DA-LSTM-VAE: Dual-Stage Attention-Based LSTM-VAE for KPI Anomaly Detection

Yun Zhao, Xiuguo Zhang *, Zijing Shang and Zhiying Cao *

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
* Correspondence: zhangxg@dlmu.edu.cn (X.Z.); czysophy@dlmu.edu.cn (Z.C.)

Abstract: To ensure the normal operation of the system, the enterprise’s operations engineer will monitor the system through the KPI (key performance indicator). For example, web page visits, server memory utilization, etc. KPI anomaly detection is a core technology, which is of great significance for rapid fault detection and repair. This paper proposes a novel dual-stage attention-based LSTM-VAE (DA-LSTM-VAE) model for KPI anomaly detection. Firstly, in order to capture time correlation in KPI data, long–short-term memory (LSTM) units are used to replace traditional neurons in the variational autoencoder (VAE). Then, in order to improve the effect of KPI anomaly detection, an attention mechanism is introduced into the input stage of the encoder and decoder, respectively. During the input stage of the encoder, a time attention mechanism is adopted to assign different weights to different time points, which can adaptively select important input sequences to avoid the influence of noise in the data. During the input stage of the decoder, a feature attention mechanism is adopted to adaptively select important latent variable representations, which can capture the long-term dependence of time series better. In addition, this paper proposes an adaptive threshold method based on anomaly scores measured by reconstruction probability, which can minimize false positives and false negatives and avoid adjustment of the threshold manually. Experimental results in a public dataset show that the proposed method in this paper outperforms other baseline methods.

Keywords: KPI anomaly detection; VAE; LSTM; attention mechanism; adaptive threshold

1. Introduction

With the vigorous development of artificial intelligence, more and more companies are introducing AI into the operation and maintenance field. The concept of artificial intelligence for IT operations (AIOps) came into being [1]. Compared with the application of artificial intelligence technology in other fields, AIOps has the necessary elements for a certain vertical field of artificial intelligence technology. First, this concerns the actual application scenarios. Almost all key technologies in AIOps are inseparable from machine learning algorithms. Second, it involves massive amounts of data. The enterprise’s operation and maintenance monitoring platform generates massive amounts of monitoring indicator data and operation and maintenance logs constantly. Third, it involves a large number of annotations. Since operation and maintenance personnel are domain experts themselves, a large amount of labeled data will be generated in their daily work. Massive data and annotations lower the threshold for researching machine learning algorithms, which is beneficial to the rapid progress of algorithm research. Therefore, AIOps has broad prospects in the field of machine learning and is worthy of the attention of researchers in the field of machine learning.

KPI (key performance indicator) refers to the monitoring indicators (such as delay, throughput, etc.) for operation and maintenance objects such as services and systems. Its storage form is also known as time series. KPI anomaly detection is an underlying core technology of AIOps. It aims to discover abnormal behaviors in the software and hardware of internet services by analyzing KPI curves, such as increased access delays, network
device failures, and sharp reductions in access users, etc. [2]. KPI anomaly detection can provide a decision-making basis for subsequent alarm, automatic stop loss, root cause analysis, etc. In the actual scenes, due to the scarcity of outlier data, various types of anomalies, and various types of KPI, it brings great challenges to anomaly detection.

At present, the industry and academia have proposed a series of anomaly detection algorithms, such as statistical method, prediction method, supervised learning method, unsupervised learning method, and deep generation model method. The most commonly used statistical method is the k-sigma [3] theory, assuming that the data obey a normal distribution, and the data with a small probability of occurrence are regarded as an outlier. However, the actual data distribution does not always follow a normal distribution. The basic idea of the prediction method is to determine whether it is abnormal by comparing the difference between the predicted value and the real value. This method mainly includes the traditional time series prediction models ARIMA [4], LSTM [5], and Holter–Winter [6]. The method based on supervised machine learning can avoid parameter adjustment in traditional statistical algorithms, such as the scalable general anomaly detection framework EGADS [7] proposed by Yahoo and the integrated supervision method Opprentice [8] proposed by Tsinghua NetMan Lab. Although machine learning methods can achieve higher accuracy, the disadvantage is that they require a great deal of manual annotation, covering comprehensive data types and abnormal conditions, which is extremely difficult to achieve in actual scenes [9]. Anomaly detection based on unsupervised learning and deep generative models has become more and more popular in recent years, such as one-class SVM [10], GMM [11], VAE [12–14], and CVAE [15]. The deep generative model can more fully capture the complex features of the data and has higher accuracy, so it has received extensive attention.

As a typical representative of the deep generative model, VAE has achieved good results in KPI anomaly detection. However, it does not consider the time dependence of the data, which limits its applicability to time series. Moreover, there is only one fixed-length vector between the encoder and decoder, which leads to its limitations. This fixed-length vector cannot fully store all the related information in the sequence, and it needs to pay attention to the relevant parts of the sequence selectively. In addition, whether VAE uses reconstruction error or reconstruction probability as anomaly scores, a threshold must be set to obtain a more accurate result. However, for massive different types of KPIs, it is difficult to set thresholds uniformly.

Based on the above analysis, this paper proposes a new network structure (DA-LSTM-VAE) that combines attention mechanism, LSTM, and VAE and applies it to KPI anomaly detection. This method aims to automatically discover anomalies in KPI time series, such as abnormal points or abnormal intervals, and then notify operations engineer through alarms. The main contributions of this paper are as follows:

1. An improved VAE network structure is proposed. The traditional neurons in the encoder and decoder of VAE are replaced by LSTM, which can better extract the time dependence of KPI data. In addition, LSTM-VAE can automatically extract the potential features of KPI data, avoiding the dependence on manual feature extraction;

2. The dual-stage attention mechanism is introduced in LSTM-VAE for the first time, and the new model structure is named DA-LSTM-VAE. In the first stage, attention is weighted to the original KPI time series. In this way, important input sequences can be highlighted, thereby achieving the effect of denoising. In the second stage, attention is weighted to latent variables of LSTM-VAE. In this way, important latent variable representations across all time steps can be adaptively selected to capture the long-term dependence of KPI time series;

3. In the anomaly detection stage, an adaptive threshold method is proposed. The reconstruction probability is taken as the anomaly score. Reconstruction probability is a probability measure considering the variability of variable distribution, which is more objective than reconstruction error. The possible fluctuation range of current
anomaly scores is judged according to historical anomaly scores so as to achieve the effect of adaptive threshold.

2. Related Work

2.1. Anomaly Detection Based on VAE

The variational autoencoder (VAE) [16] is an unsupervised generative network model consisting of an encoder and decoder. The encoder is used to learn the distribution of training data and generate compression values of training data, and the decoder reconstructs the compressed data. An et al. [12] proposed an anomaly detection method based on VAE and introduced a novel probabilistic anomaly score that takes into account the variability of the data. Kawachi et al. [13] proposed a new abnormal probability representation method, which defined the normal and anomaly distributions using the analogy between a set and the complementary set. Finally, these distributions are applied to the unsupervised VAE. Xu et al. [14] proposed donut and applied the deep generative model to KPI anomaly detection for the first time. Donut puts forward innovations such as M-ELBO and MCMC iteration based on VAE, which has excellent performance on periodic KPIs. Li et al. [17] proposed ACVAE, a KPI anomaly detection algorithm through active learning and contrast-VAE-based detection models. The out-of-band information (including background information and feedback information) is integrated in VAE, so it has a better detection effect. Zhang et al. [18] proposed an unsupervised KPI anomaly detection approach, named AnoTransfer, by combining a novel VAE-based KPI clustering algorithm with an adaptive transfer learning strategy. VAE can jointly train a basic model by clustering KPI fragments with similar shapes. The performance of the anomaly detection model is improved by using KPI with a similar shape in transfer learning. However, VAE does not consider the time dependence of the data, thus limiting its applicability to time series. Therefore, introducing LSTM into VAE can extract the time dependence and correlation of data to obtain higher data reconstruction accuracy.

2.2. Anomaly Detection Based on LSTM-VAE

If the operation and maintenance data detection is regarded as a Seq2Seq [19] problem, the sequence can be understood as time series. Encoder–decoder is a widely used architecture in the Seq2Seq problem. Malhotra et al. [20] proposed an encoder–decoder based on LSTM. The encoder learns the vector representation of input time series, and the decoder uses this representation to reconstruct time series. Then, the reconstruction error is used to detect anomalies. Daehyung et al. [21] proposed a VAE based on LSTM, which can preferably capture the time dependence of data. In addition, a state-based threshold is introduced to increase detection sensitivity and reduce false-alarm rates. Lin et al. [22] combined VAE and LSTM. The VAE encoder first extracts the features of input data and then inputs extracted features into LSTM network for fault detection or classification. Su et al. [23] combined VAE and GRU. GRU captures the complex time dependence between multivariate observations in space, and VAE maps observations to random variables. In the phase of anomaly detection, this method judges entity anomalies by several univariate time series with low reconstruction probability. Niu et al. [24] proposed LSTM-based VAE-GAN for time series anomaly detection. The anomaly score consists of the reconstruction error of VAE and the discrimination results of the discriminator, which makes it more able to distinguish anomalies from normal data. Chen et al. [25] proposed a joint model predictor anomaly detector (PAD). The prediction block (LSTM) takes clean input from the reconstructed time series by VAE, which makes it robust to the anomalies and noise for prediction task. In the meantime, the LSTM block maintains the long-term sequential patterns, which are out of the sight of a VAE encoding window. This leads to the better performance of VAE in anomaly detection than when it is trained alone. Chen et al. [26] proposed a LSTM-based semi-supervised VAE anomaly detection strategy called LR-SemiVAE. By introducing an LSTM network into the VAE encoder and decoder, the model can fully learn the time dependence of KPI time series. However, the biggest limitation of the encoder–decoder model
is that the only connection between the encoder and decoder is a fixed-length vector, which cannot fully store all relevant information in the sequence. It is necessary to selectively pay attention to the relevant parts of sequence to improve reconstruction performance, so attention mechanism is introduced into encoder–decoder [27]. In addition, traditional attention mechanism is only used in the input stage of the decoder [28–30]. Yao et al. [31] also introduced an attention mechanism in the input stage of the encoder. In this way, the encoder can focus on the important driving factors of input data instead of treating all factors equally. Therefore, this paper introduces attention mechanism in both the encoder and decoder input stage of LSTM-VAE.

3. Anomaly Detection Method

3.1. Method Flow

The flow of anomaly detection method proposed in this paper is shown in Figure 1, which includes two modules: offline training and online detecting. Data preprocessing is the common module of two modules, which mainly standardizes the data and fills in missing values. After data preprocessing, the data are input to the offline training module for learning, thereby capturing the normal pattern of the data sequence. The offline training module uses historical data batch processing to train the model. The whole process minimizes the loss function so that the model is optimal. According to the trained model, the online detecting module calculates the reconstruction probability of new observation value after preprocessing and obtains anomaly scores. Then, the adaptive threshold is used to determine whether anomaly scores are within the normal fluctuation range. Anomaly detection results are finally output.

![Method flow chart](image)

Figure 1. Method flow chart.

In order to make the method in this paper easier to understand, the main symbols in this paper are summarized as shown in Table 1.

Table 1. Main symbols table.

| Symbolic Representation | Symbolic Interpretation |
|-------------------------|-------------------------|
| $x_t$                   | KPI time series after data preprocessing |
| $h_t$                   | LSTM hidden layer state |
| $e_t$                   | Attention scores at the time attention stage |
| $a_t$                   | Attention weights at the time attention stage |
Table 1. Cont.

| Symbolic Representation | Symbolic Interpretation |
|-------------------------|-------------------------|
| \( \tilde{x}_t \)     | KPI time series obtained through time attention stage |
| \( \mu_z \)           | Mean value estimated by encoder |
| \( \sigma_z \)        | Variance estimated by encoder |
| \( \varepsilon \)      | Auxiliary noise variables |
| \( z_t \)             | Latent variables obtained by reparameterization |
| \( q_\theta(z_t|\tilde{x}_t) \) | Approximate posterior distribution |
| \( p_\theta(z_t) \)   | True posterior distribution |
| \( l_t \)             | Attention scores at the feature attention stage |
| \( \beta_t \)         | Attention weights at the feature attention stage |
| \( c_t \)             | Context variables obtained through feature attention stage |
| \( y_t \)             | Reconstructed KPI time series |
| \( L_{\text{vae}} \)  | Loss function of the model |
| \( \lambda_{KL} \)    | Weight coefficient of KL loss |
| \( S_t \)             | Anomaly scores for KPI time series |

3.2. Data Preprocessing

At present, most methods focus on the discussion of algorithms and ignore the necessity of data preprocessing. In the real world, data are usually incomplete, inconsistent, and highly susceptible to noise. In order to eliminate the influence of data on the algorithm, the missing value and standardization of data are firstly processed in this paper.

3.2.1. Missing Value Processing

Data loss occurs in the process of data transmission, sampling, and recording. Therefore, it is necessary to preprocess time series before establishing time series model. The processing of missing values in a time series is an important part of preprocessing. Improper processing will result in a large number of errors and seriously affect results. Traditional missing-value processing methods include zero value or mean value replacement. The missing-value processing method in ref. [14] is adopted in this paper. This method fills missing points with 0 instead of using some existing algorithms. Since VAE is a generating model, it can generate data and fill in missing values. Here, we assume that the complete data are represented as \( X = \{X_{\text{obs}}, X_{\text{mis}}\} \), where \( X_{\text{obs}} \) represents observable data, and \( X_{\text{mis}} \) represents missing data. The input sample \( X \) is transformed from \( \{X_{\text{obs}}, X_{\text{mis}}\} \) to \( \{X_{\text{obs}}, X'_{\text{mis}}\} \) after reconstruction, that is, the reconstructed sample \( X' \). \( X' \) is the input of MCMC [32] iterative. According to Bayes’ theorem, the posterior probability of missing data under the condition of observed data can be obtained. As shown in Formula (1),

\[
P(X_{\text{mis}}'|X_{\text{obs}}) = \int f(X_{\text{obs}}|X_{\text{mis}}', \alpha)k(\alpha|X_{\text{obs}})d\alpha, \tag{1}
\]

where \( k(\alpha|X_{\text{obs}}) \) is the posterior probability of \( \alpha \) under the observable data, but the above formula cannot be directly integrated. \( P(X_{\text{mis}}'|X_{\text{obs}}) \) is simulated by MCMC method, and then, \( X_{\text{mis}}' \) of the time series was randomly extracted from it.

3.2.2. Data Standardization

Data standardization (normalization) processing is a basic work of data mining. Different evaluation indicators often have different dimensions and dimension units, which will affect the results of data analysis. In order to eliminate the dimension impact between indicators, data standardization processing is required to solve the comparability between data indicators. After the original data restandardized, all indicators are in the same order of magnitude, which is suitable for comprehensive comparative evaluation. This paper uses the maximum–minimum normalization method to process the KPI data so that the data are distributed between 0–1. As shown in Formula (2),
\[ x^* = \frac{x - \min}{\max - \min}. \] (2)

3.3. Offline Training

Offline training aims to provide a model for online detecting, and its model is LSTM-VAE based on dual-stage attention. The attention mechanism is introduced in both the encoder and decoder input stages of LSTM-VAE to highlight important time points and features, and considering the disappearance of \( KL \) divergence during model training, the model is optimized by minimizing the loss function.

3.3.1. Dual-Stage Attention-Based LSTM-VAE (DA-LSTM-VAE) Model

Offline training aims to provide a model for online detecting, and its model is LSTM-VAE based on dual-stage attention. The attention mechanism is introduced in both the encoder and decoder input stages of LSTM-VAE to highlight important time points and features, and considering the disappearance of \( KL \) divergence during model training, the model is optimized by minimizing the loss function. The dual-stage model includes both the early stage of stimulus selection and the later stage of stimulus selection. The first stage selects the primary stimulus characteristics, and the second stage uses the classification information to decode the stimulus [33]. The dual-stage attention mechanism can select the relevant driving sequence and enhance the long-term dependence of time series [31,34]. This paper names the dual-stage attention as time attention and feature attention, respectively. In this paper, a sliding window with a certain step length is used to divide the time series into sub-sequences, which correspond to the input variables. The time attention is used to weight the original data sample \( x \) to obtain the updated time series \( \tilde{x} \). \( \tilde{x} \) is encoded as latent variables \( z \). \( z \) is transformed into context variable \( c \) by the feature attention. \( c \) is decoded to output the reconstructed data sample \( y \).

Time attention weights original time series to highlight important input sequences. The process is shown in Figure 2.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure2.png}
\caption{Attention-mechanism-weighted original time series.}
\end{figure}

The encoder used in this paper is based on LSTM neural network units [35]. In the time attention phase, it is necessary to calculate the hidden layer state \( h_t \) of encoder at the current moment based on input \( x_t \) at the current moment and the hidden layer state \( h_{t-1} \) of encoder at the previous moment. The calculation process is shown in Formula (3).

\[ h_t = \text{LSTM}(h_{t-1}, x_t), \] (3)

LSTM is a special RNN, which can solve the long-term dependency problem well. The state-update process of the LSTM unit is shown as follows:

\[ f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \] (4)
\[ i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \]  
\[ \tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \]  
\[ C_t = f_t \ast C_{t-1} + i_t \ast \tilde{C}_t, \]  
\[ O_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \]  
\[ h_t = O_t \ast \tanh(C_t), \]

where \( f_t, i_t, \tilde{C}_t, C_t, O_t \) represent the forgetting gate, input gate, candidate memory unit state, memory unit, and output gate, respectively. \( W_f, W_i, W_o, W_c \) represent the weight matrix of corresponding gate and memory unit, respectively; \( b_f, b_i, b_o, b_c \) represent the corresponding gate and memory unit bias items, respectively. \( \sigma \) represents the sigmoid activation function.

Next, the attention mechanism is constructed through hidden layer state \( h_{t-1} \) and unit state \( C_{t-1} \) of the encoder at the previous moment.

\[
e_t = \alpha_t^T \tanh(W_e[h_{t-1}; C_{t-1}] + U_e x + b_e),
\]

\[ a_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \]

Among them, \([*;*]\) is an aggregation operation. \( v_e, W_e, U_e, \) and \( b_e \) are learnable network parameters. The attention score \( e_t \) depends on the current input and historical hidden state. After \( e_t \) is normalized by the SoftMax() function, the attention weight coefficient \( a_t \) is obtained. In the time attention stage, this paper only aims to obtain the weight of attention at different moments, so it does not carry out the weighted summation operation. The updated time series can be expressed as

\[
\tilde{x} = (a_1 x_1, a_2 x_2, \ldots, a_T x_T),
\]

Feature attention assigns different weights to the latent space representation of time series to highlight important features. The process is shown in Figure 3. At this stage, the attention mechanism is used to optimize the latent variables of LSTM-VAE. First, updated time series \( \tilde{x} \) are input to the LSTM-based encoder, which estimates the mean \( \mu_z \) and variance \( \sigma_z \) of latent variables \( z \) by two linear modules to approximate the posterior \( q_y(z|x) \). Then, the reparameterization operation is carried out to obtain latent variables \( z \); that is, \( z = \mu_z + \sigma_z \odot \epsilon \), where \( \epsilon \sim N(0, 1) \), and \( \odot \) represents the product at the element level. The latent variable \( z \) is transformed into the context vector \( c \) by the feature attention. Finally, the LSTM-based decoder outputs reconstructed time series \( y \) through the generation network \( p_\theta(y_t|c_i) \).

Figure 3. The attention mechanism optimizes latent variables of LSTM-VAE.
Different from the first stage, the attention mechanism at this time is based on the previous hidden layer state $d_{t-1}$ and unit state $C'_{t-1}$ of the decoder. The calculation process is as follows:

$$I_t = \sigma^T \tanh(W_t[d_{t-1}; C'_{t-1}] + U_t z + b_t),$$

$$\beta_i = \frac{\exp(I_i)}{\sum_{j=1}^{I} \exp(I_j)}$$

Among them, $[\ast; \ast]$ is an aggregation operation. $\sigma_t$, $W_t$, $U_t$, and $b_t$ are learnable network parameters. The attention score $I_t$ depends on current input $z_t$ and historical hidden states $d_{t-1}$ and $C'_{t-1}$ of the decoder. After $I_t$ is normalized by the SoftMax() function, the attention weight coefficient $\beta_i$ is obtained. It expresses the importance of latent variable $z_i$ to the output result so that important features are taken into consideration when outputting. The context vector $c_t$ is the weighted sum of all latent variables $\{z_1, z_2, \ldots, z_l\}$. The calculation formula is as follows:

$$c_t = \sum_{i=1}^{I} \beta_i z_i$$

After the context vector $c_t$ is obtained, the conditional generation distribution can be calculated according to $p_\theta(\tilde{x}_t|c_t)$. If its log likelihood $\log p_\theta(\tilde{x}_t|c_t)$ in the conditional generation distribution is larger, the effect of reconstruction is better.

### 3.3.2. Loss Function

In order to solve the problem of overfitting in AE [36] training, VAE adds $KL$ loss to the loss function. It measures the distance between approximate posterior distribution $q_\phi(z_t|x_t)$ and true posterior distribution $p_\theta(z_t)$. The objective of VAE’s loss function optimization is to maximize the likelihood function of the generated data and minimize the $KL$ divergence of the approximate posterior distribution to the true posterior distribution. Since this paper uses the attention mechanism to measure the importance of latent variable $z_t$, the context vector $c_t$ is obtained. Therefore, the improved loss function is shown in Formula (16).

$$L_{vae} = \lambda_{KL} [D_{KL}(q_\phi(c_t|x_t) || p_\theta(c_t))] - E_{q_\phi(c_t|x_t)} \log p_\theta(x_t|c_t),$$

It can be seen that the loss function consists of two parts: reconstruction loss and $KL$ loss. However, when VAE and LSTM are trained together, the problem of $KL$ divergence disappears. This paper uses the $KL$ cost-annealing method proposed in [37]. This method multiplies the $KL$ term by a weight coefficient $\lambda_{KL}$. The coefficient size is 0 at the beginning of training and gradually increases to 1 as the number of training increases. Thus, the reconstruction term is paid attention to in the early stage of training.

### 3.4. Online Detecting

A trained model is used to determine whether the observed value of a time step is abnormal during online detecting. This paper uses normal data to train the model and extracts the normal pattern of data. When the data contain anomalies, they cannot be reconstructed well; that is, it has a low reconstruction probability. Then, reconstruction probability is used as anomaly scores. The higher scores, the more likely that it is an abnormality. Finally, adaptive threshold is applied to anomaly scores to obtain the anomaly detection results.

### 3.4.1. Calculation of Reconstruction Probability

This paper adopts the method proposed in [12], using reconstruction probability as anomaly scores. The reconstruction probability is calculated by random hidden variables that can output the parameters of the distribution of original input variables rather than the input data. Therefore, it is more principled and objective than reconstructing errors. It
uses Monte Carlo to estimate the reconstruction item of VAE. The calculation formula is as follows:

\[
E_{q(c_t|\tilde{x}_t)}[\log p_\theta(\tilde{x}_t|c_t)] \approx \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\tilde{x}_t|\mu_c(t, l), \sigma_c(t, l)),
\]

(17)

The calculation process can be described as: First, the original data are passed through the encoder to obtain the distribution \(\mu_z\) and \(\sigma_z\) of data, and latent variable \(z\) is obtained by sampling in the distribution. The latent variables \(z\) are transformed into context vectors \(c\) by the attention layer. The decoder outputs parameters \(\mu_c\) and \(\sigma_c\) of context vector \(c\). These parameters are used to calculate the probability of generating original data from the distribution. For each data \(c(t, l)\) in the extracted sample \(L\), loop about \(l\). After the loop is over, multiple \(\mu_c(t, l), \sigma_c(t, l)\) can be obtained, that is, multiple \(c\). Finally, the log likelihood of the input data \(\tilde{x}_t\) is calculated on the approximate posterior distribution of \(c\), and the average reconstruction probability on all samples \(c\) is calculated.

3.4.2. Calculation of Anomaly Score

Anomaly scores are negative reconstruction probabilities. If the sample has a lower reconstruction probability, it will have a higher anomaly score. The reconstruction probability \(E_{q(c|\tilde{x}_t)}[\log p_\theta(\tilde{x}_t|c_t)]\) can be calculated immediately at time \(t\). By taking the additive inverse of reconstruction probability, anomaly scores can be obtained. The calculation formula is as follows:

\[
S_t = -E_{q(c|\tilde{x}_t)}[\log p_\theta(\tilde{x}_t|c_t)],
\]

(18)

3.4.3. Adaptive Threshold

After calculating anomaly scores, it is often necessary to set a threshold to determine whether it is an anomaly. Adjusting different values will change the detection rate and false-alarm rate accordingly. The raw anomaly scores represent an instantaneous measure of the predictability of current input data, which is suitable for predictable scenes. However, in many practical applications, the underlying system is inherently noisy and unpredictable. For example, the latency indicator of a website is usually low, but occasionally, random jumps and corresponding peaks in anomaly scores are not uncommon. Anomalies usually occur continuously in actual application scenes. However, setting a fixed threshold for original anomaly scores can lead to many false positives and false negatives. Therefore, this paper proposes an adaptive threshold method to reduce false alarms as much as possible. The adaptive threshold can be defined as:

\[
[S_t - \sigma, S_t + \sigma],
\]

(19)

Among them, \(\sigma\) is the floating range of anomaly scores \(S_t\). Considering the time relevance of anomaly scores, we calculate the possible fluctuation range of anomaly scores at the current moment based on historical anomaly scores [38]. This paper uses standard deviation \(\sigma\) to describe the range of anomaly scores, which is defined as follows:

\[
\text{avg} = \frac{1}{w} \sum_{i=t-w+1}^{t} S_i,
\]

(20)

\[
V = \sqrt{\frac{1}{w} \sum_{i=t-w+1}^{t} \left( \frac{S_i - \text{avg}}{w} \right)^2},
\]

(21)

\[
\sigma = \rho \cdot V,
\]

(22)

Among them, \(w\) is the size of the sliding window. \(\text{avg}\) is the average of historical anomaly scores. \(V\) is the standard deviation. \(\rho\) is the standard deviation coefficient. Therefore, anomaly is determined based on whether anomaly score at the current moment is within normal fluctuation range:
(1) Normal: $S_t - \sigma \leq S_t \leq S_t + \sigma$;
(2) Abnormal: $S_t < S_t - \sigma$ or $S_t > S_t + \sigma$.

4. Experiment

This section verifies the validity of this method by comparing it with different methods in a public dataset. In the environment of Python 3.6, we used Keras to implement the method in this paper and used Tensorflow as the backend engine. The hyperparameters in the experiment are set as follows: the time step size is 60, the hidden layer size of LSTM is 128, the latent variable dimension is 10, the batch size is 256, the number of iterations is 300, Adam is selected as the optimizer, and the initial value of learning rate is 0.0005.

4.1. Dataset

This paper uses the dataset of AIOPS Challenge held by Tsinghua University in 2018 (http://iops.ai/competition_detail/?competition_id=5&flag=1 (accessed on 10 September 2021)). According to the organizer’s description, the data are collected from the real operation and maintenance environment of top internet companies such as Sogou, Tencent, eBay, Baidu, and Alibaba. There are 28 KPIs in the dataset, including service KPIs that can reflect the scale and quality of web services, such as web response time, web page visits, and connection errors. It also includes machine KPIs that can reflect the health status of machines (servers, routers, switches), such as CPU usage, memory usage, disk IO, network card throughput, etc. We randomly selected three KPIs from them to conduct experiments. The details of the dataset are shown in Table 2. The data sampling interval is one minute, and sampling time is 91 days. It can be found that the positive and negative ratios of data are extremely unbalanced, and anomaly ratios are both less than 10%. Figure 4 shows the visualization of three KPIs. It can be seen that the curves of KPI 1 and KPI 2 exhibit a certain periodicity, while the curve of KPI 3 is stable and non-periodic.

| Table 2. Description of three KPIs in dataset. |
|-------------------------------|------------------|------------------|------------------|
| Data Set                      | KPI 1            | KPI 2            | KPI 3            |
| Total points                  | 128,562          | 129,035          | 129,128          |
| Anomaly points                | 10,550/8.21%     | 7666/5.94%       | 7863/6.09%       |
| Missing points                | 3233/0.02%       | 2735/0.02%       | 2667/0.02%       |
| Duration                      | 91 days          | 91 days          | 91 days          |
| Sample frequency              | 1412.77          | 1417.97          | 1418.99          |

Figure 4. Three KPIs from real production environments (Blue represents normal and red represents abnormal).
4.2. Baseline and Evaluation Metrics

In order to evaluate the performance of the method in this paper, we implemented three baseline methods. Three baselines that also use VAE as the basic structure were selected for comparative experiments. The hyperparameters of all methods are the same in the experiment:

- **VAE [14]**: Donut is an unsupervised anomaly detection method based on VAE. Through the improved variational lower bound and Markov chain Monte Carlo interpolation technology, donut can be used without labels. During anomaly judgment, it scores anomalies on the data through the reconstruction probability and then determines anomalies according to the set threshold value;

- **LSTM-VAE [21]**: This method proposes a long–short-term memory-based variational autoencoder (LSTM-VAE), which can extract the time dependence of sequence data and achieve better performance than donut in KPI anomaly detection with partial time dependence. In addition, for the problem that sometimes non-anomalous has a higher anomaly score, this method introduces a state-based threshold, which can reduce false alarms and increase sensitivity by changing anomaly scores;

- **LSTM-VAE-Attention [30]**: This method introduces an attention mechanism based on a variational recurrent autoencoder. It uses a variational self-attention mechanism (VSAM) to improve the performance of encoding–decoding processes. Finally, the reconstruction probability is used as anomaly scores to detect anomaly patterns.

This paper uses **Precision**, **Recall**, and **F1-score** as evaluation metrics. **Precision** represents the proportion of correct prediction being positive to total prediction being positive. **Recall** represents the proportion of correct prediction being positive to total actual being positive. **F1-score** is the weighted harmonic average of **Precision** and **Recall**. The calculation formula is as follows:

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad (23)
\]

\[
\text{Recall} = \frac{TP}{TP + FN}, \quad (24)
\]

\[
\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (25)
\]

where **TP** is the number of anomaly points correctly detected, **FP** is the number of normal points incorrectly identified as anomaly points, and **FN** is the number of anomaly points incorrectly identified as normal points.

4.3. Experimental Results and Comparative Analysis

This paper uses a 10-fold method to test the effectiveness of four methods. The best **Precision**, **Recall**, and **F1-score** of four methods are shown in Table 3. It can be seen that four methods perform well in periodic KPI 1 and KPI 2. However, VAE performs poorly in non-cyclical KPI 3. LSTM-VAE takes into account the time correlation of sequence, so it has higher accuracy than VAE. LSTM-VAE-Attention uses attention mechanism to pay attention to important latent variables, so the effect is slightly improved than LSTM-VAE. DA-LSTM-VAE in this paper obtained an **F1-score** in the range of 0.85–0.95, which is better than other methods. We use the data histogram to visualize data in the table, as shown in Figure 5.

Before time series are entered into the encoder, different weights are assigned to different time points in time step using the attention mechanism. Similar patterns can be observed for other time steps. Figure 6a shows the weight of attention at each time point in the training set. Figure 6b shows the weight of attention at each time point in the test set.

It can be observed that since the training set does not contain anomalies, the attention weights fluctuate within a stable interval. Time points in the test set that contain anomalies are assigned a larger weight, which leads to significant fluctuations.
Table 3. Anomaly detection results of each method in three KPIs.

| Dataset | Method               | Precision | Recall | F1-Score |
|---------|----------------------|-----------|--------|----------|
| KPI 1   | VAE                  | 0.83      | 0.76   | 0.79     |
|         | LSTM-VAE             | 0.84      | 0.80   | 0.81     |
|         | LSTM-VAE-Attention   | 0.90      | 0.85   | 0.87     |
|         | DA-LSTM-VAE          | 0.92      | 0.89   | 0.90     |
| KPI 2   | VAE                  | 0.88      | 0.82   | 0.84     |
|         | LSTM-VAE             | 0.91      | 0.84   | 0.87     |
|         | LSTM-VAE-Attention   | 0.93      | 0.89   | 0.90     |
|         | DA-LSTM-VAE          | 0.98      | 0.89   | 0.93     |
| KPI 3   | VAE                  | 0.78      | 0.70   | 0.74     |
|         | LSTM-VAE             | 0.84      | 0.78   | 0.80     |
|         | LSTM-VAE-Attention   | 0.87      | 0.82   | 0.84     |
|         | DA-LSTM-VAE          | 0.89      | 0.85   | 0.86     |

Figure 5. Data visualization effect in Table 3.

Figure 6. The attention mechanism assigns different weights to different moments in time series.

DA-LSTM-VAE uses normal data for training, thereby learning the normal pattern of the data. In this way, the model can reconstruct normal data sequences well. In order to prove that our model can effectively improve the encoding–decoding process and enhance the reconstruction effect, as shown in Figure 7, the reconstruction effect of each method on a time step of a normal time series was compared.
During anomaly judgment, this paper uses reconstruction probability as anomaly probability. Thus, it is possible to detect anomalies.

When the time series contains anomalies, the anomaly series will not be reconstructed well. In order to observe this phenomenon intuitively, Figure 8 shows the comparison between time series with anomalies and reconstructed time series.

![Figure 8. Reconstruction effect of anomaly time series.](image)

It can be observed from Figure 8 that the anomaly interval of the time series cannot be reconstructed well, so it has a higher reconstruction error and a lower reconstruction probability. Thus, it is possible to detect anomalies.

During anomaly judgment, this paper uses reconstruction probability as anomaly scores. Setting a fixed threshold directly on anomaly scores will not only lead to a large number of false positives and false negatives but also need to adjust the threshold manually. Therefore, this paper proposes an adaptive threshold method for the above problems, which judges the possible fluctuation range of anomaly scores at the current moment based on historical anomaly scores. The effect of anomaly judgment on anomaly scores through the adaptive threshold is shown in Figure 9.

It can be seen from Figure 9 that if anomaly scores exceed the upper or lower bound of threshold, they will be judged as anomaly.

In this paper, LSTM is used as the encoder and decoder of VAE, which can better capture the time correlation of sequence data. In the experiment, the test set containing
some anomalies was selected to verify the effect of time correlation. The hour of the time series was extracted as the label; that is, there are 24 types of labels. Principal component analysis (PCA) [39] and t-distributed stochastic neighbor embedding (t-SNE) [40] are used to reduce the dimension of latent variables to two dimensions for visualization. Figure 10a shows the visualization of PCA. Figure 10b shows the visualization of t-SNE.

The visualization effect shows that the time series present the phenomenon of color gradients in latent variables space. The model maps the sequence aligned in time to the same region of latent variable space. In other words, the model captures the temporal correlation of sequence. However, anomalous moments present a large deviation, such as 8 o’clock.

To highlight the advantages of introducing attention mechanism, we also selected part of the test set. T-SNE is used to visualize latent variable z and context vector c, respectively. Figure 11a shows the two-dimensional representation of latent variable z. It can be found that the difference between normal data and abnormal data in latent variable space is not very obvious, and there is more overlap. Figure 11b shows the two-dimensional representation of context vector c. It can be found that normal data and abnormal data are well distinguished. This demonstrates the use of attention mechanism to optimize latent variables, and a better potential representation is obtained.
In this paper, we proposed a novel dual-stage attention-based LSTM-VAE (DA-LSTM-VAE) for KPI anomaly detection. In this method, LSTM is used as the encoder and decoder of VAE to capture the time dependence and correlation characteristics of time series. Dual-stage attention consists of time attention in the input stage of encoder and feature attention in the input stage of the decoder. Time attention weights original time series to highlight important time points. Feature attention optimizes the latent variable representations of LSTM-VAE to highlight important features. Through the above improvements, our method can improve the encoding and decoding process, better capture the long-term dependence of time series, and learn better potential representations. In addition, this paper proposes an adaptive threshold method, which can greatly reduce false positives and false negatives and guarantee the accuracy of KPI anomaly detection. Experiments based on KPI time series data show that the KPI anomaly detection method proposed in this paper has a better detection effect than baseline methods. If the method in this paper is applied to the real environment, it includes two steps of offline training and online detection. The offline training step is to learn the normal mode of data, and the online detection step is to detect anomalies according to the trained model. By monitoring the KPI time series in real time, abnormal time points can be automatically discovered so that operations engineers can handle them in time and ensure the smooth operation of the application.

**Author Contributions:** Y.Z., conceptualization, methodology, investigation, validation, formal analysis, writing—original draft, and writing—review and editing; X.Z., supervision, validation, formal analysis, and writing—review and editing; Z.C., supervision, validation, formal analysis, and writing—review and editing; Z.S., supervision, validation, formal analysis, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (Grant No. 52231014).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request.

**Acknowledgments:** The authors would like to thank all anonymous reviewers and editors for their helpful suggestions for the improvement of this paper.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could appear to influence the work reported in this paper.

---

**Figure 11.** Two-dimensional visualization of latent variable $z$ and context vector $c$.

5. Conclusions
References

1. Pei, D.; Zhang, S.; Pei, C. Intelligent operation and maintenance based on machine learning. *Commun. CCF* 2017, 13, 68–72.

2. He, S.; Yang, B.; Qiao, Q. Overview of Key Performance Indicator Anomaly Detection. In Proceedings of the IEEE Region 10 Symposium (TENSYMP), Jeju, Republic of Korea, 23–25 August 2021.

3. Ceschini, G.; Gatta, N.; Venturini, M.; Hubauer, T.; Murarasu, A. Optimization of statistical methodologies for anomaly detection in gas turbine dynamic time series. *J. Eng. Gas Turbines Power* 2018, 140, 032401. [CrossRef]

4. Reis, B.Y.; Mandl, K.D. Time series modeling for syndromic surveillance. *BMC Med. Inform. Decis. Mak.* 2003, 3, 2. [CrossRef] [PubMed]

5. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long Short Term Memory Networks for Anomaly Detection in Time Series. In Proceedings of the 23rd European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 22–24 April 2015; pp. 89–94.

6. Ghanbari, M.; Kinsner, W.; Ferens, K. Anomaly detection in a smart grid using wavelet transform, variance fractal dimension and an artificial neural network. In *Proceedings of the IEEE Electrical Power and Energy Conference (EPEC)*, Ottawa, ON, Canada, 12–14 October 2016; pp. 1–6.

7. Laptev, N.; Amizadeh, S.; Flint, I. Generic and Scalable Framework for Automated Time-series Anomaly Detection. In Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Sydney, Australia, 10–13 August 2015; pp. 1939–1947.

8. Liu, D.; Zhao, Y.; Xu, H.; Sun, Y.; Pei, D.; Luo, J.; Jing, X.; Feng, M. Apprentice: Towards practical and automatic anomaly detection through machine learning. In *Proceedings of the ACM Measurement Conference*, Tokyo, Japan, 28–30 October 2015; pp. 211–224.

9. Shi, J.; He, G.; Liu, X. Anomaly Detection for Key Performance Indicators through Machine Learning. In Proceedings of the 6th IEEE International Conference on Network Infrastructure and Digital Content (NICD), Guiyang, China, 22–24 August 2018.

10. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* 2016, 58, 121–134. [CrossRef]

11. Yang, X.; Latecki, L.J.; Pokrajac, D. Outlier Detection with Globally Optimal Exemplar-Based GMM. In Proceedings of the Siam International Conference on Data Mining (SDM), Sparks, NV, USA, 30 April–2 May 2009; pp. 145–154.

12. An, J.; Cho, S. Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* 2015, 2, 1–18.

13. Kawachi, Y.; Koizumi, Y.; Harada, N. Complementary Set Variational Autoencoder for Supervised Anomaly Detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 15–20 April 2018; pp. 2366–2370.

14. Xu, H.; Feng, Y.; Chen, J.; Wang, Z.; Qiao, H.; Chen, W. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In *Proceedings of the 27th World Wide Web (WWW) Conference*, Lyon, France, 23–27 April 2018; pp. 187–196.

15. Li, Z.; Chen, W.; Pei, D. Robust and Unsupervised KPI Anomaly Detection Based on Conditional Variational Autoencoder. In *Proceedings of the 37th International Performance Computing and Communications Conference (IPCCC)*, Orlando, FL, USA, 17–19 November 2018; pp. 1–9.

16. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, 14–16 April 2014.

17. Li, Z.; Zhao, Y.; Geng, Y.; Zhao, Z.; Wang, H.; Chen, W.; Jiang, H.; Vaidya, A.; Su, L.; Pei, D. Situation-Aware Multivariate Time Series Anomaly Detection Through Active Learning and Contrast VAE-Based Models in Large Distributed Systems. *IEEE J. Sel. Areas Commun.* 2022, 40, 2746–2765. [CrossRef]

18. Zhang, S.; Zhong, Z.; Li, D.; Fan, Q.; Sun, Y.; Zhu, M.; Zhang, Y.; Pei, D.; Sun, J.; Liu, Y.; et al. Efficient KPI Anomaly Detection Through Transfer Learning for Large-Scale Web Services. *IEEE J. Sel. Areas Commun.* 2022, 40, 2440–2445. [CrossRef]

19. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.

20. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Shroff, G. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. Presented at ICML 2016 Anomaly Detection Workshop, New York, NY, USA, 24 June 2016.

21. Daehyun, P.; Hoshi, Y.; Kemp, C.C. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder. *IEEE Robot. Autom. Lett.* 2017, 3, 1544–1551.

22. Lin, S.; Clark, R.; Birke, R.; Schonborn, S.; Roberts, S. Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 4–8 May 2020; pp. 4322–4326.

23. Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Pei, D. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th International Conference on Knowledge Discovery Data Mining (KDD)*, Anchorage, AK, USA, 4–8 August 2019; pp. 2828–2837.

24. Niu, Z.; Yu, K.; Wu, X. LSTM-Based VAE-GAN for Time-Series Anomaly Detection. *Sensors* 2020, 20, 3738. [CrossRef] [PubMed]

25. Chen, R.; Shi, G.; Zhao, W.; Liang, C. A joint model for anomaly detection and trend prediction on it operation series. *Neurocomputing* 2021, 448, 130–139. [CrossRef]

26. Chen, N.; Tu, H.; Duan, X.; Hu, L.; Guo, C. Semisupervised anomaly detection of multivariate time series based on a variational autoencoder. *Appl. Intell.* 2022. [CrossRef]
27. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
28. Vaswani, A.; Shazeer, N.; Parmar, N. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
29. Zhang, C.; Song, D.; Chen, Y. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence/31st Innovative Applications of Artificial Intelligence Conference/9th AAAI Symposium on Educational Advances in Artificial Intelligence, Hawaii, HI, USA, 27 January–1 February 2018; pp. 1409–1416.
30. Pereira, J.; Silveira, M. Unsupervised Anomaly Detection in Energy Time Series Data Using Variational Recurrent Autoencoders with Attention. In Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1275–1282.
31. Yao, Q.; Song, D.; Chen, H. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 19–25 August 2017; pp. 2627–2633.
32. Rosenthal, J.S. Using Markov Chain Monte Carlo. *J. Am. Stat. Assoc.* 1995, 90, 558–566. [CrossRef]
33. Hübner, R.; Steinhauser, M.; Lehle, C. A Dual-Stage Two-Phase Model of Selective Attention. *Psychol. Rev.* 2010, 117, 759–784. [CrossRef] [PubMed]
34. Liu, Y.; Gong, C.; Yang, L. DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Syst. Appl.* 2020, 143, 113082. [CrossRef]
35. Hochreiter, S.; Jürgen, S. Long Short-Term Memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef] [PubMed]
36. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of data with neural networks. *Science* 2006, 313, 504–507. [CrossRef] [PubMed]
37. Bowman, S.R.; Vilnis, L.; Vinyals, O. Generating Sentences from a Continuous Space. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp. 10–21.
38. Deng, M.; Wu, B. Self-adaptive Threshold Traffic Anomaly Detection Based on ϕ-Entropy and the Improved EWMA Model. In Proceedings of the 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), ELECTR NETWORK, Chongqing, China, 12–14 June 2020; pp. 725–730.
39. Bro, R.; Smilde, A.K. Principal Component Analysis. *J. Mark. Res.* 2014, 6, 2812–2831. [CrossRef]
40. Laurens, V.D.M.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* 2008, 9, 2579–2605.