TS-Bert: Time series Anomaly Detection via Pre-training Model Bert

Weixia Dang[1,2], Biyu Zhou[1,2](✉), Lingwei Wei[1,2], Weigang Zhang[1,2], Ziang Yang[1,2], and Songlin Hu[1,2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
zhoubiyu@iie.ac.cn

Abstract. Anomaly detection of time series is of great importance in data mining research. Current state of the art suffer from scalability, over reliance on labels and high false positives. To this end, a novel framework, named TS-Bert, is proposed in this paper. TS-Bert is based on pre-training model Bert and consists of two phases, accordingly. In the pre-training phase, the model learns the behavior features of the time series from massive unlabeled data. In the fine-tuning phase, the model is fine-tuned based on the target dataset. Since the Bert model is not designed for the time series anomaly detection task, we have made some modifications thus to improve the detection accuracy. Furthermore, we have removed the dependency of the model on labeled data so that TS-Bert is unsupervised. Experiments on the public data set KPI and yahoo demonstrate that TS-Bert has significantly improved the f1 value compared to the current state-of-the-art unsupervised learning models.

Keywords: Anomaly Detection · Pre-training Model · Time Series Analysis.

1 INTRODUCTION

In large-scale distributed systems, monitoring is inevitable. Analyzing monitoring data is of great significance to ensure the quality of service of the system and protect corporations against malicious attacks. From the perspective of data analysis, this means that we need to perform real-time analysis on large volume of time series generated by monitoring systems in order to detect potential errors and anomalies.

However, anomaly detection on large volume of time series is not easy. Specifically, it has the following challenges:

Challenge 1: *Efficiency*. Due to the large scale of the problem, time series anomaly detection algorithms require rapid detection of anomalies. In industrial applications, time series anomaly detection systems need to process hundreds of millions of time series in real time.

Challenge 2: *Generalization*. The patterns of time series are diverse, it is time-consuming, laborious and unrealistic to build a model for each pattern of

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

time series. The generalization ability of industrial anomaly detection services is important to work well on various patterns of time series.

Challenge 3: *Lack of labels*. In industrial applications, obtaining large-scale, high-quality time series anomaly detection label data requires a lot of manpower and material resources. This means that algorithms that overly rely on labels are practically infeasible.

We argue that a good time series anomaly detection algorithm should be efficient, unsupervised, and can be adapted to most scenarios with some simple adjustments. Unfortunately, existing proposals [5, 8–10, 15, 17, 20, 23] fail to adequately meet all these requirements. According to our observations, feature extraction and long-distance dependent modeling have great influence on the accuracy of time series anomaly detection. However, the widely adopted feature extraction models for sequences such as CNN [22], RNN [4] and LSTM [12] do not deal well with long-distance dependencies. In this paper, we propose to model time series by referring to the Bert [7] model in natural language processing (NLP) field. The reasons are three folds: (1) The core algorithm of Bert is Transformer, which can solve the long-distance dependence issue by orchestrating self-attention modules. (2) Bert is a pre-training model [7,13], which can learn effectively from large-scale raw text to alleviate the dependence on supervised learning during the pre-training phase. This pattern can be borrowed to time series abnormal detection to solve the problem of missing tags. Besides, the pattern of pre-training is essentially transfer learning. By fine-tuning with target datasets, it can be applied to many scenarios, thus is more generic. (3) Bert is a thorough open source framework which can be used directly, thus saving development costs.

Based on the above analysis, we propose a pre-training model TS-Bert based on the Bert model in NLP to solve the time series anomaly detection problem in this paper. Technically, TS-Bert involves two phases: pre-training and fine-tuning. In the pre-training phase, the model learns the behavior features of the time series from a large amount of unlabeled data. Note that the pre-training task includes masked LM and next sentence prediction. In the fine-tuning phase, the parameters of the pre-training model is tuned according to the specific time series anomaly detection task. Since the Bert model is not designed for this task, we have made some modifications to the model to improve the detection accuracy. It is worth noting that although the original Bert model is supervised, TS-Bert is an unsupervised model. By using SR [15] method to generate labels, we remove the dependency of the model on labeled data. Moreover, the model can perform better when partial labeled data is provided.

The major contributions are highlighted as below:

- For the first time, we introduce the pattern of pre-training and fine-tuning to the field of time series anomaly detection.
- We propose to adopt the Bert model in NLP field to model time series thus can address the long-distance dependent modeling issue. Accordingly, we solve the problems of mapping the Bert model to the time series anomaly detection task, such as large knowledge span, mismatched input format, etc.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

– We conducted extensive experiments to verify the performance of TS-Bert. Simulation results on two widely used public datasets demonstrate that our method is 21% and 11% more accurate on KPI dataset and yahoo dataset than the state-of-the-art solution.

2 Related Work

This section summarizes the work closely relevant to this paper.

The traditional statistical model [5,11,16] such as ARIMA [23] and Holt-winter [5], typically utilize handcrafted features to model normal/anomaly patterns. These methods are only applicable to time series data that conform to certain characteristics, with poor universality.

Supervised anomaly detection [8–10,17] using labeled data to train anomaly detection classifiers. With the rapid development of machine learning and data mining, many time series anomaly detection methods based on supervised learning models [8–10, 17] have been proposed. However, obtaining large-scale, high-quality time series anomaly detection label data requires a lot of manpower and material resources. This means that anomaly detection algorithms based on supervised learning are not feasible in large-scale industrial applications.

Unsupervised anomaly detection does not require labels during the training process which alleviates the dependence of supervised learning. In recent years, many unsupervised time series anomaly detection algorithms have been proposed. VAE [20] is a reconstruction models, which learns the representation for normal time series by reconstructing the original input based some latent variables. SR [15] combines the benefits of Spectral Residual and convolutional neural network to detect anomalies in time series. Current state of the art anomaly detection approaches based on unsupervised models suffer from scalability and a large number of false positive compromising user experience.

Pre-training model has received widespread attention in recent years [7, 13]. It designs pre-tasks to learn rich semantic representations from a large amount of unlabeled data and the learned data representations can be transferred to downstream tasks. In this way, the problem of label data shortage can be solved. BERT is a pre-training learning model that obtains the the state-of-the-art results in various natural language processing tasks [6, 18, 19, 21]. As far as we know, TS-Bert is the first pre-trained model for the field of time series anomaly detection.

3 METHODOLOGY

Time series anomaly detection (TSAD) is to check whether the current data has obviously deviated from the normal situation through historical data analysis. The TSAD problem can be defined as follows.

Problem Definition: The input of TSAD is denoted by $T \in R^n$*, where* n *is the length of timestamps. The task of TSAD is to produce an output vector* $Y \in R^n$*, where* $y_i \in \{0, 1\}$*,* 0 *means normal, and* 1 *means abnormal.*

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

We propose TS-Bert to address the TSAD problem. The framework of TS-Bert is shown in Figure 1. To improve the robustness of our model, all the time series datasets are normalized before use. In the pre-training phase, the model will learn the behavior features of the time series from massive unlabeled data. After that the parameters of the pre-trained model will be fine-tuned according to the target dataset. Since the Bert model is not designed for the TSAD task, we have made some modifications to the model thus to improve the detection accuracy. Furthermore, we removed the dependency of the model on labeled data so that TS-Bert is unsupervised. In the following, we will describe TS-Bert in detail. The notations used is summarized on Table 1.

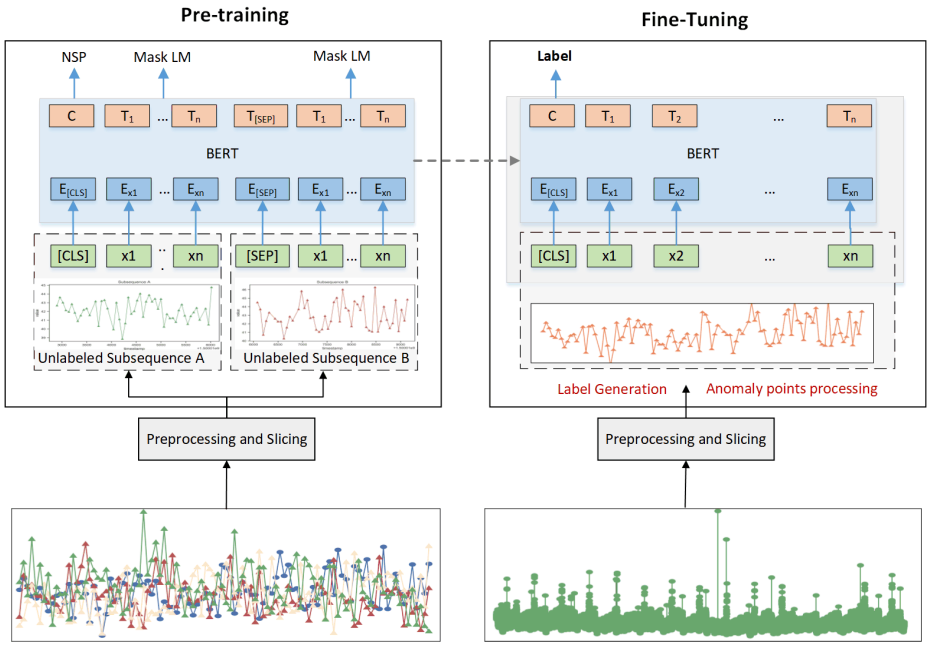

Fig. 1: The framework of TS-Bert.

3.1 Data Preprocessing

In our method, all the input time series are normalized with the maximum and minimum values. In order to map time series to the input format of the Bert model, we expand the normalized time series by *scale* times. The *scale* size is the normalization multiple which approximately equals to the dictionary size used by the Bert model. The formula of data preprocessing stage is as follows.

$$\widetilde{T} = \frac{T - min(T)}{max(T) - min(T)} \times scale \tag{1}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

Table 1: Notations

Symbol	Description
T	a time series
n	the length of timestamps
Y	the output vector of an anomaly detection algorithm
$scale$	the normalization multiple
$\widetilde{T}$	the time series after preprocessing
p_w	the sequence length in the pre-training phase
h_w	the length of the historical information
s_w	the length of the sliding window when anomaly timestamp is replaced
H	the hidden size of Bert
C	the final hidden vector of Bert
θ	the threshold to generate anomaly detection results using SR
δ	the delay time

where $min(T)$ and $max(T)$ are the maximum and minimum value of the time series T respectively.

3.2 Pre-training

The task of the pre-training phase is to learn the behavioral features of time series from massive datasets. Our pre-training task follows the pre-training task of the Bert model, which includes Masked LM and Next Sentence Prediction (NSP). We masks some positions of the input sequence at random, and then predicts the values of those mask positions. Through Masked LM, we obtain a bidirectional pre-trained model. Next Sentence Prediction is to learn the relationships of different time series by judging whether two time series fragments are adjacent. We obtain pre-training data from a large volume of time series, and assuming that there exists no anomaly in the data.

Given k entries of time series $X = (x_1, x_2, ...x_k)$, where x_i is i-th time series with length L_i. We aim to get pre-training data $D = (d_1, d_2, ...d_k)$. The details of pre-training are as follows (also shown in Algorithm 1).

Step 1: For each time series x_i, we preprocess it through the method described in Section 3.1. The time series after preprocessing is denoted by $\widetilde{x_i}$.

Step 2: We slice $\widetilde{x_i}$ into subsequences d_i with length p_w. Obviously, $d_i = (s_1, s_2, ...s_m)$ and $m = \lceil \frac{L_i}{p_w} \rceil$. Mapped to the NLP field, d_i is equivalent to a document, and s_i is a sentence in the document.

Step 3: Next, we build a dictionary of time series. Time series dictionary V = (0, 1, ..., scale, [CLS], [SEP]). [CLS] is used as the aggregate sequence representation of the classification task. Bert learns the relationship between sequences by packing two sequences into a single sequence, and [SEP] is used to separate the two sequences.

Step 4: Then, we map s_i to a format that the Bert model can understand. We round the timestamp values of s_i and map them to the dictionary to get the encoding in the dictionary, named input_ids.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

Algorithm 1 Pre-training the Bert model using the time series data set.

Input: $X = (x_1, x_2, ...x_k)$, where x_i is i-th time series with length L_i; p_w: the subsequence length. scale: the normalization multiple.
Output: Pre-trained Bert model.

for i in range$(1, k)$ **do**
  $\widetilde{x_i} = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \times scale$
  for j in range$(0, \lceil \frac{L_i}{p_w} \rceil)$ **do**
    $s_j = x_i[j \times p_w, ..., (j+1) \times p_w]$
    $d_i.append(s_j)$
  end for
  $D.append(d_i)$
end for
Build time series dictionary V = (0, 1, ..., scale, [CLS], [SEP]).
Map D to a format that the Bert model can understand.
Pre-train Bert according to the pre-training tasks.
Save the parameters of the pre-trained model.

Step 5: Finally, we pre-train Bert according to the pre-training tasks Masked LM and NSP to obtain a pre-trained model.

3.3 Fine-tuning

The purpose of fine-tuning is to fine-tune the parameters of the pre-trained model according to the target dataset. We plug the inputs of TSAD task and outputs into Bert and fine-tune all the parameters end-to-end. Given a time series T with n timestamps, the detailed process of fine-tuning is shown below (also illustrated in algorithm 2).

Data preprocessing. We preprocess T through the method described in Section 3.1. The time series after preprocessing is named $\widetilde{T}$.

Label generation. The fine-tuning of the Bert model requires labels. However, obtaining large-scale, high-quality time series anomaly detection label data requires a lot of manpower and material resources. To alleviate this problem, we use the state-of-the-art anomaly detection method, Spectral Residual(SR) [15] to generate the labels in the training data on fine-tuning stage. Through SR, we get the label vector $Y \in R^n$, where $y_i \in \{0, 1\}$. SR is a light-weighted method which adds little overhead to the entire model.

Time series slicing and cleaning. We detect anomaly events at the timestamp t based on S_i, which contains the historical information and the value of the time series at t. The length of the historical information we take is h_w. The prerequisite of high accuracy of the model is that there are no abnormal points in the historical information. However, in practice, historical information is likely to contain abnormal points. Therefore, we need to clean these abnormal points. Specifically, the anomaly timestamps are replaced with the mean value of the sliding window s_w on the left of the sequence. The length of s_w cannot exceed h_w. The calculation of S_i is shown in the following formula.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

Algorithm 2 Fine-tune based on time series anomaly detection tasks.

Input: T: time series; n: the length of time series; h_w: the history window size; s_w: the slice window size;

Output: The Bert model after fine tuning.

Build time series dictionary V = (0, 1, ..., scale, [CLS], [SEP]).
for i in range(0, n) **do**
  $\widetilde{T}_i = \frac{T_i - min(T)}{max(T) - min(T)} \times scale$
  $Y_i = SR(\widetilde{T}_i)$
  if $Y_i == 0$ **then**
    $T_i^* = \widetilde{T}_i$
  else
    $T_i^* = \frac{\sum_{j=0}^{s_w-1} \widetilde{T}_{i-s_w+j}}{s_w}$
  end if
  if $i \leq h_w$ **then**
    $S_i = [T^*_{i-h_w+1}, T^*_{i-h_w+2}, \cdots, T_i]$
  else
    $S_i = [T_0, \cdots, T_1^*, \cdots, T_i]$
  end if
  Map S_i to a format that the Bert model can understand.
end for
Add additional output layer to the Bert.
Fine tune the model according the feature.
return The Bert model after fine tuning.

$$S_i = \begin{cases} [T^*_{i-h_w+1}, T^*_{i-h_w+2}, \cdots, T_i] & i > h\_w \\ [T_0, \cdots, T_1^*, \cdots, T_i] & i \leq h\_w \end{cases} \tag{2}$$

$$T_j^* = \begin{cases} T_j & j \text{ is nomal} \\ \frac{\sum_0^{s_w-1} T_{j-s_w+i}}{s_w} & j \text{ is anomal} \end{cases} \tag{3}$$

Input representation mapping. First, we build a time series dictionary, and then map $S[i]$ to the dictionary, named input_ids. The creation of the dictionary and mapping method to dictionary is the same as the pre-training stage. Then, we get the input representation of Bert according input_ids.

The addition of a decoder. To fine-tune Bert on time series anomaly detection dataset, we use the final hidden vector $C \in R^H$(H is the hidden size of Bert) as the aggregate representation. The new parameters introduced during fine-tuning are classification layer weights $W \in R^{2\times H}$. We compute a standard classification loss with C and W, i.e., $log(softmax(CW^T))$.

4 Evaluation

In this section, we conduct simulations to verify the effectiveness of TS-Bert. Firstly, the evaluation settings are described. Then the evaluation results on public datasets are provided.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

4.1 Settings

Datasets. Two public datasets are adopted in the evaluations, namely KPI dataset [1] and Yahoo dataset [2]. The detailed statistics and settings of these two datasets are shown in Table 2. Figure 2 and Figure 3 illustrate a sequence in the two datasets, respectively.

Table 2: Statistics of datasets

Dataset	Curves	Training set size	Testing set size	Anomaly Rate
KPI	58	3004066	2918847	134114/2.26%
Yahoo	367	286483	286483	3896/0.68%

KPI dataset collects KPI(Key Performance Indicators) data from many Internet companies after desensitization. These KPI can be roughly divided into two types: service KPI and machine KPI. Service KPI is a performance indicators reflecting the scale and quality of Web services. Machine KPI refers to the performance indicators that can reflect the health status of the machine.

Yahoo dataset is an open dataset for anomaly detection released by Yahoo lab. Part of the time series is synthetic (i.e., simulated); while the other part comes from the real traffic of Yahoo services. The anomaly points in the simulated curves are algorithmically generated and those in the real-traffic curves are labeled by editors manually. A time series from the real traffic of Yahoo services is shown in Figure3. Since the Yahoo data set does not divide the training set and the test set. We randomly divide the time series of the Yahoo dataset into two halves, the first half is utilized for fine-tuning the pre-trained model Bert while the second half is leveraged for evaluation.

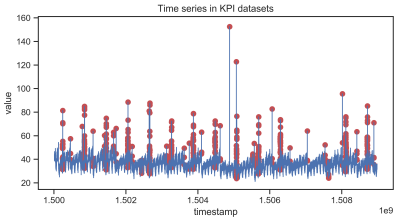

Fig. 2: A user's time series in KPI data set.

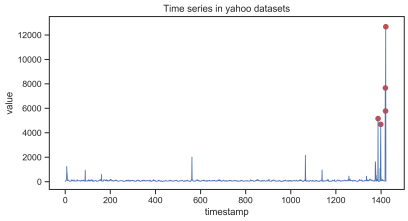

Fig. 3: A time series in Yahoo data set

Benchmarks. We compare TS-Bert with state-of-the-art models for TSAD, including SR [15], SR+CNN [15], FFT [14], DONUT [20]. In view of the fact that the training process of SR+CNN requires normal data provided by Microsoft, which cannot be obtained, we directly quote the result of [15].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

Metrics. The usual metrics used in the TSAD problem are precision, recall and F1-score. In this paper, we also use this set of metrics to indicate the performance of our model. For ease of reference, we use Metrics-1 to represent it.

In practice, anomalous observations usually form contiguous segments since they occur in a contiguous manner. In this case, the authors in [1] propose to use a "delayed" manner to calculate TP(true positive), TN(true negative), FP(false positive), FN(false negative). Assume that the delay is δ, the calculation rules of TP, TN, FP and FN are as follows.

- *For a marked continuous abnormal segments:* If there is an anomaly point detected correctly and its timestamp is at most δ steps after the first anomaly of the segment, it is considered that the anomaly detection algorithm has successfully detected the whole continuous abnormal interval, so each abnormal point in the abnormal segment is counted as TP; otherwise, each abnormal point in the continuous abnormal segment is counted as FN.
- *For time points without abnormal markings:* If the anomaly detection algorithm outputs an anomaly, it is counted as FP; otherwise, it is counted as TN.

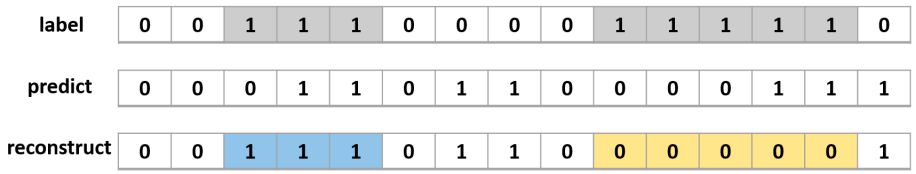

Fig. 4: Illustration of the evaluation protocol. There are 15 contiguous points in the time series, where the first row indicates the labels list of time series; the second row shows the anomaly detection results; and the third row shows adjusted results according to the evaluation protocol.

When $\delta = 2$, the illustration for the evaluation protocol are shown in Figure 4. There are two abnormal segments in the original time series. When $\delta = 2$, the anomaly detection algorithm successfully detected the first continuous abnormal segment, but failed to detect the second continuous abnormal segment.

This set of metrics is also adopted in this paper to indicate the performance of our model. For ease of reference, we use Metrics-2 to represent it.

Parameter settings. The frame structure of the Bert model we used is consistent with the uncased_L12_H-768_A12($Bert^{Base}$) released by Google. It contains 12-layer, 768-hidden, 12-heads with 110M parameters for the encoder. In our model, we set *scale* as 30000 which is approximately equal to $Bert^{Base}$ dictionary size. We set the sequence length p_w in the pre-training phase as 50. p_w is equal to half of the maximum sequence length in the fine-tuning phase, because

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

the pre-training phase packs them into a sequence when learning the relationship between the two sequences. The history length h_w is set to 100, which limits the size of historical data. The length of the sliding window when anomaly timestamp is replaced s_w is set as 1 in KPI dataset, 5 in Yahoo dataset through a grid search.

We obtain pre-training data from a large volume of time series, and assuming that there exists no anomaly in the data. The pre-training data set is selected from the normal segments of the KPI and Yahoo datasets, and Cluster-trace-v2018 [3] of provided by the Alibaba Open Cluster Trace Program. Cluster-trace-v2018 includes about 4000 machines in a periods of 8 days. There are both online services (aka long running applications, LRA) and batch workloads colocated in every machine in the cluster. Over the past year, the scale of colocation of online services and batch workloads have greatly increased resulting in the improvement of the overall resource utilization.

The pre-training data contains 4530 time series. According to p_w, the time series are slicing into 320838 subsequences with length 50 which contain 16041900 timestamps totally. The pre-training tasks are consistent with the original Bert model, which include Masked LM and Next Sentence Prediction (NSP). The hyperparameter settings for pre-training are shown in Table 2. After pre-training, the pre-training dataset on the Masked LM and NSP tasks has an accuracy of 0.69 and 0.99 respectively. Through pre-training, the Bert model learns the behavioral characteristics of time series from massive data.

The fine-tuning of the Bert model requires labels. To alleviate this problem, we use the state-of-the-art anomaly detection method, Spectral Residual(SR) [15] to generate the labels in the training data on fine-tuning stage. SR is a light-weighted method which adds little overhead to the entire model. Following [15], we set the threshold θ as 3 to generate anomaly detection results using SR. The hyper-parameter max_seq_length in the fine-tuning stage is the same as the pre-training stage, and the other hyper-parameters as $Bert_{Base}$.

Table 3: Hyperameter settings for pre-training

Parameter Name	Description	Values
max_seq_length	the maximum total input sequence length	100
max_predictions_per_seq	the maximum number of masked LM per sequence	20
train_batch_size	total batch size for training	32
eval_batch_size	Total batch size for eval	8
learning_rate	the initial learning rate for Adam	5e-5
num_warmup_steps	the number of warmup steps	10000

4.2 Results

Comparison with SOTAs. For TS-Bert, we set hyper-parameters as Section 4.3 described. As mentioned, anomalies usually occur in a continuous segment,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

thus it is also reasonable to use Metric-2 to evaluate the model's performance. We report *Precision*, *Recall* and F_1*-score* under Metric-2 separately for each dataset. We set the delay time $\delta = 7$ for KPI dataset, $\delta = 3$ for Yahoo dataset. As shown in Table 4, TS-Bert shows excellent generalization capability and achieves the best F1-scores consistently on two public datasets. Specifically, we achieve 21% and 11% improvement over the best state-of-the-art performance on KPI dataset and Yahoo dataset. Besides, we find that the detection time of TS-Bert is milliseconds, which can meet the needs of most online applications.

Table 4: Results on unsupervised model.

	KPI			yahoo		
Model	Precision	Recall	F1-score	Precision	Recall	F1-score
FFT	0.382	0.534	0.445	0.599	0.301	0.400
DONUT	0.371	0.326	0.347	0.013	0.825	0.026
SR	0.782	0.675	0.725	0.535	0.780	0.634
SR+CNN	0.797	0.747	0.771	0.816	0.542	0.652
TS-Bert	0.897	0.976	**0.935**	0.801	0.664	**0.726**

Intuitively, using the labels generated by the SR [15] method and training the model, the accuracy of the model is at most infinitely close to the SR method. However, before fine-tuning, we used large-scale time series data to learn the semantic information within and between segments of the normal time series. Fine-tuning only uses a small amount of data set to fine-tune the model parameters after pre-training. If the knowledge learned in the pre-training stage is sufficient, the model is sufficient to learn normal time series features. Therefore, fine-tuning on a pre-trained model using large-scale data, the anomaly detection results obtained is better than the label generation method used in the fine-tuning process.

In the previous experiments, we can see that the TS-Bert model shows convincing results in the case of unsupervised anomaly detection. In addition, we can obtain more satisfactory results when the exception label is available. For example, using only half of the training set labels to fine-tune Bert on the KPI data set, the F_1 scores reached 0.938 on the test dataset.

Impact of indicator metrics. In practice, anomalies usually occur in a continuous segment thus it is also reasonable to use Metric-2 to evaluate the model's performance. But we require a model to detect anomalies as soon as possible to take quick actions, if the delay of Metric-2 is too long, it will damage the user experience.

In this subsection, we compare TS-Bert with the state-of-the-art anomaly detection method SR under Metric-2 with different delay values. Figure 5 illustrates the F_1 scores of TS-Bert and SR for different delay value of δ on two datasets. Notice that the F_1 scores becomes larger as the delay δ increases. Over-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

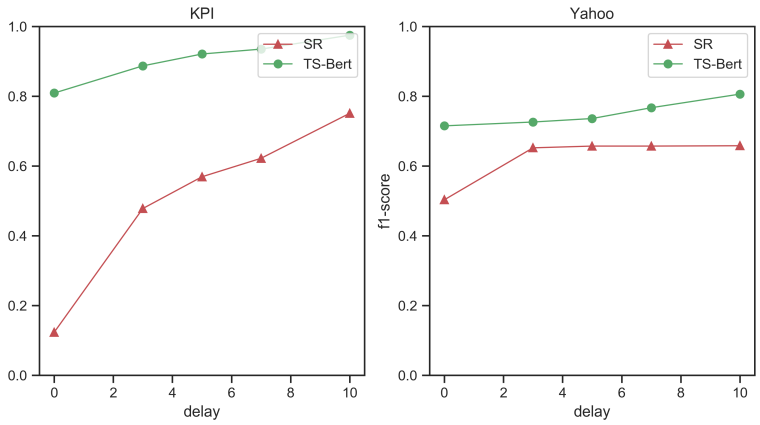

Fig. 5: Comparison with different delay value.

all, our model TS-Bert achieves better performance, even when the acceptable delay is very small.

Impact of s_w. We also analyze the influence of s_w to the performance of TS-Bert. We use the historical information and the current value to detect anomaly events, which is sensitive to irregular and abnormal instance in the history data. To alleviate this problem, we replace those anomaly timestamps in the history data with the mean value of the sliding window s_w on the left of the sequence. The length of s_w cannot exceed h_w.

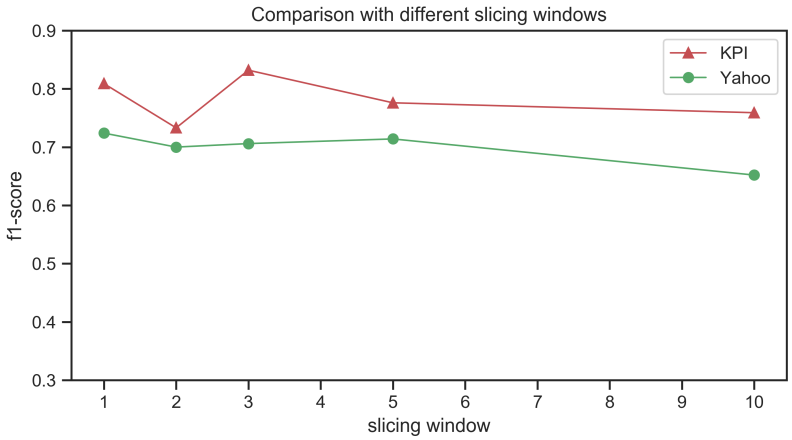

Fig. 6: Comparison with different slicing windows.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

In order to exclude the impact of evaluation indicators on the results, we use the Metric-1 as the indicator. Figure 6 shown the F_1 scores of TS-Bert for different s_w metrics on two dataset. We notice that different settings of s_w achieve similar performer on two datasets. That is, when the value of s_w is relatively small, the mean value of the sliding window is closer to the normal value of the point. When s_w is relatively large, the mean value of the sliding window is quite different from the normal value of the point. For example, if s_w is set as 100, the F_1 scores only 0.31 on KPI dataset.

5 Conclusion

In this paper, we introduce the pattern of pre-training and fine-tuning to the field of TSAD and propose a novel framework TS-Bert based on Bert model in NLP to solve the TSAD problem. TS-Bert includes two phases: pre-training and fine-tuning. Through pre-training on massive time series datasets, the TS-Bert model learns the behavioral characteristics of time series very well. We have made some modifications thus to improve the detection accuracy. In addition, to alleviate the problem of missing labeled data, we use the SR method to generate labels on the fine-tuning phase. Evaluation results show that TS-Bert outperforms the state-of-the-art solution on two public datasets. Specifically, we achieve 21% and 11% accuracy increase on KPI dataset and yahoo dataset, respectively.

Acknowledgements

Supported by the National Key Research and Development Program of China 2017YFB1010001

References

1. http://iops.ai/dataset_detail/?id=10/
2. https://yahooresearch.tumblr.com/post/114590420346/a-benchmark-dataset-for-time-series-anomaly
3. https://github.com/alibaba/clusterdata/blob/v2018/cluster-trace-v2018/trace_2018.md
4. Canizo, M., Triguero, I., Conde, A., Onieva, E.: Multi-head cnn–rnn for multi-time series anomaly detection: An industrial case study. Neurocomputing **363**, 246–260 (2019)
5. Chatfield, C.: The holt-winters forecasting procedure. Journal of the Royal Statistical Society: Series C (Applied Statistics) **27**(3), 264–279 (1978)
6. Chen, Q., Zhuo, Z., Wang, W.: Bert for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909 (2019)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Görnitz, N., Kloft, M., Rieck, K., Brefeld, U.: Toward supervised anomaly detection. Journal of Artificial Intelligence Research **46**, 235–262 (2013)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17

9. Laptev, N., Amizadeh, S., Flint, I.: Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1939–1947 (2015)
10. Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Jing, X., Feng, M.: Opprentice: Towards practical and automatic anomaly detection through machine learning. In: Proceedings of the 2015 Internet Measurement Conference. pp. 211–224 (2015)
11. Lu, W., Ghorbani, A.A.: Network anomaly detection based on wavelet analysis. EURASIP Journal on Advances in Signal Processing **2009**, 1–16 (2008)
12. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016)
13. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
14. Rasheed, F., Peng, P., Alhajj, R., Rokne, J.: Fourier transform based spatial outlier mining. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 317–324. Springer (2009)
15. Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., Zhang, Q.: Time-series anomaly detection service at microsoft. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3009–3017 (2019)
16. Said, S.E., Dickey, D.A.: Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika **71**(3), 599–607 (1984)
17. Shipmon, D., Gurevitch, J., Piselli, P.M., Edwards, S.: Time series anomaly detection: Detection of anomalous drops with limited features and sparse examples in noisy periodic data (2017)
18. Vig, J., Ramea, K.: Comparison of transfer-learning approaches for response selection in multi-turn conversations. In: Workshop on DSTC7 (2019)
19. Wu, X., Lv, S., Zang, L., Han, J., Hu, S.: Conditional bert contextual augmentation. In: International Conference on Computational Science. pp. 84–95. Springer (2019)
20. Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al.: Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 World Wide Web Conference. pp. 187–196 (2018)
21. Yang, W., Zhang, H., Lin, J.: Simple applications of bert for ad hoc document retrieval. arXiv preprint arXiv:1903.10972 (2019)
22. Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., Chawla, N.V.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1409–1416 (2019)
23. Zhang, Y., Ge, Z., Greenberg, A., Roughan, M.: Network anomography. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. pp. 30–30 (2005)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_17