal ARIMA, polynomial regression and ANN.

In the future, SampEn-DNN approach will be applied to different tasks on time series modeling and forecasting. For public opinion monitoring, SampEn-DNN will be extended to predict the daily numbers of posts or micro-blogs for more BBSs or micro-blogging websites. Meanwhile, SampEn-DNN approach will be applied to other areas to test its effectiveness, such as weather forecasting, control engineering, etc.

Author Contributions: J.C. and W.Z. conceived and designed the experiments; J.C. and P.Z. performed the experiments; J.C. and W.Z. analyzed the data; and J.C., Y.D. and L.L. wrote the paper.

Funding: This study was supported by National Natural Science Foundation of China under grant Nos. 71601023, 61703010 and L1624049; the Supplementary and Supportive Project for Teachers at Beijing Information Science and Technology University (2018–2020) (5029011103); the Development of University Connotation Scientific Research Level Promotion Project at Beijing Information Science and Technology University (5211823510); and National Key R&D Program of China (2017YFB1400500).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, K.; Shi, L.; Ye, W.; Li, X. A survey of Internet public opinion mining. In Proceedings of the 2014 IEEE International Conference on Progress in Informatics & Computing, Shanghai, China, 16–18 May 2014; pp. 173–179.

2. Tang, X.J. Exploring on-line societal risk perception for harmonious society measurement. *J. Syst. Sci. Syst. Eng.* **2013**, *22*, 469–486. [CrossRef]
3. Pond, P. Twitter time: A temporal analysis of tweet streams during televised political debate. *TVNM* **2016**, *17*, 142–158. [CrossRef]
4. Boecking, B.; Hall, M.; Schneider, J. Event prediction with learning algorithms—A study of events surrounding the Egyptian Revolution of 2011 on the basis of micro blog data. *Policy Internet* **2015**, *7*, 159–184. [CrossRef]
5. Wang, Y.J. Stock market forecasting with financial micro-blog based on sentiment and time series analysis. *Shanghai Jiaotong Univ. Sci.* **2017**, *22*, 173–179. [CrossRef]
6. Chen, X.G.; Duan, S.; Wang, L. Research on trend prediction and evaluation of network public opinion. *Concurr. Comp. Pract. Exp.* **2017**, *29*, 4212. [CrossRef]
7. Yin, F.; Zhang, B.; Su, G.; Zhang, R. Research on the public opinion pre-warning based on the logistic model. In Proceeding of the 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 24–26 November 2017; pp. 381–385.
8. Zhang, J.L. A study on the evaluation of the analytic hierarchy process based Enterprise network public opinion crisis response. *J. Quant. Econ.* **2015**, *32*, 60–63. [CrossRef]
9. Hu, H.; Wen, Y.; Chua, T.S.; Li, X. Cost-optimized microblog distribution over geo-distributed data centers: Insights from cross-media analysis. *ACM Trans. Intel. Syst. Tech.* **2017**, *8*, 1–18. [CrossRef]
10. Lan, Y.X.; Zeng, R.X. Research of emergency network public opinion on propagation model and warning phase. *J. Intel.* **2013**, *32*, 17–19.
11. Yang, C.; Su, G.Q.; Lan, Y.X.; He, Y.H. Research on Emergency network public opinion based on the statistical regression model. *J. Chin. People's Armed Police Force Acad.* **2014**, *30*, 80–83.
12. Wang, H.; Li, Y.; Feng, Z.; Feng, L. ReTweeting analysis and prediction in microblogs: An epidemic inspired approach. *China Commun.* **2013**, *10*, 13–24. [CrossRef]
13. Xu, T.; Xu, X.; Hu, Y.; Li, X. An entropy-based approach for evaluating travel time predictability based on vehicle trajectory data. *Entropy* **2017**, *19*, 165. [CrossRef]
14. Men, B.; Long, R.; Zhang, J. Combined forecasting of streamflow based on cross entropy. *Entropy* **2016**, *18*, 336. [CrossRef]
15. Zhang, W.; Du, Y.H.; Yoshida, T.; Wang, Q.; Li, X. SamEn-SVR: Using sample entropy and support vector regression for bug number prediction. *IET Softw.* **2018**, *12*, 183–189. [CrossRef]
16. Zhang, Y.J.; Ma, J.L.; Liu, J.L.; Xiao, S.Z. Psr-svr network public opinion prediction model. *Metall. Min. Ind.* **2015**, *9*, 787–794.
17. Chen, J.D.; Zhou, X.J.; Tang, X.J. An empirical feasibility study of societal risk classification toward BBS posts. *Syst. Sci. Syst. Eng.* **2018**, *27*, 709–726. [CrossRef]
18. Jin, X.L.; Cheung, C.M.K.; Lee, M.K.O.; Chen, H.P. How to keep members using the information in a computer-supported social network. *Comput. Hum. Behav.* **2009**, *25*, 1172–1181. [CrossRef]
19. Chen, J.D.; Tang, X.J. Ensemble of multiple kNN classifiers for societal risk classification. *Syst. Sci. Syst. Eng.* **2017**, *26*, 433–447. [CrossRef]
20. Qu, Y.; Wu, P.F.; Wang, X. Online community response to major disaster: A study of Tianya forum in the 2008 Sichuan earthquake. In Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS'09), Waikoloa, Big Island, HI, USA, 5–8 January 2009; pp. 1–11.
21. Takens, F. Detecting strange attractors in turbulence. *Lect. Notes Math.* **1981**, *898*, 366–381.
22. Sauer, T.; Yorke, J.A.; Casdagli, M. Embedology. *J. Stat. Phys.* **1991**, *65*, 579. [CrossRef]
23. Richman, J.S.; Moorman, J.R. Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. Heart Circ. Physiol.* **2000**, *278*. [CrossRef]
24. Govindan, R.B.; Wilson, J.D.; Eswaran, H.; Lowery, C.L.; Preißl, H. Revisiting sample entropy analysis. *Phys. A Statist. Mech. Appl.* **2007**, *376*, 158–164. [CrossRef]
25. Kuremoto, T.; Kimura, S.; Kobayashi, K.; Obayashi, M. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing* **2014**, *137*, 47–56. [CrossRef]
26. Shumway, R.H.; Stoffer, D.S. Time series regression and exploratory data analysis. In *Time Series Analysis and its Applications: With R Examples*, 3rd ed.; Springer: New York, NY, USA, 2011.
27. Box, G.E.P.; Jenkins, G. Part one-stochastic models and their forecasting. In *Time Series Analysis: Forecasting and Control*, 3rd ed.; Prentice-Hall: London, UK, 1994.

28. Hazewinkel, M. Taylor series. In *Encyclopedia of Mathematics*; Springer: Dordrecht, The Netherlands, 2001.
29. Zhang, R.; Duan, Y.; Zhao, Y.; He, X. Temperature compensation of Elasto-Magneto-Electric (EME) sensors in cable force monitoring using BP neural network. *Sensors* **2018**, *18*, 2176. [CrossRef] [PubMed]
30. Cantrell, C.D. *'Vector spaces' in Modern Mathematical Methods for Physicists and Engineers*; Cambridge University Press: Cambridge, UK, 2000.
31. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
32. Memisevic, R.; Hinton, G.E. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Comput.* **2010**, *22*, 1473–1492. [CrossRef] [PubMed]
33. Mann, H.B.; Whitney, R. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **1947**, *18*, 50–60. [CrossRef]