Sequential VAE-LSTM for Anomaly Detection on Time Series

Run-Qing Chen, Guang-Hui Shi, Wan-Lei Zhao, Chang-Hui Liang

Abstract—In order to support stable web-based applications and services, anomalies on the IT performance status have to be detected timely. Moreover, the performance trend across the time series should be predicted. In this paper, we propose SeqVL (Sequential VAE-LSTM), a neural network model based on both VAE (Variational Auto-Encoder) and LSTM (Long Short-Term Memory). This work is the first attempt to integrate unsupervised anomaly detection and trend prediction under one framework. Moreover, this model performs considerably better on detection and prediction than VAE and LSTM work alone. On unsupervised anomaly detection, SeqVL achieves competitive experimental results compared with other state-of-the-art methods on public datasets. On trend prediction, SeqVL outperforms several classic time series prediction models in the experiments of the public dataset.

Index Terms—Time series, Unsupervised anomaly detection, Robust Trend prediction.

1 Introduction

Due to the steady growth of cloud computing and the wide spread of various web services, a big volume of IT operation data are generated on the daily basis. IT operations analytics are introduced to discover patterns from these huge amounts of time series data. The primary goal of operations analytics is to automate or monitor IT systems based on the operation data via artificial intelligence. It is widely known as artificial intelligence for IT operations (AIOps). Two fundamental tasks in AIOps are trend prediction and anomaly detection on the key performance indicators (KPIs), such as the time series about the number of user accesses and memory usage, etc.

In general, a sequence of KPIs is given as a univariate time series $X = \{x_1, \cdots, x_t, x_{t+1}, \cdots, x_{n-1}, x_n\}$, where the subscript represents the time stamp and x_t is the real-valued status at one time stamp. The trend prediction is to estimate status x_{t+1} given the status from *1* to t are known. While anomaly detection is to judge the status on time stamp t is abnormal given the status across all t time stamps are known. In practice, these two tasks are expected to work jointly to undertake automatic performance monitoring on the KPIs. Most of the KPIs are the reflections of the user behaviors, habits, and schedule [1]. Since these events are largely repeated periodically, the KPI sequences are mostly stationary and periodic on the daily or weekly basis. Therefore they are believed predictable though the latent factors that impact the status are hard to be completely revealed.

Although KPIs largely exhibit regular patterns, they are mixed with noises. As a result, performing anomaly detection and trend prediction on these time series are non-trivial in practice. First of all, it is unrealistic to expect a large number of labeled data available to train anomaly detection models. On the one hand, KPIs are in big amount while the abnormal status are relatively in a rare occurrence. On the other hand, both the regular patterns of KPIs and the appearance of abnormal status drift as time goes on. Annotating big amount of training data therefore would require painstaking efforts and is error-prone. As a result, unsupervised anomaly detection is preferred. Moreover, due to the presence of anomalies in the KPIs, the popular prediction models such as long short-term memories (LSTMs) [2], [3] which perform well in the ideal environment fail to return decent results. For this reason, the robustness of the prediction model is highly valued.

There is a rich body of works for both unsupervised anomaly detection and robust trend prediction in the recent literature. Due to the great success of deep learning in many tasks, it has been recently introduced into both unsupervised anomaly detection and robust trend prediction. In the state-of-the-art works, these two tasks are addressed separately. Generative models such as variational auto-encoders (VAEs) [4] are adopted in anomaly detection. The time series are sliced into windows of equal size and time status within each window are encoded [1], [5]. Abnormal status is identified as it is far apart from the decoded normal status. Encouraging results are achieved from these approaches [1], [5]. Unfortunately, the performance turns out to be unstable as it ignores the temporal relationship between the sliding windows during the encoding. Typically, LSTM is adopted [2], [3] in the trend prediction. The advantage is that LSTM is able to capture the latent correlations between long term and short term status along the time series, even such correlation is non-linear. Due to its high model complexity, it is unfortunately sensitive to anomalies and noises. The problem is alleviated by ensemble learning [6], [7], nevertheless several folds of computational overhead become inevitable.

Different from existing solutions, a joint model called Sequential VAE-LSTM (SeqVL) is proposed in this paper. In

- *Run-Qing Chen, Chang-Hui Liang and Wan-Lei Zhao are with Xiamen University, Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Fujian, China. E-mail: wlzhao@xmu.edu.cn*
- *Guang-Hui Shi is with Bonree Inc., Beijing, China*

our solution, VAE and LSTM are integrated as a whole to address both unsupervised anomaly detection and robust trend prediction. The advantages of such framework are at least two folds.

- Firstly, VAE is adopted for unsupervised anomaly detection. LSTM block in SeqVL propagates the sequential patterns latent across neighboring windows to the VAE block during the training. The temporal relationships between the windows, which have been missed in existing VAE-based detection approaches, are therefore supplied to the VAE block.
- Secondly, LSTM is adopted for trend prediction. LSTM takes the re-encoded time series from the output of the anomaly detection (VAE block). Such design considerably reduces the impact of abnormal data and noises on the trend prediction block.

As a result, the prediction block (LSTM) makes use of the clean input from VAE. Meanwhile, the detection block (VAE) is trained with time series segments across which the sequential order is maintained by LSTM. This leads to considerably better performance than using LSTM and VAE alone for either anomaly detection or trend prediction on IT operations.

The remainder of this paper is organized as follows. Related works about unsupervised anomaly detection and trend prediction are presented in Section 2. The proposed model, namely SeqVL is presented in Section 3. The effectiveness of our approach both for trend prediction and anomaly detection is studied on two datasets in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Trend Prediction on Time Series

Trend prediction on the time series is an old topic as well as a new subject. On the one hand, it is an old topic in the sense it could be traced back to nearly *100* years ago [8]. In such a long period, classic approaches such as ARIMA [9], [10], Kalman Filter [11] and Holt-Winters [12] were proposed one after another. The implementations about these classic algorithms are found from recent packages such as Prophet [13] and hawkular [14]. Although efficient, the underlying patterns usually underfit due to their low model complexity. On the other hand, this is a new issue in the sense the steady growth of the big volume of IT operation data, which are mixed noises and anomalies, impose new challenges to this century-old issue.

Recently LSTMs is adopted for trend prediction for its superior capability of capturing long-term patterns on temporal data [6], [7]. In [6], [7], multiple prediction models are trained from one time series, the prediction is made by ensembling multiple predictions into one. The LSTMs model is also modified to performing online trend prediction in [15]. It allows the learnt model to be adaptive to emerging patterns of time series by balancing the weights between the come-in status and historical status.

2.2 Unsupervised Anomaly Detection on Time Series

IT operations data are in big amount and the anomalies are present in different patterns from one time series to another. It is therefore infeasible to train the detection model in a supervised manner. For this reason, the research focus in the literature is on unsupervised anomaly detection.

The first category of approaches is built upon the trend prediction. Specifically, when the status is far apart from the predicted status value at one time stamp, it is considered as an anomaly. In [16], ARIMA is adopted for trend prediction, then the detection is made based on the predicted status. However, due to poor prediction performance from ARIMA, precise anomaly detection is not achievable. Recently, due to its good capability in capturing patterns from time series with lags of unknown duration, a stacked LSTM [17] is proposed to perform anomaly detection. However, the uncertainty of the prediction model itself is overlooked in the approach. To address this issue, Research from Uber introduced Bayesian networks into LSTM auto-encoder. MC dropout is adopted to estimate the prediction uncertainty of the LSTM auto-encoder [18], [19]. In addition to the uncertainty of the prediction model, historical prediction errors is considered in a recent approach from NASA [20]. Nevertheless, all these detectors rely on largely the performance of the trend prediction. Inferior performance is observed when the time series show drifting patterns.

Another type of detection approaches divides the time series into a series of segments via a sliding window. Then conventional outlier detection approaches such as one-class SVM [21], [22] or SR [23] are adopted for anomaly detection within each window. Considering the patterns from both the regular status and the anomalies drift as time goes on, iForest [24] and robust random cut forest (RRCF) [25] are proposed. The latter reduces detection false positives considerably. Recently SPOT and DSPOT are proposed [26] to distinguish the anomalies from regular patterns with an adaptive threshold based on Extreme Value Theory [27]. As the anomalies are under different distribution from normal status, VAE is adopted to encode the regular patterns in each window [1]. The performance of VAE based approach is further boosted with an adversarial training [5].

In the above detection approaches, the temporal relationship between the windows is ignored. In the training, segments are treated independently and are shuffled randomly. Different from [1], [5], our joint model SeqVL maintains the sequential order between segments due to the incorporation of LSTM block.

3 The Proposed Model

In this section, a framework that integrates LSTM and VAE for both trend prediction and anomaly detection is presented.

3.1 Data Preprocessing

Given a raw operations time sequence $R = \{r_1, \cdots, r_t, \cdots, r_n\}$, it is not unusual that the operation status are missing on some time stamps due to sudden server down or network crash. Conventionally, the missing values are either simply filled with zeros or with average/median of all the status. Another common practice is to perform the linear interpolation with adjacent status. These schemes are effective when the missing status are

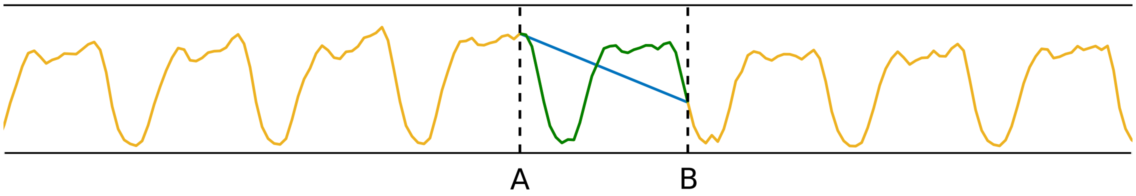


Fig. 1: Fill missing status with adjacent periods. In the time series, the status between time stamp A and time stamp B are missing. The blue line is recovered by the linear interpolation with the status from A and B. In contrast, the green curve is recovered by the first-order linear interpolation with the same time slot from the adjacent periods.

very few and sporadic. While when the missing status last for one or several periods, these schemes are no longer effective. They lead to considerable performance degradation for either prediction or detection since such scheme changes the original distribution of the time series.

As a consequence, a new way is adopted to fill the missing status before the time series are fed to the training and prediction. In our approach, the missing status are filled with adjacent periods. Specifically, when the duration of missing status is less than or equal to M units of time stamp, where M is a constant, the first-order linear interpolation is performed with the adjacent points (status ahead and after) to fill the missing status. When the duration of the missing data is greater than M units of time, the linear interpolation is performed with the status of the same time slot from the adjacent periods. This scheme is shown in Fig. 1. If the status of the same time slot from the adjacent periods is still missing. Our approach goes to the period ahead/after again until the status of the same time slot is available. In our implementation, we choose one day as the period, which is safe as IT operations are largely relevant to human activities. Constant M is set to *3* and *7* respectively for hour-level and minute-level datasets.

After filling the missing status, each time series is undergone z-score normalization with Eqn. 1.

$$x_t = \frac{r_t - \bar{\mu}}{\bar{\sigma}}, \tag{1}$$

where $\bar{\mu}$ is the mean and $\bar{\sigma}$ is the standard deviation. In our implementation, these two parameters are estimated on the training set.

As each time series is normalized, it is cut into segments with a sliding window. The width of the sliding window is w_0, which is introduced as a hyper-parameter. Similar to existing works, the step size of the window sliding is fixed to *1*. The status in one segment is given as $\mathbf{x}_t = \{x_{t-w_0+1}, \cdots, x_t\}$. After segmenting time series with sliding window, the time series is decomposed into a collection of segments viz., $S = \{\mathbf{x}_{w_0}, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_n\}$. Given t is the current time stamp, the status at time stamp $t+1$ is x_{t+1}. For any trend prediction approach, x_{t+1} is to be predicted based on previous status from segments $\mathbf{x}_{w_0}, \cdots, \mathbf{x}_t$. While for an anomaly detection approach, it is expected to judge whether status at time stamp t, namely x_t is normal based on the same duration of segments. Finally, for training, the order of segments is maintained and the segments are cut into segment sequences of length L, a hyper-paramenter called segment sequence length.

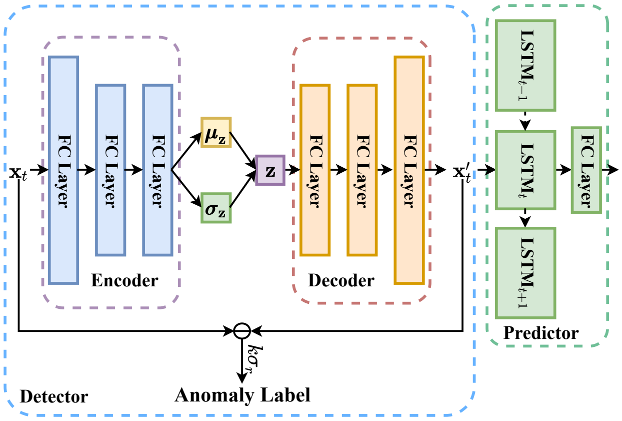


Fig. 2: The network structure of SeqVL. Two major blocks, VAE and LSTM, form a sequential structure. The segment $\mathbf{x}_t$ is reconstructed to $\mathbf{x}'_t$ with VAE for anomaly detection, and then $\mathbf{x}'_t$ is taken as the input of LSTM for robust trend prediction.

3.2 Sequential VAE-LSTM

3.2.1 *Anomaly Detection*

In this paper, we aim to address anomaly detection and trend prediction under one framework. Let's consider the anomaly detection first. Assuming that 1) the latent variable of the segment $\mathbf{x}_t$, namely $\mathbf{z}$ follows multivariate standard Gaussian distribution $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and 2) the anomalous status are rare occurrences, a time series of $\mathbf{x}'_t$ that is free of anomlies can be largely reconstructed by variational autoencoder (VAE) as Eqn. 2,

$$\mathbf{x}'_t = \mathrm{VAE}(\mathbf{x}_t). \tag{2}$$

The anomaly detection pipeline with VAE is shown in the left part of Fig. 2. As shown in the figure, the standard VAE is adopted in the design. As an approximation to the intractable true posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$, the approximate posterior distribution, set to follow a diagonal Gaussian distribution: $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\sigma}^2_\mathbf{z}\mathbf{I})$, is fitted by the encoder. Therefore, on the encoder side, the time series of one segment $\mathbf{x}_t$ is encoded into $\boldsymbol{\mu}_\mathbf{z}$ and $\boldsymbol{\sigma}_\mathbf{z}$ by a three-layer encoder. On the decoder side, sampled from $\mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\sigma}^2_\mathbf{z}\mathbf{I})$, $\mathbf{z}$ is decoded to $\mathbf{x}'_t$ with symmetric structure as the encoder. According to the evidence lower bound [4], [28], our VAE is trained with loss function as Eqn. 3,

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\mathbf{x}_t) &= \|\mathbf{x}_t - \mathbf{x}'_t\|_2^2 + \text{KL}\Big(\mathcal{N}(\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\sigma}_\mathbf{z}^2\mathbf{I})\Big\|\mathcal{N}(\mathbf{0},\mathbf{I})\Big) \\ &= \|\mathbf{x}_t - \mathbf{x}'_t\|_2^2 + \frac{1}{2}\Big(-\log\boldsymbol{\sigma}_\mathbf{z}^2 + \boldsymbol{\mu}_\mathbf{z}^2 + \boldsymbol{\sigma}_\mathbf{z}^2 - 1\Big),\end{aligned} \tag{3}$$

where the first term is the reconstruction loss of $\mathbf{x}_t$ and the second term is Kullback-Leibler divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$.

Since the anomaly status are in rare occurrence, whose distributions are different from that of the normal status, these anomalies are not recovered by the decoder. As a result, the anomaly detection becomes as easy as checking the reconstruction loss. In our implementation, according to [1], the difference that the last status in the segment $\mathbf{x_t}$ from that in the recovered segment $\mathbf{x}'_t$, namely x_t in $\{x_{t-w_0+1}, \ldots, x_t\}$ from x'_t in $\{x'_{t-w_0+1}, \ldots, x'_t\}$, is checked in order to detect anomalies as soon as possible in detection. However, different from [1], if the squared error of x_t from x'_t is higher than $k\sigma_r$, x_t is viewed abnormal. $k\sigma_r$ is set the same for time series from the same evaluation dataset. k is threshold that is chosen according to the evaluation metrics and σ_r is the standard deviation of the squared errors of x_t from x'_t. Compared to [1], this threshold is able to be adaptive well to the different distributions of the squared errors.

Implementation details Due to the symmetric structure of VAE, the size of the input layer of the encoder and the output layer of the decoder is set to be the same as window size w_0. ReLU is adopted as the activation function for both layers. The number of $\mathbf{z}$ dimensions is set to K. The layer of $\boldsymbol{\mu}_\mathbf{z}$ and the layer of $\boldsymbol{\sigma}_\mathbf{z}$ which learns $\log\boldsymbol{\sigma}_\mathbf{z}$ to cancel the activation function, are both fully-connected layers. Because of the symmetry of the auto-encoder, the hidden layers of the encoder and the decoder are both two layers with the ReLU activation function, each of which is with h_l units.

With the assumption that anomaly status follow different distribution from the normal and they are in a rare occurrence, simple VAE is already able to undertake the anomaly detection. However, VAE alone is unable to fulfill the trend prediction since VAE is unable to encode/decode a future status x_{t+1} outside the window. Meanwhile, the recovered $\mathbf{x}'_t$ is expected free of anomalies. If $\mathbf{x}'_t$ is used for trend prediction, the prediction block becomes robust to noises and possible anomalies. In the following, we are going to show how the output of VAE is capitalized for trend prediction by LSTM.

3.2.2 Trend Prediction

To achieve trend prediction in our framework, LSTM is adopted in our design. As shown in right part of Fig. 2, LSTM takes the output from VAE block, and it is expected to predict x_{t+1} based on $\mathbf{x}'_t$. Namely, the loss function is given as

$$\mathcal{L}_{\text{LSTM}}(\mathbf{x}'_t, x_{t+1}) = \|x_{t+1} - y_t\|_2^2, \tag{4}$$

where y_t is the expected output from LSTM. The loss function simply measures the mean squared error between the true status at time stamp $t+1$ and the predicted value y_t.

In the LSTM block, given the output of the previous time stamp is $\mathbf{h}_{t-1}$, the reconstructed $\mathbf{x}'_t$ is taken as the input of the current time stamp, the state of the current time stamp is computed by Eqn. 5

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}'_t] + \mathbf{b}_c). \tag{5}$$

Then the update gate and the forget gate at the current time stamp are computed with Eqn. 6 and Eqn. 7

$$\boldsymbol{\Gamma}_u = \sigma(\mathbf{W}_u[\mathbf{h}_{t-1}, \mathbf{x}'_t] + \mathbf{b}_u), \tag{6}$$

$$\boldsymbol{\Gamma}_f = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}'_t] + \mathbf{b}_f), \tag{7}$$

where $\sigma(\cdot)$ is the activation function that controls the flow of information. The state of the current time stamp is updated with Eqn. 8

$$\mathbf{c}_t = \boldsymbol{\Gamma}_u \times \tilde{\mathbf{c}}_t + \boldsymbol{\Gamma}_f \times \mathbf{c}_{t-1}. \tag{8}$$

The output gate is governed by Eqn. 9

$$\boldsymbol{\Gamma}_o = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}'_t] + \mathbf{b}_o). \tag{9}$$

Finally, the output of the current time stamp is computed as follows.

$$\mathbf{h}_t = \boldsymbol{\Gamma}_o \times \tanh(\mathbf{c}_t) \tag{10}$$

In order to map $\mathbf{h}_t$ to the predicted value y_t, a fully connected layer is introduced to attached to LSTM block. The predicted value y_t is computed as Eqn. 11.

$$y_t = \mathbf{w}_y\mathbf{h}_t + b_y \tag{11}$$

During the training of the whole network, the loss functions of unsupervised anomaly detection and trend prediction should be balanced. So the overall loss function for the network is

$$\mathcal{L}_{\text{SeqVL}}(\mathbf{x}_t, x_{t+1}) = \mathcal{L}_{\text{VAE}}(\mathbf{x}_t) + \lambda\mathcal{L}_{\text{LSTM}}(\mathbf{x}'_t, x_{t+1}), \tag{12}$$

where $\mathcal{L}_{\text{VAE}}(\mathbf{x}_t)$ is the loss function of unsupervised anomaly detection and $\mathcal{L}_{\text{LSTM}}(\mathbf{x}'_t, x_{t+1})$ is the loss function of trend prediction. λ is another hyper-parameter to balance the learning of the two tasks.

Notice that the proposed framework is an unsupervised anomaly detection in the sense no labeled data is required. Moreover, the trend prediction block is a natural extension over the anomaly detection block as we make full use of the output from VAE. The reconstructed time series segment from the VAE considerably reduces the noises mixed with the input segment. The LSTM is therefore able to capture the data regular patterns well. As will be revealed in the experiment, the framework performs well on both tasks. To the best of our knowledge, this is the first piece of work that integrates the anomaly detection and trend prediction into one framework. Note that both the anomaly detection block and trend prediction model in our framework are trained and therefore boost the performance of each other. This is essentially difference from [18], [19], [20], which are trained solely for prediction but used for detection.

4 Experiments

In this section, the effectiveness of the proposed approach is studied in comparison to approaches that are designed for anomaly detection and trend prediction in the literature. **KPI** [29] and **Yahoo** [30] are adopted in the evaluation. **KPI** dataset is released by the AIOps Challenge Competition [29], which contains desensitized time series of **KPI** with anomaly annotation from real-world applications and services. The raw data are harvested from Internet companies such as Sogou, Tencent, eBay, Baidu, and Alibaba. They are minute-level operations time series. In our evaluation, this dataset is used for both anomaly detection and trend prediction. **Yahoo** dataset is released by Yahoo Labs. It is mainly built for anomaly detection evaluation. It contains both real and synthetic time series. The brief information about these two datasets are summarized in Tab. 1. Since the status from **Yahoo** dataset demonstrate different periodic patterns across the time stamps, it is not suitable for trend prediction evaluation. So following the convention in the literature [23], it is adopted for anomaly detection only.

On **KPI** dataset, parameter λ in Eqn. 12 is empirically set to *1*. Other hyper-parameters are selected according to [1]. The window size w_0 is set to *120*, which is equivalent to *2* hours. The number of **z** dimensions, namely K is set to be *5*. h_l and the size of h_t are set to *100*. In training, the segment sequence length L is set to *256* and the number of epoch is set to *250*. Adam optimizer [31] is used with an initial learning rate of 10^{-3}. The learning rate decays by *0.75* every *10* epoch. l_2-regularization is applied to all the layers with a weight of 10^{-3}. The gradients are cliped below *10.0* by norm. On **Yahoo** dataset, parameter λ in Eqn. 12 is set to *10*. The window size w_0 is set to *30*. h_l and the size of h_t are set to *24*. The segment sequence length L is set to *300*. The gradients are cliped below *12.0* by norm. The learning rate decays by *0.8* every *10* epoch. The rest of configurations on the training is kept the same as on **KPI** dataset.

4.1 Evaluation Protocol

For robust trend prediction, Mean Squared Error (MSE) as shown in Eqn. 13 is adopted in the evaluation.

$$\mathrm{MSE} = \sqrt{\frac{1}{N - w_0} \sum_{t=w_0}^{N-1} (x_{t+1} - y_t)}, \tag{13}$$

where w_0 is the window size and y_t is the predicted value from the LSTM block. It basically measures the average error that the predicted y_t differs from x_{t+1}. In the evaluation, the first half of the time series is used to train the model, while the second half is used for evaluation. Since the anomalies in the time series are annotated, they are removed from the series when they are used for prediction evaluation. The removed time stamps are filled with the expected normal status. This is achieved with the same scheme used for filling missing status during the pre-processing step. After calculating the MSE of each time series, we plot them as box plots to visualize the MSE score as well as the stability of the prediction models. The lower is MSE and the shorter is the width of the box, and thus the higher is the prediction precision.

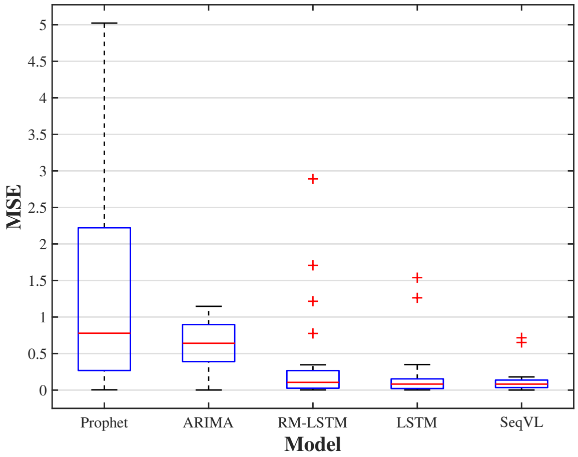
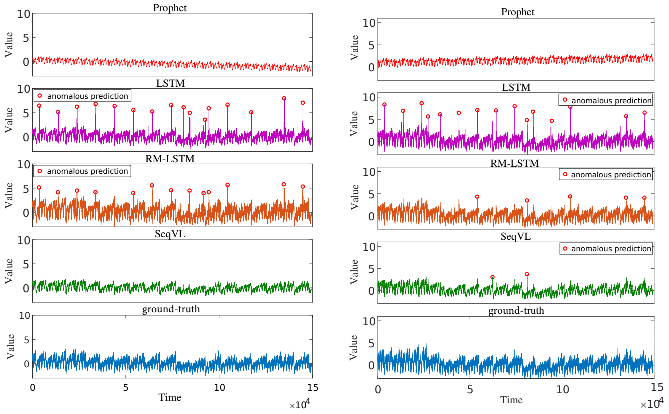


Fig. 3: The prediction performance of SeqVL in comparison to Prophet, LSTM and RM-LSTM on **KPI** dataset.



(a) Sequence-16

(b) Sequence-18

Fig. 4: The prediction results from SeqVL, Prophet, LSTM and RM-LSTM on Sequence-16 and Sequence-18 from **KPI** dataset. As shown from the figure, LSTM and RM-LSTM are sensitive to anomalies, there are many anomalous predictions in the results.

For unsupervised anomaly detection, following [23], [29], each time series is divided into two halves. The first half is used for training and the second half is used for evaluation. We evaluate our model with three metrics, precision, recall and F_1-score. In practice, people do not care whether an anomaly is detected successfully at the moment it appears, instead we care about in which time duration an anomaly is successfully detected within a small allowed delay. As a result, following the practice in [1], [23], [29], if the model detects anomalies no later than D time stamps after the start time stamp of the anomaly interval, each abnormal time stamp in the anomaly interval is viewed as a true positive. Otherwise, each abnormal time stamp in the anomaly interval is counted as a false negative.

4.2 Robust Trend Prediction

The prediction performance of our approach is studied on **KPI** dataset. It is compared to representative approaches in the literature and industry. They are classic approaches

TABLE 1: Summary over the datasets

Dataset	# Series	# Time-stamps	# Anomalies	Granularity
KPI	29	5,922,913	134,114/2.26%	Minute
Yahoo	367	572,966	3,896/0.68%	Hour

ARIMA and Prophet, which is recently developed by Facebook. Our approach is also compared to standard LSTM that is popularly used for trend prediction. Usually the time series are mixed with noises and anomalies, which impact the performance of LSTM. In order to see the performance of LSTM with relatively clean data, another run, namely RM-LSTM is produced. For RM-LSTM, the input time series is cleaned. Namely, the apparently large or small status along the time series (suspected anomalies) are removed and filled with expected normal values, which is the same as filling the missing status in the preprocessing step. For all the approaches, they are trained on the first half of the time series, and tested on the rest.

The prediction performance is shown in Fig. 3. As shown in the figure, the performance from classic approaches is very poor. Both ARIMA and Prophet show high prediction errors. As pointed out in [32], the standard ARIMA normally converges to a constant in long-term prediction when the time series is stationary. In contrast, LSTMs perform significantly well. Particularly, our approach demonstrates the smallest prediction error. Although the configuration of our approach, standard LSTM and RM-LSTM are similar. The performance difference between them is still quite significant. The input sequence of RM-LSTM is cleaner, nevertheless its performance is still slightly poorer than standard LSTM. This is mainly because cleaning data with a hard threshold may hurt the normal data distribution as well. In contrast, our way that cleans the time sequence by VAE turns out to be a much better choice. Fig. 4 shows two sequences from **KPI** dataset that are reconstructed by Prophet, LSTM, RM-LSTM and SeqVL. It is clear to see LSTM and RM-LSTM are able to adapt to the trend of the time series well. However, many anomalies are produced since they are too sensitive to noises. In contrast, the sequences are reconstructed by SeqVL with very few anomalies thanks to the clean input from the VAE block.

4.3 Unsupervised Anomaly Detection

Our anomaly detection approach is compared to representative approaches in the literature. They are VAE, DONUT [1], SPOT, DSPOT [26], SR and SR-CNN [23]. The results of SPOT, DSPOT, SR, and SR-CNN are quoted from [23]. Among these approaches, VAE is actually part of our approach. For fair comparisons, its hyper-parameters are set to be the same as SeqVL. Essentially, DONUT is a variant of VAE. Its hyper-parameters on **KPI** are set following the setting presented in [1]. While on **Yahoo** dataset, its hyper-parameters are set to be the same as of SeqVL for fair comparisons. SR-CNN is an approach proposed recently. The CNN model used for detection is trained with extra large amount of anomaly-free time series, in which the anomalies are artificially injected. This is the only supervised approach considered in our study.

TABLE 2: Performance comparison on Anomaly Detection Task on **KPI** and **Yahoo**. The supervised approach is marked with '*'

	KPI			**Yahoo**		
Approach	F_1-score	Precision	Recall	F_1-score	Precision	Recall
SPOT	0.217	0.786	0.126	0.338	0.269	0.454
DSPOT	0.521	0.623	0.447	0.316	0.241	0.458
DONUT	0.595	0.735	0.500	0.501	0.669	0.401
VAE	0.657	0.711	0.611	0.569	0.684	0.487
SR	0.622	0.647	0.598	0.563	0.451	0.747
***SR-CNN**	**0.771**	0.797	0.747	0.652	0.816	0.542
SeqVL	0.664	0.716	0.619	**0.661**	0.891	0.526

The performance of anomaly detection from all afore mentioned approaches are presented on Tab. 2. As shown in the table, the proposed approach considerably outperforms all the state-of-the-art unsupervised approaches on both datasets. The performance remains stable across two different datasets. In contrast, approaches such as DSPOT, DONUT, and SR see a significant performance fluctuation across different datasets.

As shown by our approach, the detection performance of VAE is boosted when the LSTM block is incorporated for trend prediction. Compared with standard VAE, the segments fed to our network are organized in sequential order. Such that the temporal patterns go beyond the sliding window are learnt. To this end, it is clear to see our network is not a simple combination of two classic neural networks. Although SR-CNN performs the best, a big amount of clean time series is required for the training. According to [23], *65* million points is used for training, which is around *10* times bigger than that of **KPI** and **Yahoo** datasets. In practice, it is unrealistic to find such a big amount of time series free of anomalies.

5 Conclusion

We have presented our model SeqVL for both unsupervised anomaly detection and robust trend prediction on IT operations. In our solution, the detector (VAE) boosts its performance by training the model with segments in sequential order that is maintained by the predictor (LSTM). In turn the robust trend prediction of LSTM is achieved by taking the reconstructed status from the detector (VAE). As shown in the experiments, our approach outperforms the state-of-the-art unsupervised approaches for anomaly detection as well as the approaches for trend prediction on public datasets. Adaptive thresholding which is expected to further boost the performance of anomaly detection is our future research work.

Acknowledgements

We would like to express our sincere thanks to Bonree Inc., Beijing, China, for their full support of this work.

References

[1] H. Xu, Y. Feng, J. Chen, Z. Wang, H. Qiao, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, and D. Pei, "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," in *Proceeding of the International Conference on World Wide Web*, vol. 2, pp. 187–196, ACM, 2018.

[2] S. Hochreiter and S. Jürgen, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, nov 1997.

[3] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, Jun. 1989.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, Dec. 2013.

[5] W. Chen, H. Xu, Z. Li, D. Pei, J. Chen, H. Qiao, Y. Feng, and Z. Wang, "Unsupervised anomaly detection for intricate KPIs via adversarial training of vae," in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1891–1899, IEEE, Apr. 2019.

[6] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer, and E. Smirni, "Practise: Robust prediction of data center time series," in *Proceedings of the IFIP/IEEE International Conference on Network and Service Management*, pp. 126–134, IEEE, Nov. 2015.

[7] A. Zameer, J. Arshad, A. Khan, and M. A. Z. Raja, "Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks," *Energy Conversion and Management*, vol. 134, pp. 361–372, Feb. 2017.

[8] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, pp. 443–473, Jan. 2006.

[9] G. E. Box and G. M. Jenkins, *Time series analysis: Forecasting and control*. Holden-Day, 1976.

[10] J. D. Salas, *Applied modeling of hydrologic time series*. Water Resources Publication, 1980.

[11] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[12] P. Kalekar, "Time series forecasting using holt-winters exponential smoothing," *Kanwal Rekhi School of Information Technology*, pp. 1–13, 2004.

[13] Facebook, "Prophet: Tool for producing high quality forecasts for time series data that has multiple seasonality with linear or nonlinear growth.," 2017.

[14] Hawkular, "Hawkular for monitoring services: Metrics, alerting, inventory, application performance management.," 2014.

[15] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, "Robust online time series prediction with recurrent neural networks," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics*, pp. 816–825, IEEE, Oct. 2016.

[16] A. H. Yaacob, I. K. Tan, S. F. Chien, and H. K. Tan, "Arima based network anomaly detection," in *Proceedings of the IEEE International Conference on Communication Software and Networks*, pp. 205–209, IEEE, 2010.

[17] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 89–94, 2015.

[18] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Proceedings of the International Conference on Machine Learning - Time Series Workshop*, vol. 13, pp. 1–5, Jan. 2017.

[19] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *Proceedings of the IEEE International Conference on Data Mining Workshops*, vol. 2017-Novem, pp. 103–110, IEEE, Nov. 2017.

[20] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 387–395, ACM, 2018.

[21] C. Campbell and K. P. Bennett, "A linear programming approach to novelty detection," in *Advances in Neural Information Processing Systems*, 2001.

[22] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Piatt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*, pp. 582–588, 2000.

[23] H. Ren, Q. Zhang, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, and J. Tong, "Time-series anomaly detection service at microsoft," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3009–3017, ACM, Jun. 2019.

[24] Z. Ding and M. Fei, "An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window," *IFAC Proceedings Volumes*, vol. 46, pp. 12–17, Jan. 2013.

[25] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *Proceedings of the International Conference on Machine Learning*, vol. 48, pp. 2712–2721, 2016.

[26] A. Siffer, P. A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F1296, pp. 1067–1075, ACM Press, 2017.

[27] L. de Haan and A. Ferreira, *Extreme Value Theory*. Springer Series in Operations Research and Financial Engineering, Springer New York, 2006.

[28] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, Feb. 2015.

[29] AIOpsChallenge, "KPI anomaly detection competition," 2017.

[30] YahooLabs, "S5 - a labeled anomaly detection dataset, version 1.0," 2015.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, Dec. 2014.

[32] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.