Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 9478952, 9 pages
https://doi.org/10.1155/2017/9478952

Research Article

Robust and Adaptive Online Time Series Prediction with Long Short-Term Memory

Haimin Yang,[1] Zhisong Pan,[1] and Qing Tao[2]

[1]*College of Command and Information System, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China*
[2]*1st Department, Army Officer Academy of PLA, Hefei, Anhui 230031, China*

Correspondence should be addressed to Zhisong Pan; hotpzs@hotmail.com

Received 27 August 2017; Revised 23 November 2017; Accepted 3 December 2017; Published 17 December 2017

Academic Editor: Pedro Antonio Gutierrez

Online time series prediction is the mainstream method in a wide range of fields, ranging from speech analysis and noise cancelation to stock market analysis. However, the data often contains many outliers with the increasing length of time series in real world. These outliers can mislead the learned model if treated as normal points in the process of prediction. To address this issue, in this paper, we propose a robust and adaptive online gradient learning method, RoAdam (Robust Adam), for long short-term memory (LSTM) to predict time series with outliers. This method tunes the learning rate of the stochastic gradient algorithm adaptively in the process of prediction, which reduces the adverse effect of outliers. It tracks the relative prediction error of the loss function with a weighted average through modifying Adam, a popular stochastic gradient method algorithm for training deep neural networks. In our algorithm, the large value of the relative prediction error corresponds to a small learning rate, and vice versa. The experiments on both synthetic data and real time series show that our method achieves better performance compared to the existing methods based on LSTM.

1. Introduction

A time series is a sequence of real-valued signals that are measured at successive time intervals [1, 2]. Time series data occur naturally in many application areas such as economics, finance, environment, and medicine and often arrives in the form of streaming in many real-world systems. Time series prediction has been successfully used in a wide range of domains including speech analysis [3], noise cancelation [4], and stock market analysis [5, 6]. The traditional methods of time series prediction commonly use a potential model, for example, autoregressive moving average (ARMA) [7], autoregressive integrated moving average (ARIMA) [1], and vector autoregressive moving average (VARMA) [8], to mimic the data. However, these methods all need to deal with the whole dataset to identify the parameters of the model when facing new coming data, which is not suitable for large datasets and online time series prediction. To address this problem, online learning methods are explored to extract the underlying pattern representations from time series data in a sequential manner. Compared to traditional batch learning methods, online learning methods avoid expensive retraining cost when handling new coming data. Due to the efficiency and scalability, online learning methods including methods based on linear models [9], ensemble learning [10], and kernels [11] have been applied to time series prediction successfully.

Long short-term memory (LSTM) [12], a class of recurrent neural networks (RNNs) [13], is particularly designed for sequential data. LSTM has shown promising results for time series prediction. Its units consist of three gates: input gate, forget gate, and output gate. It is popular due to the ability of learning hidden long-term sequential dependencies, which actually helps in learning the underlying representations of time series. However, the time series data in real world often contains some outliers more or less especially in cyberattacks, which are commonly shown as anomalies in time series data monitoring some measurements of network traffic. Those outliers mislead the learning method in extracting the true representations of time series and reduce the performance of prediction.

In this paper, we propose an efficient online gradient learning method, which we call RoAdam (Robust Adam)

for LSTM to predict time series in the presence of outliers. The method modifies Adam (Adaptive Moment Estimation) [14], a popular algorithm for training deep neural networks through tracking the relative prediction error of the loss function with a weighted average. Adam is based on standard stochastic gradient descent (SGD) method without considering the adverse effect of outliers. The learning rate of RoAdam is tuned adaptively according to the relative prediction error of the loss function. The large relative prediction error leads to a smaller effective learning rate. Likewise, a small error leads to a larger effective learning rate. The experiments show that our algorithm achieves the state-of-the-art performance of prediction.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, we introduce some preliminaries. Section 4 presents our algorithm in detail. In Section 5, we evaluate the performance of our proposed algorithm on both synthetic data and real time series. Finally, Section 6 concludes our work and discusses some future work.

2. Related Work

In time series, a data point is identified as an outlier if it is significantly different from the behavior of the major points. Outlier detection for time series data has been studied for decades. The main work focuses on modeling time series in the presence of outliers. In statistics, several parametric models have been proposed for time series prediction. The point that deviated from the predicted value by the summary parametric model including ARMA [15], ARIMA [16, 17], and VARMA [18] is identified as an outlier. Vallis et al. [19] develop a novel statistical technique using robust statistical metrics including median, median absolute deviation, and piecewise approximation of the underlying long-term trend to detect outliers accurately. There also exist many machine learning models for time series prediction with outliers. The paper [20] proposes a generic and scalable framework for automated time series anomaly detection including two methods: plug-in method and decomposition-based method. The plug-in method applies a wide range of time series modeling and forecasting models to model the normal behavior of the time series. The decomposition-based method firstly decomposes a time series into three components: trend, seasonality, and noise and then captures the outliers through monitoring the noise component. The paper [21] gives a detailed survey on outlier detection.

LSTM has shown promising results for time series prediction. Lipton et al. uses LSTM to model varying length sequences and capture long range dependencies. The model can effectively recognize patterns in multivariate time series of clinical measurements [22]. Malhotra et al. use stacked LSTM networks for outliers detection in time series. A predictor is used to model the normal behavior and the resulting prediction errors are modeled as a multivariate Gaussian distribution, which is used to identify the abnormal behavior [23]. Chauhan and Vig also utilize the probability distribution of the prediction errors from the LSTM models to indicate the abnormal and normal behaviors in ECG time series [24]. These methods are not suitable for online time series prediction because they all need to train on time series without outliers to model the normal behavior in advance. In this paper, our online learning method for time series prediction is robust to outliers through adaptively tuning the learning rate of the stochastic gradient method to train LSTM.

3. Preliminaries and Model

In this section, we formulate our problem to be resolved and introduce some knowledge about Adam, a popular algorithm for training LSTM.

3.1. Online Time Series Prediction with LSTM. In the process of online time series prediction, the desirable model learns useful information from $\{x_1, x_2, \ldots, x_{t-1}\}$ to give a prediction $\widetilde{x}_t$ and then compare $\widetilde{x}_t$ with x_t to update itself, where $\{x_1, x_2, \ldots, x_{t-1}\}$ is a time series, $\widetilde{x}_t$ is the time series data point forecasted at time t, and x_t is the real value. LSTM is suitable for discovering dependence relationships between the time series data by using specialized gating and memory mechanisms.

We give the formal definition of a neuron of a LSTM layer as follows. The jth neuron of a LSTM layer at time t, c_t^j consists of input gate i_t^j, forget gate f_t^j, and output gate o_t^j and is updated through forgetting the partially existing memory and adding a new memory content $\widetilde{c}_t^j$. The expressions of i_t^j, f_t^j, o_t^j, and c_t^j are shown as follows:

$$\begin{aligned} i_t^j &= \sigma\left(W_i x_t + U_i h_{t-1} + V_i c_{t-1}\right)^j, \\ f_t^j &= \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)^j, \\ o_t^j &= \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_{t-1}\right)^j, \\ c_t^j &= f_t^j c_{t-1}^j + i_t^j \widetilde{c}_t^j. \end{aligned} \tag{1}$$

Note that W_i, W_f, W_o, U_i, U_f, and U_o are the parameters of the jth neuron of a LSTM layer at time t. σ is a logistic sigmoid function. V_i, V_f, and V_o are diagonal matrices. h_{t-1} and c_{t-1} are the vectorization of h_{t-1}^j and c_{t-1}^j. The output h_t^j of this neuron at time t is expressed as

$$h_t^j = o_t^j \tanh\left(c_t^j\right). \tag{2}$$

In our model of online time series prediction, we set a dense layer to map the outputs to the target prediction, which is formulated as

$$y = g\left(W_d h_t + b_d\right), \tag{3}$$

where $g(\cdot)$ is the activation function of the dense layer, W_d is the weights, b_d is the bias, and h_t is the vectorization of h_t^j. The objection of our model at time t is to update the parameters $W_t = \{W_i, W_f, W_o, W_d, U_i, U_f, U_o, V_i, V_f, V_o, b_d\}$. The standard process is

$$W_{t+1} = W_t - \eta \nabla l\left(x_t, \widetilde{x}_t\right), \tag{4}$$

where η is the learning rate and $l(x_t, \widetilde{x}_t)$ is the loss function.

RoAdam. Parameters carried over from Adam have the same default values: $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. For parameters specific to our method, we recommend default values $\beta_3 = 0.999$, $k = 0.1$, $K = 10$.
Require: η: learning rate
Require: $\beta_1, \beta_2 \in [0, 1)$: exponential decay rates for moment estimation in Adam
Require: $\beta_3 \in [0, 1)$: exponential decay rate for computing relative prediction error
Require: k, K: lower and upper threshold for relative prediction error
Require: ϵ: fuzz factor
Require: $l(W)$: loss function
Require: W_0: initial value for parameters
$m_0 = v_0 = 0$
$d_0 = 1$
$l(W_0) = l(W_{-1}) = 1$
$t = 0$
while stopping condition is not reached **do**
$g_t = \nabla_W l(W_{t-1})$
$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
$\widehat{m}_t = m_t / (1 - \beta_1^t)$
$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
$\widehat{v}_t = v_t / (1 - \beta_2^t)$
if $\|l(W_{t-1})\| \geq \|l(W_{t-2})\|$ **then**
$r_t = \min\{\max\{k, \|l(W_{t-1})/l(W_{t-2})\|\}, K\}$
else
$r_t = \min\{\max\{1/K, \|l(W_{t-1})/l(W_{t-2})\|\}, 1/k\}$
end if
$d_t = \beta_3 d_{t-1} + (1 - \beta_3) r_t$
$W_{t+1} = W_t - \eta \widehat{m}_t / (d_t \sqrt{\widehat{v}_t} + \epsilon)$
$t = t + 1$
end while
return W_t

Algorithm 1

3.2. Adam. Adam is a method for efficient stochastic optimization, which is often used to train LSTM. It computes adaptive learning rates for individual parameters from estimates of the first moment and the second moment of the gradients, only requiring first-order gradients. Adam keeps an exponentially decaying average of the gradient and the squared gradient:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{5}$$

where m_t and v_t initialized as zero are estimates of the first moment and the second moment and β_1 and β_2 are exponential decay rates for the moment estimates. We can find that m_t and v_t are biased towards zero, when β_1 and β_2 are close to 1. So Adam counteracts these biases through bias correction of m_t and v_t:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t},$$
$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t}. \tag{6}$$

The rule of updating parameters is

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\widehat{v}_t} + \epsilon} \widehat{m}_t, \tag{7}$$

where $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ by default.

4. Method

In this section, we introduce our online gradient learning method, which is called RoAdam (Robust Adam) to train long short-term memory (LSTM) for time series prediction in the presence of outliers. Our method does not directly detect the outliers and adaptively tunes the learning rate when facing a suspicious outlier.

In Algorithm 1, we provide the details of the RoAdam algorithm. The main difference between our algorithm and Adam is r_t, a relative prediction error term of the loss function. The relative prediction error term indicates whether the point is an outlier. The larger value of r_t means the current point is more suspicious to be an outlier. It is computed as $r_t = \|l(W_{t-1})/l(W_{t-2})\|$, where $l(W_{t-1}) = l(x_t, \widetilde{x}_t)$ and $l(W_{t-2}) = l(x_{t-1}, \widetilde{x}_{t-1})$. $l(x_t, \widetilde{x}_t)$ and $l(x_{t-1}, \widetilde{x}_{t-1})$ are the absolute prediction errors of x_t and x_{t-1}. In practice, a threshold is used to scheme to ensure the stability of relative prediction error term. k and K denote the lower and upper thresholds for r_t. We let $r_t = \min\{\max\{k, \|l(W_{t-1})/l(W_{t-2})\|\}, K\}$ (1), if $\|l(W_{t-1})\| \geq \|l(W_{t-2})\|$ and $r_t = \min\{\max\{1/K, \|l(W_{t-1})/l(W_{t-2})\|\}, 1/k\}$ (2) otherwise, which captures both increase and decrease of relative prediction

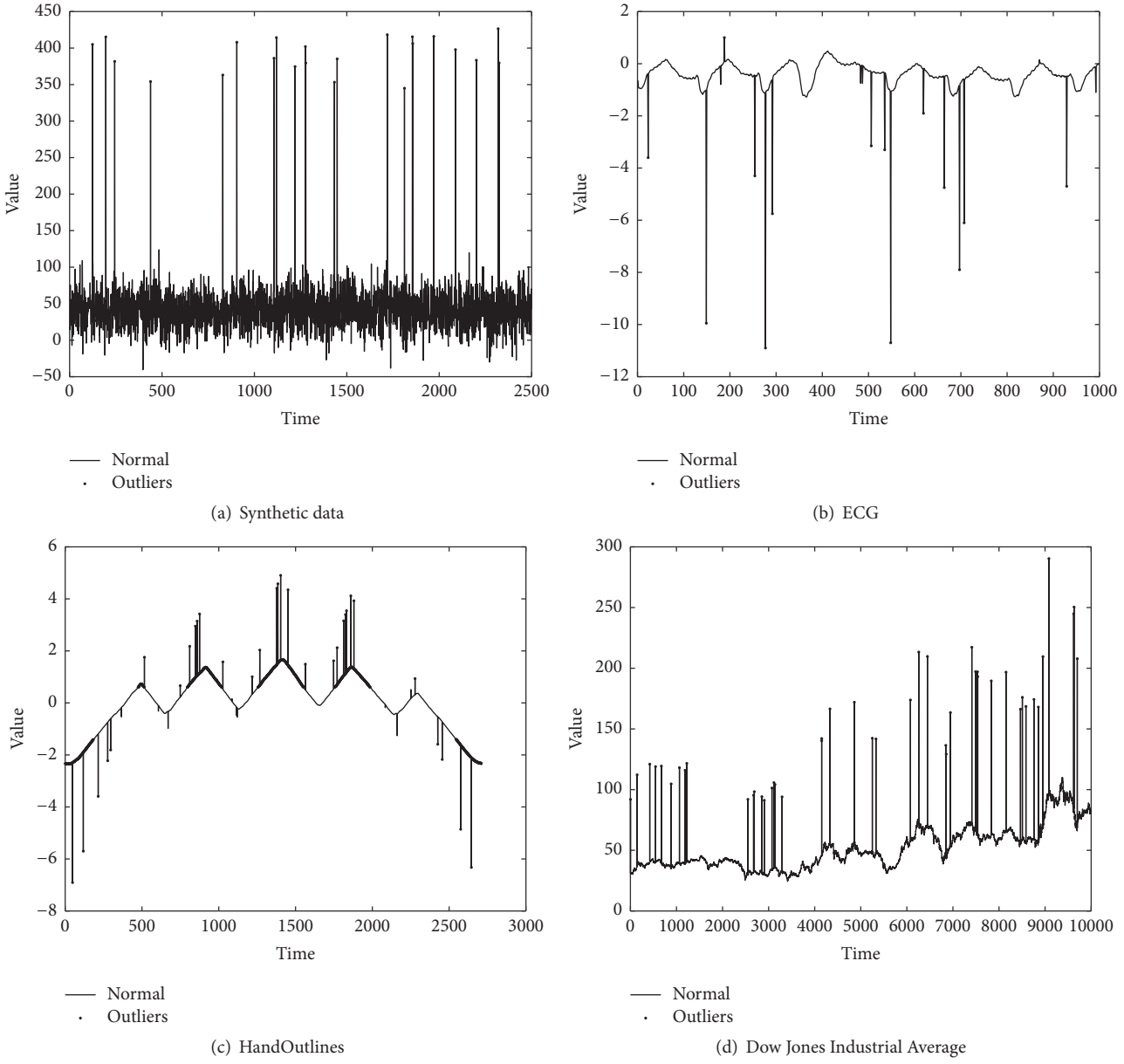

Figure 1: True value of data sets.

Table 1: Different values of r_t.

x_{t-1}	x_t Outlier	x_t Normal
Outlier	(1)	(2)
Normal	(2)	(1)

error. Our settings consider different situations when the preceding point x_{t-1} and current point x_t are at different status. The details are listed in Table 1.

To get a smoother estimate, we compute the relative prediction error with a weighted average. The final result d_t is $\beta_3 d_{t-1} + (1 - \beta_3) r_t$. Here the effect of β_3 is the same as β_1 and β_2 in Adam. In general, RoAdam is modified in the basis of Adam through multiplying the denominator $\sqrt{\hat{v}_t}$ with d_t. The large value of d_t corresponds to a small learning rate, and vice versa.

5. Experiment

In this section, we illustrate the performance of our proposed algorithm RoAdam compared to RLSTM, SR-LSTM, and RN-LSTM on both synthetic data and real time series.

5.1. Experiment Setup. RLSTM means real time LSTM, which updates the model using the newly coming data without considering the effect of outliers. SR-LSTM stands for LSTM with suspicious point removal. The difference between SR-LSTM and RN-LSTM is that once a suspicious point is

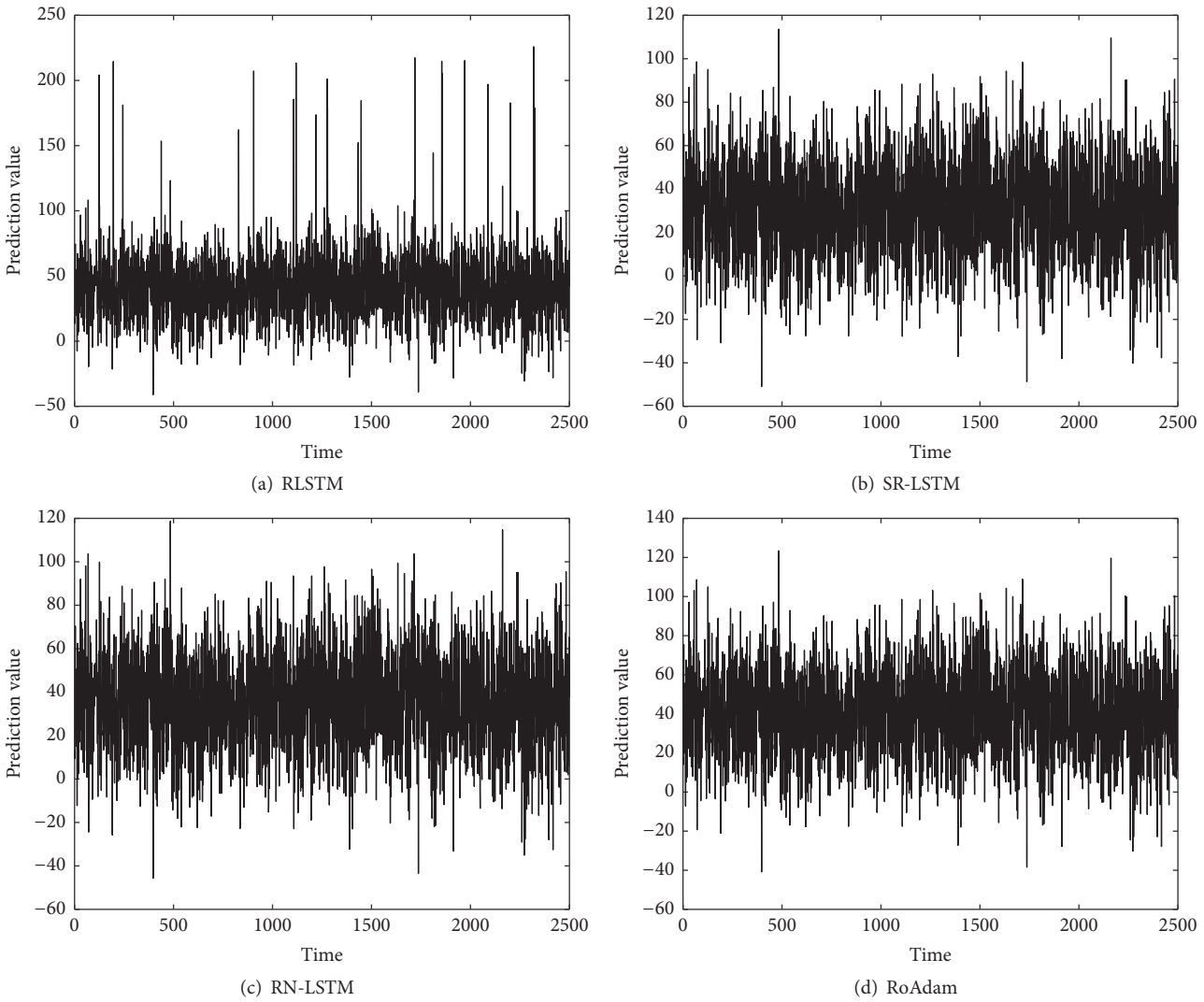

Figure 2: Prediction value of different algorithms on synthetic data.

detected as an outlier, SR-LSTM does not update on this point and RN-LSTM updates using a recent normal point. They both use the method proposed in [25] to detect the outlier. In addition, all the algorithms use the same LSTM model besides the optimizer. RLSTM, SR-LSTM, and RN-LSTM adopt the original Adam optimizer. The LSTM model has 3 layers and the number of neurons in each layer is 400. The mean squared error is chosen as the loss function and the L2 regularization with 0.0001 penalty is used. The parameters of RoAdam carried from Adam have the same default values: $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. For parameters specific to our method, we try different values and recommend default values $\beta_3 = 0.999$, $k = 0.1$, and $K = 10$.

5.2. Data Sets. To examine the prediction performance, we evaluate all the previous algorithms on synthetic data and real time series.

5.2.1. Synthetic Data. The synthetic data is sampled from a Gaussian distribution with the corresponding mean $u \in [0, 100]$ and variance $\sigma \in [10, 30]$ plus the trend component $T \in [-0.5, 0.5]$. The length l is 2,500. The outliers are injected based on a Bernoulli distribution identified by $\alpha = 0.01$ and $l \cdot \alpha$ is the expected number of outliers. The values of outliers are also sampled from a Gaussian distribution with mean $u \in [0, 1000]$ and variance $\sigma \in [10, 30]$. The expression of x_t is

$$x_t = \begin{cases} x + T, x \sim N(u, \sigma), u \in [0, 100], \sigma \in [10, 30], T \in [-0.5, 0.5], & \text{when } x_t \text{ is a normal point;} \\ x, x \sim N(u, \sigma), u \in [0, 1000], \sigma \in [10, 30], & \text{when } x_t \text{ is an outlier.} \end{cases} \tag{8}$$

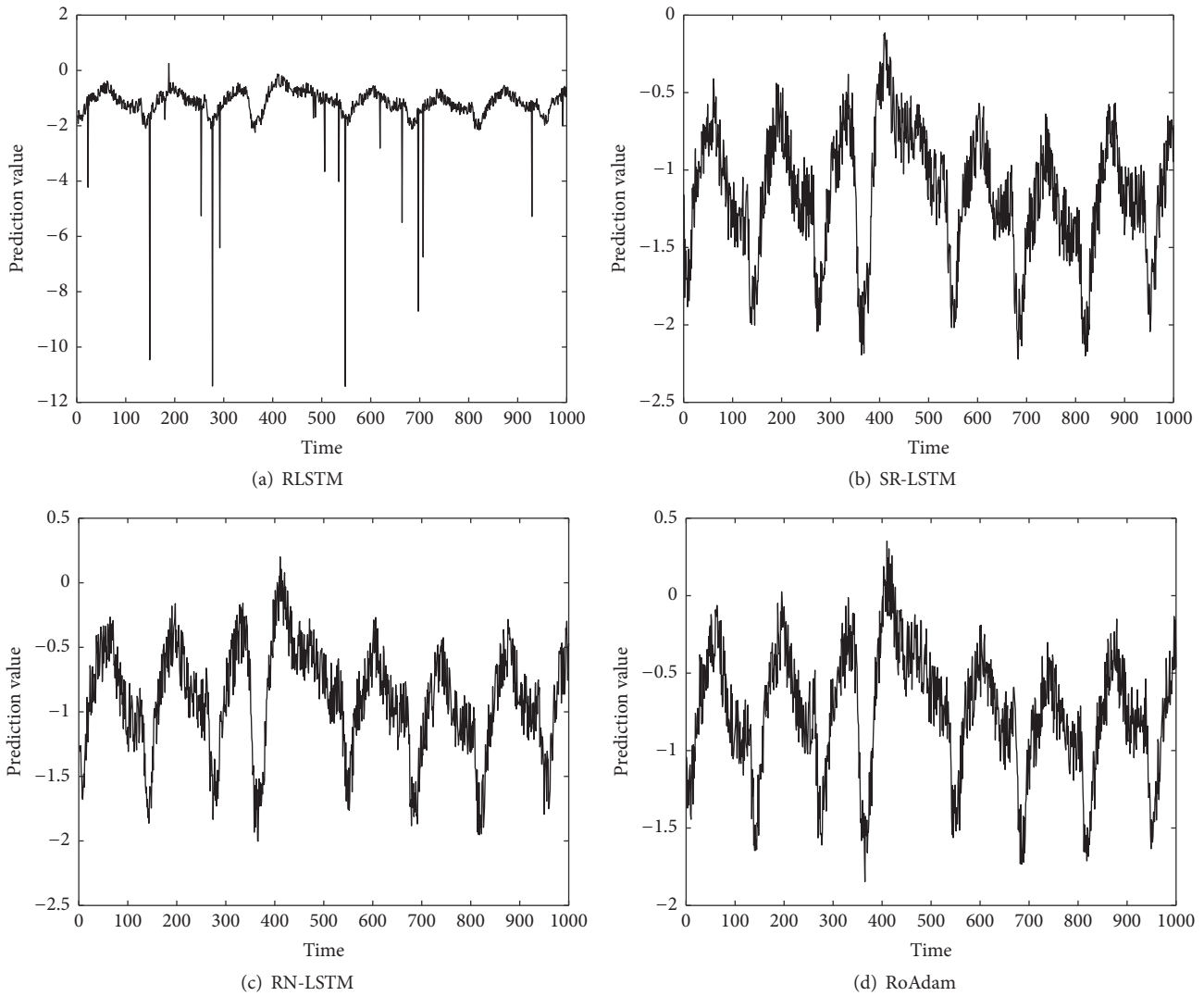

Figure 3: Prediction value of different algorithms on ECG.

5.2.2. Real Time Series. The first time series data is ECG data, which consists of 70 series of 1000 ECG measurements [26]. We choose 100 samples from ECG data set. The second one is HandOutlines, which is from the commonly used UCR (http://www.cs.ucr.edu/~eamonn/time_series_data/.). The last time series data is daily index of Dow Jones Industrial Average (DJIA) during years 1885–1962. We randomly select 1% of each real time series as outliers, whose values are 2 or 3 times bigger than the true ones. Figure 1 presents the true value of synthetic data and real time series. The x-axis is time (the number of samples) and the y-axis is true value.

5.3. Experimental Results. In this section, we test RMSE of the algorithms mentioned above to examine the effectiveness and efficiency.

$$\text{RMSE} = \frac{1}{T}\sum_{t=1}^{T}\left(X_t - \widetilde{X}_t\right)^2. \tag{9}$$

RMSE allows us to compare errors with the number of samples increasing. In addition, we average the results over 100 runs for stability.

Table 2 shows the RMSE of different algorithms both on synthetic data and real time series. We can find that RoAdam outperforms all the other algorithms on RMSE. Figures 2–5 visualize the prediction value of all the algorithms on synthetic data and real time series. The x-axis is time (the number of samples) and the y-axis is prediction value. We can observe that the prediction value produced by RLSTM has oscillations around outliers. It indicates that the prediction performance of RLSTM is indeed affected by outliers. Although SR-LSTM, RN-LSTM, and RoAdam have almost the same shape of prediction value, RoAdam has the least RMSE. The reason may be that SR-LSTM and RN-LSTM may lose some information of the normal points when they are mistaken outliers.

Table 2: RMSE on synthetic data and real time series.

Algorithm	Data			
	Synthetic	ECG	HandOutlines	DJIA
RLSTM	0.7606	0.8505	0.9756	1.8454
SR-LSTM	0.7329	0.8323	0.9411	1.7574
RN-LSTM	0.7218	0.8217	0.9376	1.6218
RoAdam	**0.4946**	**0.5626**	**0.7633**	**1.3875**

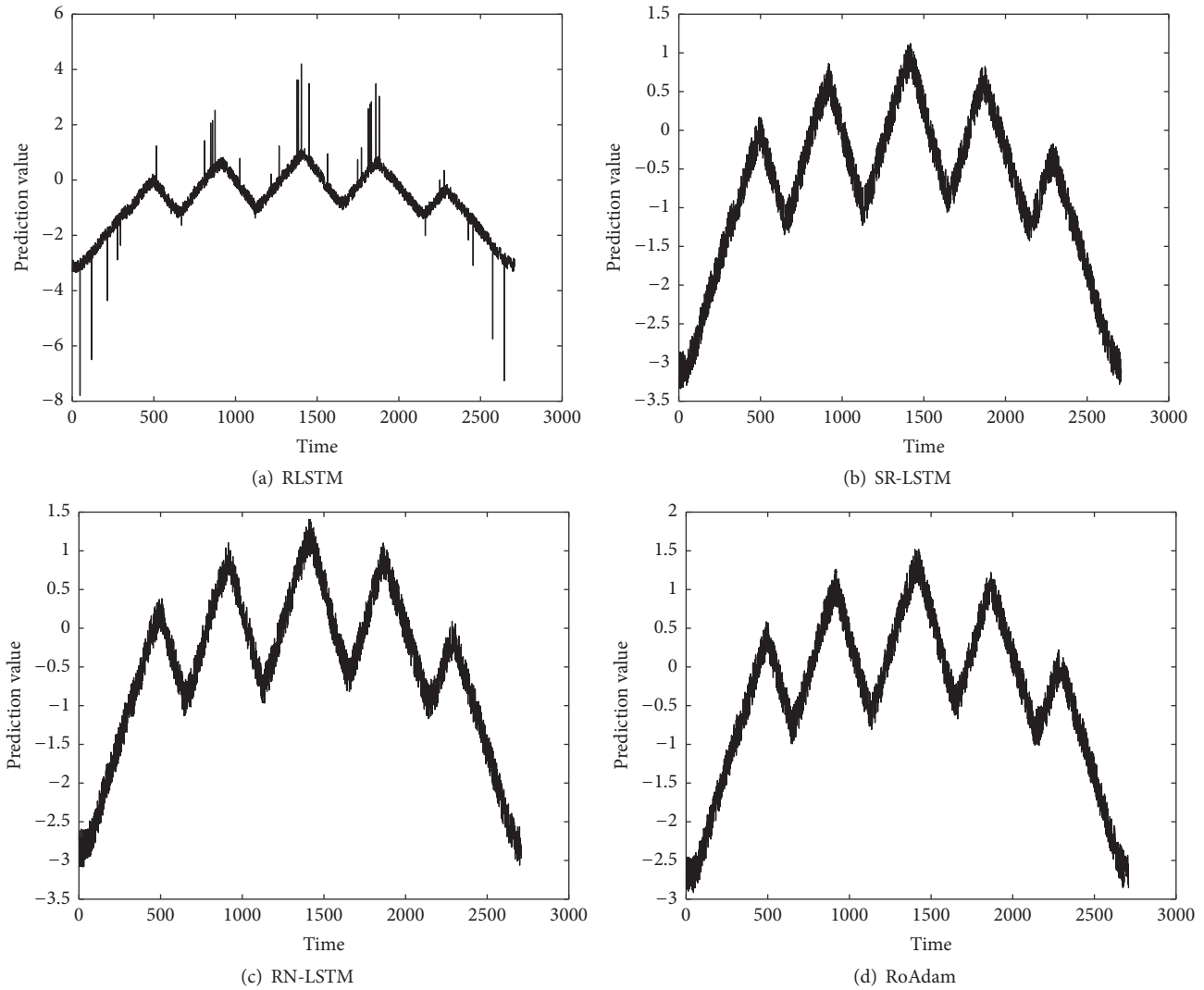

Figure 4: Prediction value of different algorithms on HandOutlines.

6. Conclusions

In this paper, we propose an efficient online gradient learning method, RoAdam for LSTM, to predict time series, which is robust to outliers. RoAdam is modified on the basis of Adam, a popular stochastic gradient algorithm for training deep neural networks. Through tracking the relative prediction error of the loss function with a weighted average, this method adaptively tunes the learning rate of the stochastic gradient method in the presence of outliers. In the process of prediction, the large value of the relative prediction error corresponds to a small learning rate, and vice versa. The experiments on both synthetic data and real time series show that our method achieves less prediction error compared to the existing methods based on LSTM.

It remains for future work to study whether our approach could be extended to time series prediction with missing data.

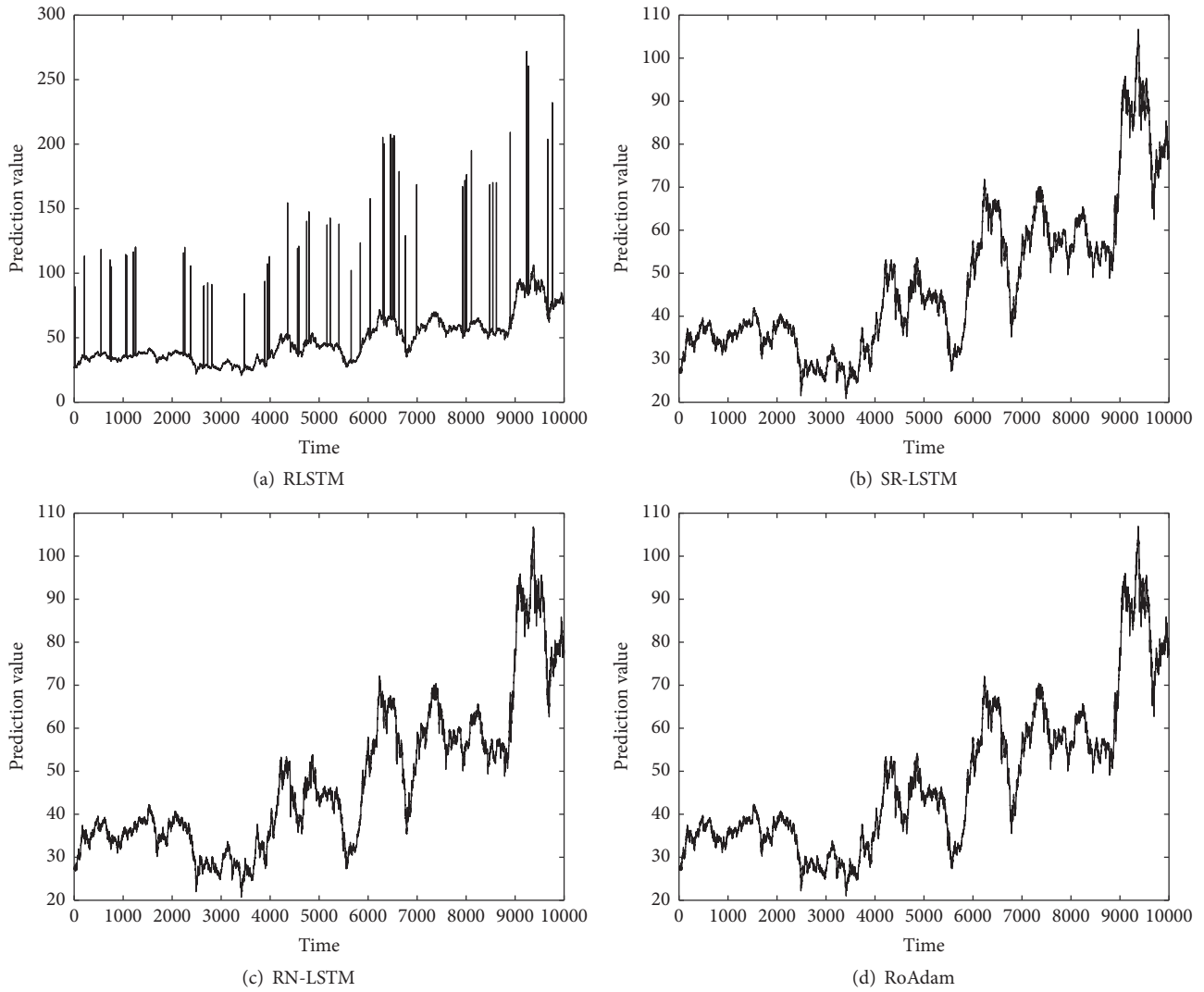

Figure 5: Prediction value of different algorithms on DJIA.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Haimin Yang participated in the draft writing and experiments. Zhisong Pan and Qing Tao participated in the design of algorithms and commented on the manuscript.

Acknowledgments

Our work is supported by the National Natural Science Foundation of China (nos. 61473149 and 61673394).

References

[1] J. D. Hamilton, *Time Series Analysis*, Princeton University Press, New Jersey, NJ, USA, 1994.

[2] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, Springer, New York, NY, USA, 2ND edition, 2006.

[3] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*, Englewood Cliffs, N.J., Prentice-Hall, New Jersey, NJ, USA, 1978.

[4] J. Gao, H. Sultan, J. Hu, and W.-W. Tung, "Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 237–240, 2010.

[5] C. W. J. Granger and P. Newbold, *Forecasting Economic Time Series*, Academic Press, New York, NY, USA, 1986.

[6] M. Nerlove, D. M. Grether, and J. L. Carvalho, *Analysis of Economic Time Series: A Synthesis*, Academic Press, New York, NY, USA, 1979.

[7] J. L. Rojo-Alvarez, M. Martınez-Ramon, M. de Prado-Cumplido et al., "Support vector method for robust ARMA system identification," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 155–164, 2004.

[8] R. S. Tsay, *Multivariate Time Series Analysis: with R And Financial Applications*, John Wiley and Sons, New Jersey, NJ, USA, 2014.

[9] O. Anava, E. Hazan, S. Mannor et al., "Online learning for time series prediction," *Journal of Machine Learning Research*, vol. 30, pp. 172–184, 2013.

[10] L. L. Minku and X. Yao, "DDD: a new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2012.

[11] C. Richard, J. C. Bermudez, and P. Honeine, "Online prediction of time serise data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2008.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the in Proceedings of International Conference on Learning Representations (ICLR '15)*, 2015.

[15] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley & Sons, New Jersey, NJ, USA, 1978.

[16] D. M. Hawkins, *Identification of Outliers*, Chapman and Hall, London, UK, 1980.

[17] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New Jersey, NJ, USA, 1987.

[18] R. S. Tsay, "Time series model specification in the presence of outliers," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 132–141, 1986.

[19] O. Vallis, J. Hochenbaum, and A. Kejariwal, "A novel technique for long-term anomaly detection in the cloud," in *Proceedings of 2014 6thUSENIX Workshop on Hot Topics in Cloud Computing*, 2014.

[20] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '15)*, pp. 1939–1947, Australia, August 2015.

[21] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

[22] Z. C. Lipton, D. C. Kale, C. Elkan et al., Learning to diagnose with lstm recurrent neural networks, https://arxiv.org/pdf/1511.03677.pdf.

[23] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory networks for anomaly detection in time series," in *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN '15)*, pp. 89–94, April 2015.

[24] S. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA '15)*, October 2015.

[25] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 2, pp. 240–254, 1994.

[26] A. J. Bagnall and G. J. Janacek, "Clustering time series from ARMA models with clipped data," in *Proceedings of The 2004 ACM SIGKDD International Conference*, p. 49, August 2004.