

# A Novel Framework for Online Sales Burst Prediction

Rui Chen and Jiajun Liu\*

Big Data Analytics and Intelligence Lab,  
School of Information,  
Renmin University of China  
{r\_chen, jiajunliu}@ruc.edu.cn

**Abstract.** With the rapid growth of e-commerce, a large number of online transactions are processed every day. In this paper, we take the initiative to conduct a systematic study of the challenging prediction problems of sales bursts. Here, we propose a novel model to detect bursts, find the bursty features, namely the start time of the burst, the peak value of the burst and the off-burst value, and predict the entire burst shape. Our model analyzes the features of similar sales bursts in the same category, and applies them to generate the prediction. We argue that the framework is capable of capturing the seasonal and categorical features of sales burst. Based on the real data from JD.com, we conduct extensive experiments and discover that the proposed model makes a relative MSE improvement of 71% and 30% over LSTM and ARMA.

**Key words:** burst prediction · e-commerce

## 1 Introduction

E-commerce websites have become an ubiquitous mechanism for online shopping. Devendra first defined that electronic commerce, commonly known as e-commerce or eCommerce, consists of the buying and selling of products or services over electronic system such as internet and other computer network[1]. The effects of e-commerce have reached all areas of business, from customer service to new product design[2].

According to statistics in 2015, turnover of E-commerce has reached 18.3 trillion in China, up by 36.5% [4]. This can also be seen from the huge trading volume on some special shopping festivals in China. In the shopping festival of 11th Nov, 2016, the single-day merchandise trade of tmall.com reached 912.17 billion yuan, up by 61%. For the logistics industry, the number of orders will have a corresponding explosive growth. Nowadays, the e-commerce platforms usually launch promotional activities on a particular category of products. Hence the products in the same category always show similar sales changes.

In e-commerce field, time series prediction is wildly used. A large number of methods have been developed and applied to time series forecasting problems,

---

\* Jiajun Liu is the corresponding author.

such as sophisticated statistical methods[16] and neural network[15]. For any product, the sales series can increase or fall sharply, which we called spikes or bursts. The prediction of bursts is beneficial in several aspects, such as the storage optimization of the suppliers and the stability maintenance of the e-commerce website. In the existing studies about burst prediction, most of them are concerned about predicting the bursty features. Few of them focus on the prediction of the entire sales burst shape.

In this paper, we study the task of analyzing the bursty features of product sales, and propose a model framework to predict sales burst. The major contributions of this paper include:

1. We define a new problem of time series prediction that mainly concerns about the bursts. To formulate this problem, we split it into three parts: detecting bursts, predicting the features and predicting the shape of bursts.
2. We propose a novel framework to capture the seasonal and categorical features for predicting the entire burst series. We also take the initiative to use the reshaped nearest neighbors to simulate the burst shape.
3. We conduct extensive experiments on real datasets from sale records on JD.com and evaluate the advantages and characteristics of the proposed model. The results show that our model has a significant improvement.

The rest of the paper is organized as follows. In Section 2 we present the background and relevant literature of the problem studied. In Section 3 we give the problem formulation and notations. Datasets with the preprocessing steps are shown in Section 4. In Section 5 we give the intuition of our model and describe the layers of our model framework. Experimental results on real life data are shown in Section 6. Finally, we conclude our work in Section 7.

## 2 Related Work

In recent years, many methods have been developed and applied to time series forecasting problems, such as sophisticated statistical methods[16] and neural networks[15]. Schaidnager et al. [14] presented a parametrized time series algorithm that predicts sales volumes with variable product prices and low data support.

Burst, defined as a brief period of intensive activity followed by long period of nothingness[3], is a common phenomenon in time series. Most existing studies about burst prediction, mainly applied in social networks. Lin et al. [11] proposed a framework to capture dynamics of hashtags based on their topicality, interactivity, diversity, and prominence. Ma et al. [12] predicted the popularity of hashtags on daily basis. [8, 9] introduced a method to predict the burst of Twitter hashtags before they actually burst. Bursts of topics, sentiments and questions have been demonstrated to have a predictive power of product sales. Gruhl et al. [6] proposed to use online postings to predict spikes in sales rank. Such methods have been proved to be effective and achieved good performance. However, how to design a framework that can predict the entire burst shape, is

yet to be answered. Inspired by the success of these methods, we divided the problem into several parts and investigate using a framework with three layers.

### 3 Problem Formulation and Notations

The sales of a product can be formed a time series  $\langle x_1, x_2, \dots, x_t, \dots \rangle$ .  $x_t$  donates the count of sales at the  $t$ -th time interval. Features of a sales burst contains the start time, start value, burst time, peak value, period, off-burst time, etc. Given a product  $p$  in category  $c$ , we consider a prediction scenario: a product  $p$  has a historical sales series  $x$ , at time  $t$ , will the sales of product burst in the near future? If the product will be a bursting one, what's the shape of the entire burst?

In the rest of the paper, the technical details of the algorithms will be described mostly in vector forms. All the assumptions and notations mentioned in this paper are listed in Table 1.

**Table 1.** Table of notations

Notation	Description
$X_c$	the set of time series in category $c$
$\tilde{X}_c$	the set of reshaped burst series in category $c$
$x \in \mathbb{R}^d$	the time series of length $d$
$\tilde{x} \in \mathbb{R}^m$	the reshaped burst series of length $m$
$w \in \mathbb{R}^k$	the window with length $k$
$s \in \mathbb{R}^k \sim w$	the time sequence within the window with length $k$
$lmaxs$	the set of local max points in a time series
$lmins$	the set of local min points in a time series
$B_c$	the set of bursty feature vector for category $c$
$b = (s, b, p, T, e)$	the bursty feature vector: start time, burst time, peak value, period, off-burst value

## 4 Dataset and Set Up

### 4.1 Dataset Description

To support the comprehensive experiments, we collected product information and transaction records from a real-life e-commerce website, JD.com. The original dataset contains two parts. The first part is a 139 million set of transaction-records from 1/1/2008 to 12/1/2013, including user ID, product ID, and the purchasing date. Single-day sale of one product ranges from 0 to 7802. The second part is a 0.25 million set of product information, including product ID, categories, brand and product name. There are 167 categories in total.

### 4.2 Preprocess

In order to prepare the datasets required in the framework, we conducted the following steps.

**Filtering** Since random factors have a great impact on time series when the whole number of records is small, we select records from 2010 to 2013, which constitute 90% of all sales. Besides, we choose products with good selling frequency, using a threshold of 100 records in total.

**Time series generation** For each product, We calculate the daily sales volume as the value of each node in the time series with the transaction records.

**Smoothing** After the above process, we will get plenty of raw time series with a lot of sharp rise and fall. To detect bursts, we smooth the time series. In specific, we perform discrete kalman filter [7] with exponentially weighted moving average(EWMA) on each time series. The smoothed series in each category will form a new time series set.

**Sample bursts extraction** For category  $c$ , we have time series set  $X_c$ . Suppose  $X \in X_c, X = x_1, x_2, \dots, x_d$ , the sample bursts are extracted through the following steps. First, we find all the local maximums  $lmaxs$  and local minimums  $lmins$  for  $X$ . For the sequence between each local minimum pair,  $(lmin_i, lmin_{i+1})$ , there must be a  $lmax_i$ . So we judge whether the sequence contains a burst by the slope of  $(lmin_i, lmax_i)$ . The threshold  $delta$  is computed by the standard deviation of  $X$  and window size  $k$ .

In the process of rise and fall, sales series may have small fluctuations. Once a new burst  $b$  is detected, we will judge whether it should be merged with the previous burst  $b'$  with the features of  $b$  and  $b'$ . In specific, there are two situations when we choose to merge two bursts:

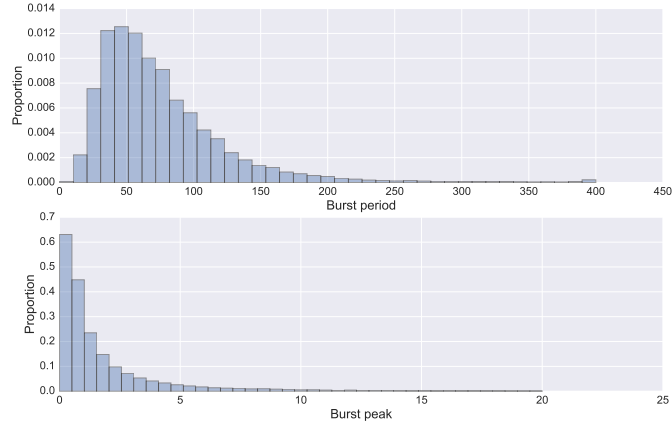
- Burst  $b'$  appeared in the falling process of burst  $b$ : merge  $b$  and  $b'$  as a new burst  $b'' = (s, p, b, T + T', e')$
- Burst  $b$  appeared in the rising process of burst  $b'$ : merge  $b$  and  $b'$  as a new burst  $b'' = (s, p', b', T + T', e')$

**Reshaping** Based on the assumption that products in the same category have similar bursty features, we propose to generate a new type of dataset for each category with the burst series. The burst series with period  $T$  in each category are reshaped into a fixed length  $m$ :

$$\bar{X} = \bar{x}_1, \bar{x}_2, \dots, \bar{x}_m, \bar{x}_i = x_k + (x_{k+1} - x_k) \times (i \times \frac{T}{m} - k), k = \lfloor i \times \frac{T}{m} \rfloor$$

All reshaped bursts will form a new dataset  $\bar{X}_c$ .

Hence, for each category  $c$ , we have three datasets: the smoothed time series dataset  $X_c$ , the reshaped bursts dataset  $\bar{X}_c$ , and the bursty feature dataset  $B_c$ . The categories with less than  $N$  products (we set  $N$  as 100) are ignored in this process, and we keep 64 categories in total. Figure 1 shows the distribution of the bursty feature datasets.



**Fig. 1.** The distribution of bursty features

## 5 The Model

### 5.1 Intuition

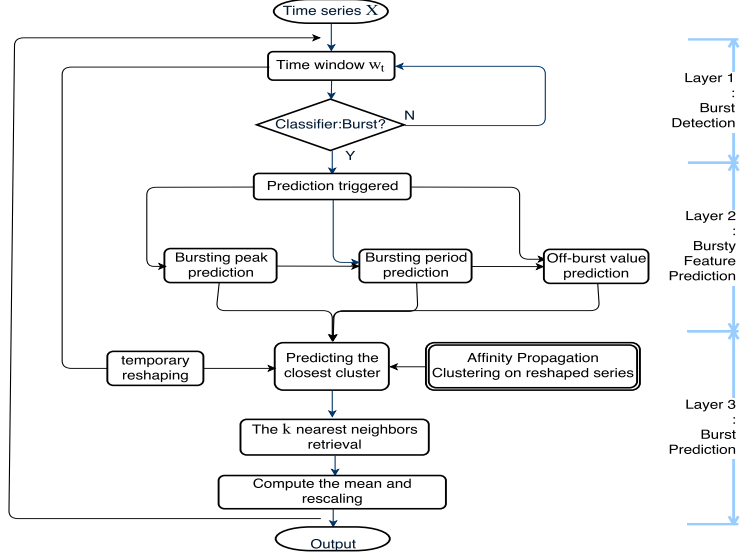
In the existing studies about burst prediction, most of them focus on how to predict the bursty features, rather than the entire burst. The performance of models are improved by the usage of various features. For sales time series, the extra information is much less than topics or hashtags on websites. Most of the time, we just have the transaction records and basic information about the products. When we are faced with a new product with no historical data, the prediction problem will become more difficult. In the absence of sufficient information and data, we need to figure out a proper approach to predict the entire burst shape. Next we will present an overview of the model framework.

### 5.2 Model Overview

We propose a model framework to predict the entire burst shape of online products, shown in Figure 2. The model has three layers, namely the burst detection layer, bursty feature prediction layer, and the burst shape prediction layer. The final layer will generate the output prediction of the whole model. The model implements the following work flow:

### 5.3 Burst Detection

The burst detection layer aims to detect the start of a burst. For time series  $X$ , we let the window  $w$  slide with the timeline from the start point of  $X$ . At time  $t$ , we get a sequence  $s_t = x_t, x_{t+1}, \dots, x_{t+d}$ , and we will use this sequence as the input of the classification model. The sliding window is to simulate and solve the real situation in e-commercial websites.



**Fig. 2.** The framework of burst shape prediction model

For the classification part, we apply a SVM-based method to solve this problem. We first define a feature space based on the time series features in [10], and add some new features such as  $max\_value$ ,  $min\_value$ ,  $id\_max$  and  $id\_min$ . The feature vector  $f$  is the input of the SVM model, and we will get the class label as output, indicating whether it will be a bursting one.

#### 5.4 Bursty Feature Prediction

Once a burst has been detected, we want to know its bursty features, such as how high the peak will reach, when it will be off-burst and what the off-burst value is. In this layer, these features are predicted by three different regression model, named  $M^{(1)}$ ,  $M^{(2)}$  and  $M^{(3)}$ . The predicted results in  $M^{(i-1)}$  and  $f$  will form a new vector, as the input for  $M^{(i)}$ . For each category  $c$ , We will train several different types of  $M^{(i)}$ , and choose the one with smallest mean square error(MSE).

#### 5.5 Burst Shape Prediction

The burst shape prediction layer aims to forecast the whole burst shape using the bursty features  $b$  and the time sequence  $s_t$ . For each category  $c$ , a clustering model is pre-trained with the reshaped bursts dataset  $\bar{X}_c$ .

The key idea is to use the corresponding part and the bursty features as the measure of similarities. First, with the predicted period  $T$ , the time sequence  $s_t$  will be reshaped into a new sequence of fixed length  $l = k \times (d/m)$ ,  $s_t = s_1, s_2, \dots, s_l$ . For each cluster center of reshaped bursts  $\bar{x} = \bar{x}_1, \bar{x}_2, \dots, \bar{x}_m$ ,

we only use the same part  $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_l$  to calculate the similarity of two sequences and find the predicted cluster  $s_t$ . There are plenty of methods to measure similarity, such as Euclidean distance, cosine similarity and so on. For time series, Dynamic Time Warping(DTW) [13] is good alternative, which is designed to find an optimal alignment between two given (time-dependent) sequences. Then, we can find all reshaped bursts in  $s_t$ , and calculate new similarities between  $s'_t$  and their corresponding parts. Here, the new similarity contains sequence similarity, absolute value of period similarity, peak similarity and off-burst value similarity. The weight of the first part is highest and for other parts we assign same weights. We choose the highest ranking  $k$  sequences, calculate its mean series  $\bar{x}$ , and rescale it with the period  $T$  as the output prediction. The rescaling method uses the reverse process of reshaping:  $X = x_1, x_2, \dots, x_T, x_i = \bar{x}'_k + (\bar{x}'_{k+1} - \bar{x}'_k) \times (i \times \frac{m}{T} - k), k = \lfloor i \times \frac{m}{T} \rfloor$ .

## 6 Experiments and Evaluation

**Setup** For each individual dataset, we randomly divide the samples into three folds: the training set, the validation set and the test set, with the proportion of 3:1:1. The model is trained using the training set, and is then tested on the validation set. Such cross-validation is performed on the same individual dataset for ten times with random splits, and the reported performance is the averaged value cross the ten iterations. Finally we test the model on the test set and report the performance.

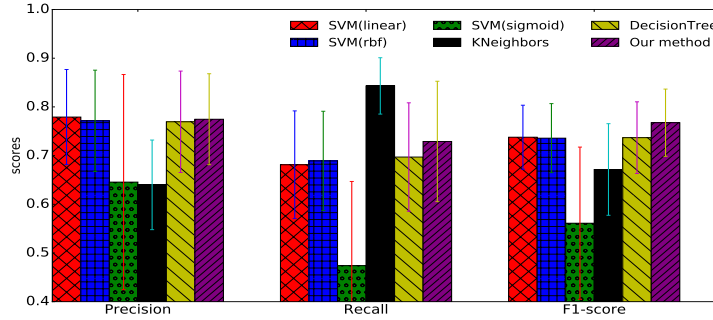
### 6.1 Evaluation of the Burst Detection Layer

In this layer, we set a maximum number of samples as 30000 to reduce training time, and randomly select the same number of positive and negative samples from the training dataset. Before the training, we evaluate the contributions of features with a L1-based linear SVC model, and then reduce the dimensionality of input feature vector of the classifier, a SVC model with rbf kernel. Precision, Recall and F1-score are reasonable metrics for the evaluation of this layer.

**Table 2.** Performance comparison of detecting bursts (partial data)

ID	F1-score					
	SVM(linear)	SVM(rbf)	SVM(sigmoid)	KNeighbors	DecisionTree	Our method
0	0.78	0.78	0.31	0.69	0.77	<b>0.85</b>
1	0.81	0.83	0.61	0.63	0.84	<b>0.85</b>
2	0.78	0.78	0.77	0.68	0.79	<b>0.80</b>
3	<b>0.75</b>	<b>0.75</b>	0.63	0.87	<b>0.75</b>	<b>0.75</b>
4	0.70	0.69	0.69	<b>0.73</b>	0.70	0.69
5	0.59	0.59	0.48	0.59	0.57	<b>0.64</b>
6	0.73	0.73	0.65	0.70	<b>0.74</b>	<b>0.74</b>
7	0.87	0.89	0.61	0.70	0.90	<b>0.91</b>
8	0.73	<b>0.74</b>	0.68	0.69	0.73	<b>0.74</b>
9	<b>0.77</b>	<b>0.77</b>	0.71	0.67	<b>0.77</b>	<b>0.77</b>

Table 2 and Figure 3 show the performance comparison of the first layer, detecting bursts. It can be observed that our model shows the best performance of F1-score for 90% categories listed in the table, and achieves the highest F1-score of 0.77 on average. Besides, our model achieves relatively high precision and recall scores on average, both scoring over 0.72. The performance of K-Nearest Neighbors is quite different, which achieves the best recall score 0.84 and the worst precision 0.63. SVM models with sigmoid kernel is the most ineffective and unstable one, with the deviations of all scores over 0.15. SVM models with linear kernel and rbf kernel always have similar performance.



**Fig. 3.** Mean average classification performance with standard deviation (Precision /Recall/F1-score: the higher the better)

In practical applications, there may be different requirements on precision and recall. These requirements can be satisfied by training optimal prediction models using different types of F-scores, and select the one with best score.

## 6.2 Evaluation of the Bursty Features Prediction Layer

For each category  $c$ , we train multiple regression models, select the best one, that is, the one with smallest MSE, as prediction model for current  $c$ . Using features we generated, we can find typical features for each  $c$ . In specific, we use the feature selection algorithm  $\chi^2$ , to compute the score of each feature, and select the  $K = 10$  highest scoring features.

Table 3 studies the performance of the proposed method and the competitive methods on 64 individual datasets of different category, evaluated by average HitRate(HR)@20% and 50%, Mean Squared Error (MSE) and Mean absolute Relative Error (MRE). HitRate is the percentage of times when the relative error is within a specific range. For example, for a set of targets that equal 10, if 50% of time the predicted value is within [6-14], the HitRate@40% is 0.5.

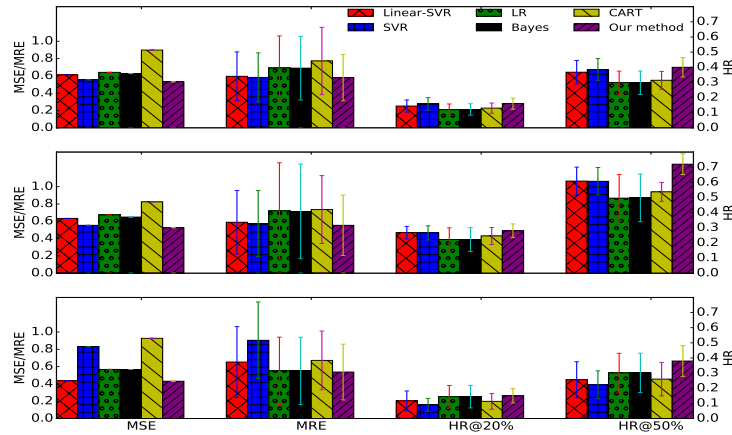
The highlighted numbers in red, blue and black indicate the winners on each model under the corresponding metric. To compare the methods quantita-



**Table 3.** Performance comparison on scores of predicting peak value, period and off-burst value of bursts (HR: the higher the better, MSE/MRE: the lower the better)

Method	Avg MSE	Avg MRE	Avg HitRate@20%	Avg HitRate@50%
Linear SVR	29.91—12096.26—3.88	1.91—50.90—0.69	0.14—0.27—0.12	0.37—0.61—0.26
SVR	26.76—10573.25—7.29	1.88—49.65—0.94	0.16—0.27—0.09	0.38—0.61—0.22
Linear Regression	30.95—12861.00—4.98	2.23—62.28—0.57	0.12—0.22—0.14	0.30—0.50—0.30
Bayes	30.22—12399.39—4.97	2.22—61.72—0.57	0.12—0.22—0.14	0.30—0.50—0.30
CART	42.95—15538.12—7.53	2.49—63.28—0.70	0.13—0.25—0.11	0.31—0.54—0.26
Our method	<b>25.66—10032.34—3.78</b>	<b>1.87—47.85—0.56</b>	<b>0.16—0.28—0.15</b>	<b>0.40—0.68—0.38</b>

tively, we also provide Figure 4 (MRE/MSE is normalized with the maximum MRE/MSE among the methods in each entry).



**Fig. 4.** Mean average prediction performance with standard deviation. The three subfigures represent the task of predicting the peak value, period and off-burst value (HR: the higher the better, MSE/MRE: the lower the better)

Our model achieves the best performance of MSE and MRE and the highest score of HR@20% and HR@50% with a relatively low deviation. For example, in the task of the peak value prediction, we find our models performance and the average of other methods performance under MSE, MRE, HR@20% and HR@50% are 25.66 vs. 32.16 (unnormalized), 1.87 vs. 2.15 (unnormalized), 0.16 vs. 0.13 and 0.40 vs. 0.33 respectively, showing that our model makes a relative improvement of 20%, 13%, 23% and 21% respectively. In the evaluation of forecasting the burst period and off-burst value, our model has a optimal performance in HR@50%, reaching 0.68 and 0.38 on average.

It can also be observed that Linear Regression and Bayes Ridge Regression show similar performance on HR@20% and HR@50%. Linear SVR and SVR

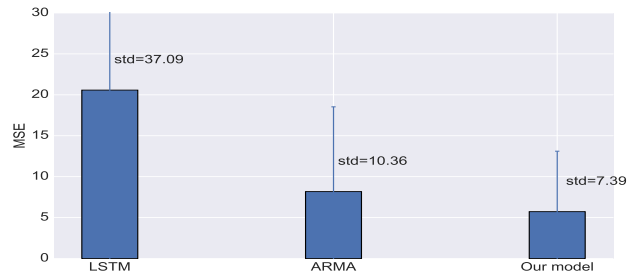
with `fbfkernel` have different performance under MSE and MRE. The CART method have the worst performance on MSE and MRE among all the methods.

### 6.3 Evaluation of the burst shape prediction layer

For this prediction task, we compare our model with two different models commonly used for time series prediction. The metric mean square error(MSE) is used to evaluate the prediction performance. We perform the Affinity Propagation Clustering [5] on the dataset of reshaped bursts. Euclidean distance is applied to calculate the input similarity matrix of the training set, and we choose DTW distance as the measure of similarity when forecasting. Besides, when the start value of the forecasted result and the last value in the window differ greatly, we apply a decay function to smooth the 30% of the predicted burst series, namely  $y_{30\%}$ :  $y = \alpha f + (1 - \alpha)g$ . Here, we set  $f$  as the straight line from the last point in the window  $w$  to the last point of  $y_{30\%}$ , and set  $g$  as  $y_{30\%}$ .  $\alpha$  is set as the exponential function  $e^{(-1/3)t}$ .

**Table 4.** Performance comparison on the entire period of bursts (MSE score, partial data)

Category ID	LSTM	ARMA	Our model
0	9.48	8.39	<b>5.94</b>
1	10.45	8.32	<b>6.43</b>
2	1.35	0.97	<b>0.78</b>
3	40.15	32.70	<b>31.72</b>
4	15.62	<b>5.01</b>	14.39
5	32.66	5.61	<b>2.82</b>
6	0.83	0.61	<b>0.28</b>
7	13.91	8.48	<b>6.51</b>
8	21.67	2.64	<b>0.86</b>
9	14.10	7.55	<b>4.69</b>

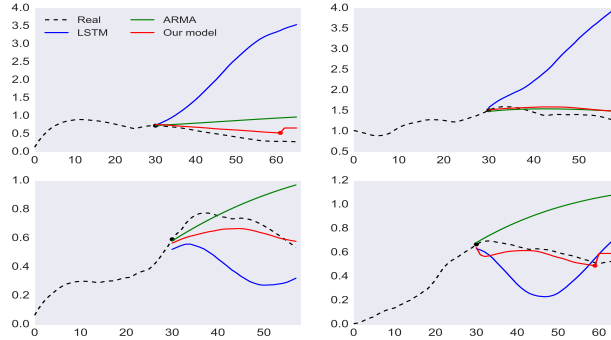


**Fig. 5.** Mean average burst series prediction MSE scores with standard deviation

Table 4 and Figure 5 shows the results of performance comparison. For the first two methods, we judge the performance on the real period of bursts. If the

predicted period of our model is less than the real one, we set the left part with the average value of our prediction. We also draw some of the predicted results into Figure 6. Conclusions can be drawn as follows:

- The LSTM model shows the worst performance on average MSE, scoring 20.58. Besides, the performance of LSTM are not that stable, with relatively high deviations of 37.09.
- Take all categories into account, our model wins 57 out of 64 times on the score of MSE. The average MSE of our method is 5.88, which significantly outperforms the other related methods, showing a relative improvement of 71% and 30%. The MSE standard deviations of our model and other methods are 37.10, 10.36 and 7.57 on average, indicating that our model makes a significant improvement by 80% and 29%.



**Fig. 6.** Samples of the predicted burst results

## 7 Conclusion

In this paper, we take the initiative to propose a burst prediction framework of online product sales. The framework includes three layers: a burst detection layer, a bursty feature prediction layer and a burst shape prediction layer. The burst detection layer detect the start of a burst with a sliding window and a optimized classification model. The three bursty features, the burst peak, period and off-burst value, are predicted by different regression model with the best training score. The entire burst shape is generated by the burst series with similar seasonal features in the same category. Extensive experiments are conducted on real datasets from JD.com. We find that in average our framework achieves 4% to 45% advantage of F1-score in the classification and up to 73% improvement of HitRate@50% on the feature prediction against other methods. The result shows that the proposed solutions are effective to the burst prediction task, with an average improvement of 71% and 30% on MSE. We expect our framework to be of great value in e-commerce field.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (No.61602487), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (No.2015030275).

## Reference

1. D Agrawal, RP Agrawal, JB Singh, and SP Tripathi. E-commerce: True indian picture. *Journal of Advances in IT*, 3(4):250–257, 2012.
2. Susan Avery. Online tool removes costs from process. *Purchasing*, 123(6):79–81, 1997.
3. Albert-László Barabási. *Bursts: the hidden patterns behind everything we do, from your e-mail to bloody crusades*. Penguin, 2010.
4. Lei Cao and Zhouping Zhang. *The 2015 annual China electronic commerce market data monitoring report*. China Electronic Commerce Research Center, Hangzhou, 2016.
5. Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.
6. Daniel Gruhl, Ramanathan Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 78–87. ACM, 2005.
7. Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
8. Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. Predicting bursts and popularity of hashtags in real-time. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 927–930. ACM, 2014.
9. Shoubin Kong, Qiaozhu Mei, Ling Feng, and Zhe Zhao. Real-time predicting bursting hashtags on twitter. In *International Conference on Web-Age Information Management*, pages 268–271. Springer, 2014.
10. Shoubin Kong, Qiaozhu Mei, Ling Feng, Zhe Zhao, and Fei Ye. On the real-time prediction problems of bursting hashtags in twitter. *arXiv preprint arXiv:1401.2018*, 2014.
11. Yu-Ru Lin, Drew Margolin, Brian Keegan, Andrea Baronchelli, and David Lazer. # bigbirds never die: understanding social dynamics of emergent hashtag. *arXiv preprint arXiv:1303.7144*, 2013.
12. Zongyang Ma, Aixin Sun, and Gao Cong. On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, 64(7):1399–1410, 2013.
13. Chotirat Ann Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 11–22. SIAM, 2004.
14. Michael Schaidnagel, Christian Abele, Fritz Laux, and Ilia Petrov. Sales prediction with parametrized time series analysis. *Proceedings DBKDA*, pages 166–173, 2013.
15. Frank M Thiesing and Oliver Vornberger. Sales forecasting using neural networks. In *Neural Networks, 1997., International Conference on*, volume 4, pages 2125–2128. IEEE, 1997.
16. Andreas S Weigend. Time series prediction: forecasting the future and understanding the past. *Santa Fe Institute Studies in the Sciences of Complexity*, 1994.