ISSN 2616-5775 Vol. 8, Issue 6: 27-35, DOI: 10.25236/AJCIS.2025.080604

Time Series Anomaly Detection Based on Hreg-VAE-LSTM

Tianbo Xu[a,*], Qian Wang[b]

Wuhu Institute of Technology, Wuhu, Anhui, 241003, China
[a]101394@whit.edu.cn, [b]2598961564@qq.com
**Corresponding author*

Abstract: *In the Industry 4.0, smart factories often face data anomalies during the collection and transmission of industrial time series data. To overcome this limitation, we propose a time series anomaly detection model based on Hreg-VAE-LSTM. The model leverages a Variational Autoencoder (VAE) module to capture local features within short time windows and employs the Hreg regularization method to mitigate the issue of data imbalance. Subsequently, a Long Short-Term Memory (LSTM) network is used to model the long-term dependencies in the sequence based on the features extracted by the VAE. This design enables the proposed algorithm to effectively detect anomalies across multiple temporal scales. Extensive experiments conducted on five real-world industrial datasets proved our model demonstrate the effectiveness and superiority of our model.*

Keywords: *Unsupervised Learning, Anomaly Detection, Time Series, Deep Learning*

1. Introduction

In the Industry 4.0, the intelligent transformation of traditional industries presents numerous challenges[1]. Given that industrial data is often represented as time series, anomaly detection plays a critical role in the process of data acquisition and analysis[2]. In many industrial scenarios, anomaly detection plays a key role in identifying sensor malfunctions[3], issuing alerts for external cyberattacks and enabling the early detection of potentially catastrophic events. Despite its significance, designing an effective anomaly detection algorithm remains highly challenging. This is primarily due to the inherent imbalance in training data, where labeled anomalous samples are scarce[4]. Furthermore, most anomalous behaviors are unknown prior to deployment, requiring detection algorithms to identify previously unseen anomalies. Consequently, anomaly detection models are often trained in an unsupervised model[5].

In this paper, we proposed a anomaly detection model that integrates the representational strength of a deep model—specifically, a Variational Autoencoder (VAE)—with the time modeling capabilities of a Long Short-Term Memory (LSTM) network. The VAE module captures structural patterns within local windows of the time series and addresses the data imbalance problem using a Hreg regularization strategy.Meanwhile, the LSTM module models long-term dependencies across the sequence. Notably, both the VAE and LSTM components operate in an unsupervised manner, eliminating the need for labeled anomalies during training.In summary, the main contributions of this work are as follows:

⬩ We employ a Variational Autoencoder (VAE) to extract local structural information from short time windows and encode it into low-dimensional embeddings.

⬩ To address the issue of data imbalance, we incorporate the Hreg regularization method, which improves the robustness of the model in detecting rare anomalies.

⬩ By leveraging the LSTMs ability to capture both short-term and long-term dependencies, our model effectively detects anomalies occurring over varying temporal scales, as validated by experimental results.

2. Related Work

Currently, a variety of algorithms have been developed for time series anomaly detection, which can be broadly categorized into supervised and unsupervised approaches depending on whether labeled data is required[6]. Given that the application scenarios addressed in this study typically involve time series data without labeled anomalies, this research focuses on unsupervised methods for time series anomaly detection[7].

Anomaly detection is a fundamental task in machine learning, which focuses on identifying data instances that significantly deviate from established patterns or normal behavior. This process entails the recognition of events or observations that do not conform to expected statistical distributions or temporal trends. An anomaly, or outlier, is typically characterized as a data point that diverges markedly from the majority of observations, suggesting it may be generated by a distinct underlying process. In the context of time series analysis, anomalies refer to temporal points exhibiting behaviors that substantially differ from those observed in preceding intervals[8]. Anomaly detection techniques are widely applicable in numerous domains, including industrial quality control, medical diagnostics, vehicle monitoring, and human activity recognition. With continuous technological advancements, research in this field has increasingly focused on machine learning and deep learning-based approaches due to their capacity for automated feature extraction and improved detection performance[9].

Autoencoders (AEs), a category of unsupervised neural networks, have demonstrated strong capabilities in anomaly detection tasks[10]. Typically trained solely on normal data, AEs learn to reconstruct input sequences by capturing underlying patterns[11]. At the inference stage, inputs that deviate from the learned distribution—such as anomalous samples—tend to yield higher reconstruction errors, making these errors a reliable metric for anomaly detection[12]. The Variational Autoencoder (VAE)extends the standard autoencoder framework by incorporating a probabilistic generative model. Unlike deterministic autoencoders, VAEs learn a distribution over the latent space, enabling the generation of new samples that resemble the original data distribution. This generative nature enhances the model's utility in unsupervised anomaly detection, especially in cases where modeling uncertainty is important[13].

Training a VAE involves maximizing the Evidence Lower Bound (ELBO)[14], which comprises two main components: a reconstruction loss that ensures the model can accurately reproduce input data, and a regularization term that minimizes the Kullback-Leibler (KL)[15] divergence between the learned posterior distribution and a predefined prior—usually a standard normal distribution. This regularization encourages the formation of a continuous, well-structured latent space, which is crucial for generative tasks and anomaly scoring[16].

3. Methods

Given an industrial time series dataset X, which $X = \{x_1, x_2, \dots, x_N\}$ where $x_i \in \mathrm{R}^m$ represents multivariate data from m different channels at time step i, our model aims to perform anomaly detection on the sequence. For any time point t such that $L \le t \le N$, we utilize the preceding L time steps to construct a sliding window $S_i = \{x_{t-L+1}, x_{t-L+2}, \dots, x_t\}$. The model outputs a prediction $y_t \in (0,1)$, where $y_t = 1$ indicates an anomaly within window S_t, and $y_t = 0$ indicates normal behavior.

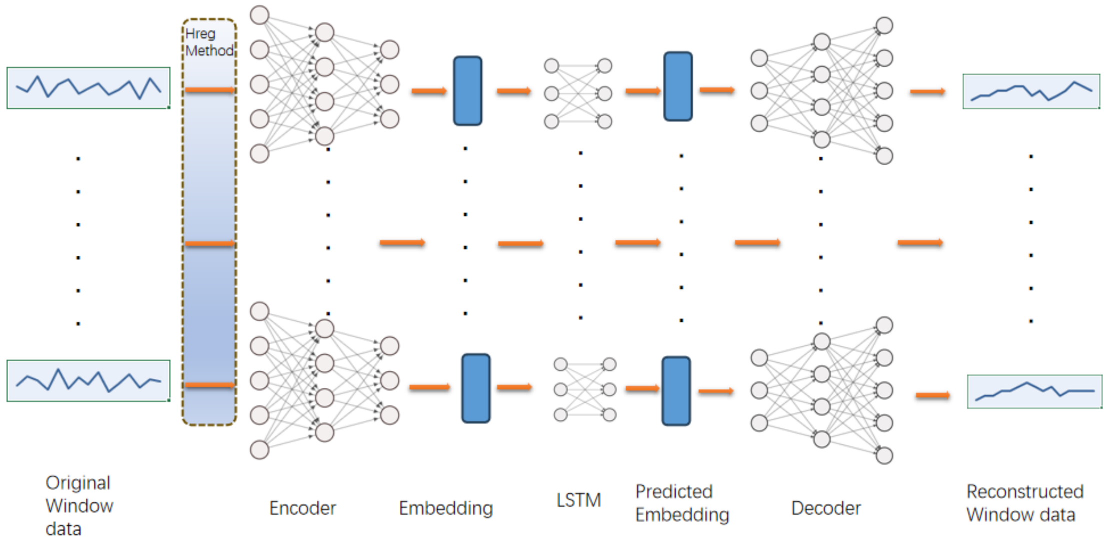


Figure 1 Hreg-VAE-LSTM model

An overview of the proposed model architecture for time series input and anomaly detection is illustrated in Figure 1. The model comprises two key components: a Variational Autoencoder (VAE) model for capturing features within each window, and a Long Short-Term Memory (LSTM) network for modeling long-term dependencies in the sequence. During the local feature extraction process, the Hreg regularization technique is applied within the VAE to mitigate the effects of data imbalance, thereby enhancing the quality of the learned representations. These balanced features are then fed into the LSTM module to infer temporal dynamics across the sequence. Together, the VAE and LSTM components enable robust detection of anomalies in multivariate time series data. Figure 1 provides a schematic overview of the proposed model.

3.1 Hreg regularization

During the feature extraction process from streaming data, parameter values often cannot be explicitly controlled. This lack of control may lead to arbitrary and inconsistent final outputs across different categories, thereby hindering meaningful comparisons between them. In particular, features with inherently high magnitudes in the input data may be disproportionately emphasized by the classification model, resulting in overly dominant parameter weights. Such imbalance can compromise the generalization ability of the model, especially in multi-class classification scenarios.

To address this issue, it is preferable to evaluate the classification output holistically, rather than processing each category independently. The Hreg regularization algorithm, as defined in Formula (1), is designed to mitigate this problem by introducing a regularization term that constrains the model's behavior during feature extraction. By doing so, it helps balance the influence of different features and ensures more stable and comparable outputs across categories. Formula (1) presents the formal definition of the Hreg regularization algorithm.

$$Hreg = x{\sim}\mathbb{P}_x{}^{\mathbb{E}}||\nabla_x f(x)||_2^n \tag{1}$$

In this context, $\nabla_x f(x)$ denotes the gradient of the model output with respect to the input features, which reflects the sensitivity or importance of each feature in determining the model's prediction. The symbol $\mathbb{E}$ represents the mathematical expectation, computed under two standard forms. Features with large input values are often assigned high importance by the classifier, which in turn leads to disproportionately large associated parameter weights. However, such features may not necessarily be crucial for identifying the correct class label. This misalignment can negatively impact the models overall performance and generalization.

To address this issue, we adopt the Hreg (High-importance Regularization) method, which introduces a regularization mechanism to balance the influence of features with different magnitudes. Specifically, Hreg imposes anL2-norm penalty that is stronger for features associated with high gradient values and weaker for those with low gradient values. This penalization helps suppress the undue influence of dominant but less relevant features, thereby improving the robustness and accuracy of the model.

3.2 Proposed Hreg-VAE-LSTM model

In this experiment, in order to train our model in an unsupervised manner, we first determined the relevant training set and test set. The training data we provided in the training set did not contain anomalies, and the remaining time series data were used as the data for our test set.

The VAE model is composed of an encoder-decoder architecture. It processes a local segment consisting of P consecutive time series readings, where the encoder maps the input to a low-dimensional latent representation of Q dimensions, and the decoder attempts to reconstruct the original input window from this latent space. To facilitate training, a series of rolling windows are extracted from the training dataset. Let $w_i = \{x_{t-p+1}, x_{t-p+2}, \dots, x_t\}$ denote the time window of length p ending at time step t. And the LSTM model operates over a sequence of embeddings generated by the VAE, which are derived from k non-overlapping windows. We define the sequence of such windows ending at time t as: $W_t = \{W_{t-(k-1)*p}, \dots, W_t\}$, where each $W_{t-(k-1)*p}$ represents a window of length p sampled at an interval of p time steps. The corresponding embedding sequence is denoted by $E_t = \{e_t^1, \dots, e_t^k\}$, where e_t^i is the embedding of the i-th window in W_t generated by the VAE.

For a training set containing N_{Train} time steps, we can extract approximately N_{train}-P windows to train the VAE module, and approximately N_{TRAIN}-P*K sequences of k non-overlapping windows to train the LSTM module. To validate model performance and mitigate overfitting, we randomly reserve 10% subset of the sequences from the training dataset is reserved for validation purposes. To ensure unbiased model evaluation, all windows and sequences within this validation subset are excluded from the training process.

In the remaining windows of the training set, we optimize the parameters of the VAE model by maximizing the Evidence Lower Bound (ELBO) loss, which balances the reconstruction loss and the Kullback–Leibler (KL) divergence. Each data window is passed through the VAE, and once the VAE model is well trained, we utilize its encoder to generate latent embeddings E_T for all windows in the training set.

To train the LSTM model, we construct sequences of embeddings from E_T. Specifically, the LSTM takes the first $k-1$ embeddings in each sequence as input and learns to predict the subsequent $k-1$ embeddings. This sequence-to-sequence prediction task enables the LSTM to capture the temporal dependencies in the embedding space inferred by the VAE, thereby facilitating the modeling of long-term trends and aiding in the detection of temporal anomalies.

$$[\hat{e}_t^2, \dots, \hat{e}_t^k] = LSTM([e_t^1, \dots, e_t^{k-1}]) \tag{2}$$

The parameters of the LSTM model are optimized by minimizing the prediction error between the predicted and actual embeddings. Specifically, the objective is to minimize the following loss function: $\min||\hat{e}_t^k - e_t^k||$, where $\hat{e}_t^k$ denotes the LSTM-predicted embedding for the k-th position in the sequence, and e_t^k is the corresponding ground truth embedding obtained from the VAE encoder. This loss encourages the LSTM to accurately model temporal dynamics within the latent space defined by the VAE.It is important to note that all parameters in both the VAE and LSTM components are optimized in an unsupervised manner—no anomaly labels are required during training.

After the training phase, our model can be deployed for real-time anomaly detection. At each time step t, the Hreg-VAE-LSTM model processes a time series segmentW_t, which consists of the previous k×p time points leading up to t. First, the encoder of the trained VAE module is used to compute the embedding sequence E_t from W_t. The first $k-1$ embeddings from E_t are then input into the LSTM module, which predicts the subsequent $k-1$ embeddings, denoted as: $[\hat{e}_t^2, \dots, \hat{e}_t^k]$. Finally, the decoder component of the VAE is used to reconstruct the input data from the predicted embeddings. This process is formally represented as:

$$\widehat{w}_{t-(k-i)*p} = Decoder(\hat{e}_t^i), \text{ i=2,…k.} \tag{3}$$

By utilizing the reconstructed windows, we can determine whether the input window W_t contains anomalies by computing an anomaly score denoted as d_t. This score reflects the discrepancy between the original and reconstructed data, and serves as the basis for anomaly detection. Specifically d_t is defined as follows:

$$d_t = \sum_{i=2}^{k} ||\widehat{w}_{t-(k-i)*p} - w_{t-(k-i)*p}||_2 \tag{4}$$

To detect anomalies, we define a threshold θ on the anomaly score function d_t. If the computed score exceeds this threshold, an anomaly alarm is triggered by assigning y_t=1, indicating that the current time window W_t is considered suspicious and may contain abnormal events.

In practice, the threshold θ should be selected based on a validation subset comprising both typical and anomalous instances, if such data is available. This enables more accurate calibration of the detection sensitivity and helps reduce both false positives and false negatives.Ultimately, the performance of the proposed model is evaluated on the validation or test set using standard anomaly detection metrics, such as the F1 score, precision or recall, depending on the evaluation criteria of the specific application domain.

4. Experiment

4.1 Datasets

We evaluated the performance of our proposed Hreg-VAE-LSTM algorithm on five real-world time series datasets. A brief description of each dataset is provided below:

Ambient Temperature (Office): This dataset records indoor environmental temperature readings collected from sensors in an office environment. Anomalies typically arise from HVAC system faults or external environmental influences.

AWS CPU Utilization: This dataset contains CPU usage metrics from Amazon Web Services (AWS) instances. Anomalies include workload spikes, unexpected traffic loads, or misconfigurations.

Amazon East Server Metrics: This dataset comprises server monitoring statistics (e.g., temperature, CPU, memory) from an Amazon East Coast data center. Anomalous behavior often indicates system faults or external attacks.

Industrial Machine Internal Temperature: This dataset includes internal temperature sensor readings from industrial machinery during normal and faulty operations. Anomalies may indicate overheating or early signs of equipment failure.

New York City Taxi Demand: This dataset records the number of taxi passengers over time in New York City. Anomalies are typically caused by holidays, weather events, or major public incidents.

Each time series was normalized prior to training, and sliding windows were applied to segment the data for Hreg-VAE- LSTM models. Ground truth anomaly labels provided with the datasets were used for evaluation purposes only and not for training, as our model operates in an unsupervised learning setting.

4.2 Evaluation Criteria

Our method was compared against three widely used time series anomaly detection algorithms: Variational Autoencoder (VAE), LSTM-based Anomaly Detection (LSTM-AD), and the Auto Regressive Moving Average model (ARMA). Table 1 presents the numerical results for all models, along with the length of the detection window used in each case.

Table 1 Experimental results (Precision, recall, F1 score and Win-l: detection window length).

Dataset	Method	Win-l	Prec	Recall	F1
Ambient temperature	Ours	168	0.812	1.0	0.895
	VAE	24	0.686	0.5	0.573
	ARMA	24	0.184	1.0	0.311
	LSTM-AD	24	1.0	0.5	0.666
Cpu utilization AWS	Ours	144	0.773	1.0	0.873
	VAE	24	0.348	0.5	0.410
	ARMA	24	0.234	1.0	0.380
	LSTM-AD	24	0.274	1.0	0.430
Cpu request EC2	Ours	192	0.979	1.0	0.990
	VAE	24	0.949	1.0	0.996
	ARMA	24	0.938	1.0	0.968
	LSTM-AD	24	1.0	0.436	0.608
Machine temperature	Ours	288	0.986	1.0	0.986
	VAE	24	0.211	1.0	0.207
	ARMA	24	0.142	1.0	0.248
	LSTM-AD	24	1.0	0.5	0.667
NYC taxi	Ours	168	0.994	1.0	0.997
	VAE	24	0.662	0.8	0.725
	ARMA	24	0.769	0.4	0.526
	LSTM-AD	24	1.0	0.2	0.333

We evaluated performance using three standard metrics: precision, recall, and F1 score. For consistency and fairness, the detection window length was kept the same across all methods for each dataset. Note that the overall time span covered by our model appears longer, which is attributed to its hierarchical architecture that enables detection of long-duration events spanning multiple time steps.

A notable challenge in evaluation arises from the fact that many anomalies occur at a single timestamp,while all detection methods operate on sliding windows. This discrepancy complicates the accurate computation of true positives, false positives, and false negatives. To address this, we adopted the evaluation strategy proposed by, which provides a simplified and consistent mechanism for aligning detected windows with timestamp-level anomaly labels. The formula for calculating accuracy is shown in Formula 5.

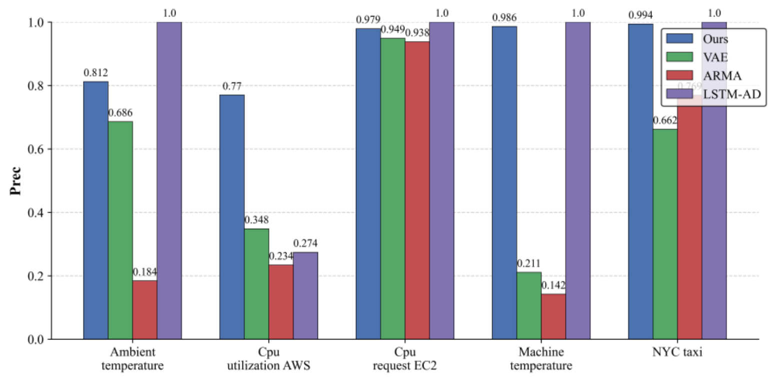
$$Pre = \frac{TP}{TP+FP} \tag{5}$$

The calculation formula for recall rate is shown in Formula 6.

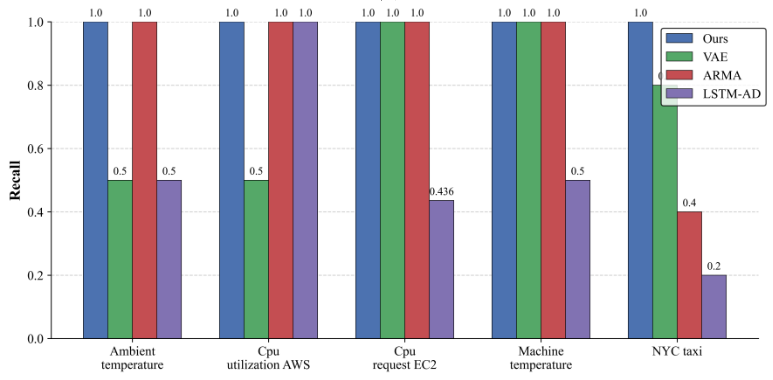
$$Rec = \frac{TP}{TP+FN} \tag{6}$$

In formulas 5 and 6, True Positive (TP) denotes the number of instances that are correctly identified as belonging to class A. False Positive (FP) refers to the number of instances incorrectly classified as class A, despite originating from other classes. False Negative (FN) represents the number of class A instances that were mistakenly identified as belonging to other categories. The F1 score, being the harmonic mean of precision and recall, offers a balanced evaluation by simultaneously accounting for both metrics. As a result, it is often regarded as a more reliable performance indicator than individual measures. The formula for computing the F1 score is presented in Formula 7.

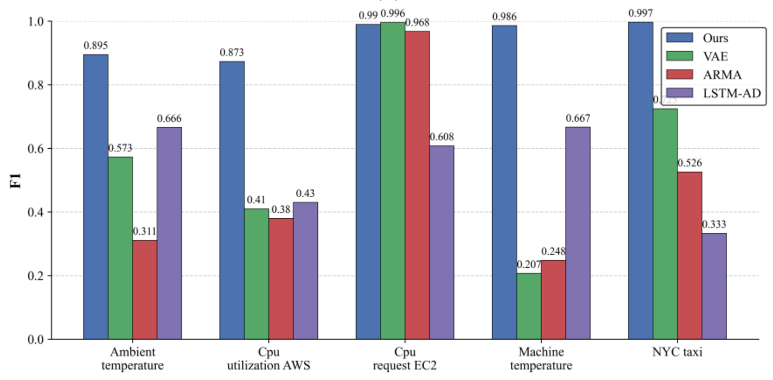
$$F1 = \frac{2*Pre*Rec}{Pre+Rec} \tag{7}$$



(a)



(b)



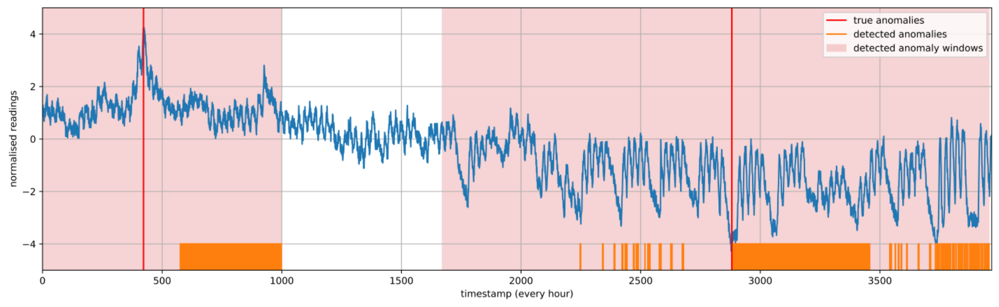
(c)

Figure 2 Comparison of models.

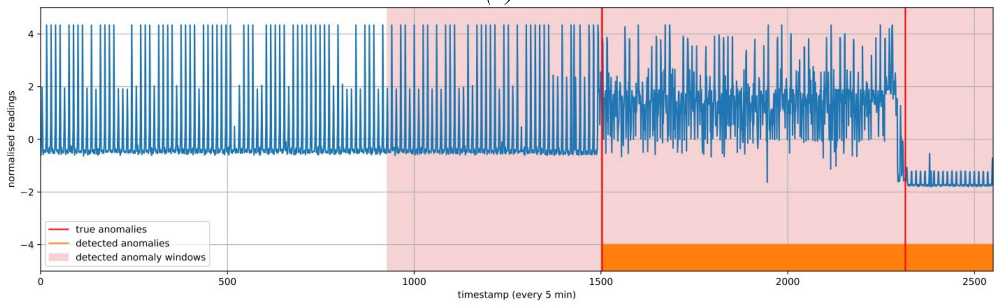
4.3 Experimental Results

Our proposed model performs best in five public datasets,The final experimental results are shown in Table 1.As shown in Table 1, our proposed Hreg-VAE-LSTM model has the highest F1 score in all five datasets. The Precision, recall, and F1 scores for each model are shown in Figure 2. In Figure 2, the performance of various anomaly detection models is compared across different evaluation metrics. The blue line represents the proposed Hreg-VAE-LSTM model, while the green, orange, and purple lines correspond to the VAE, ARMA, and LSTM-AD models, respectively. Figure 2(a) illustrates the precision of each method, and Figure 2(b) presents the recall scores. As shown in Figure 2(c), our proposed model achieves the highest F1 score among all evaluated approaches, demonstrating its superior overall detection performance.

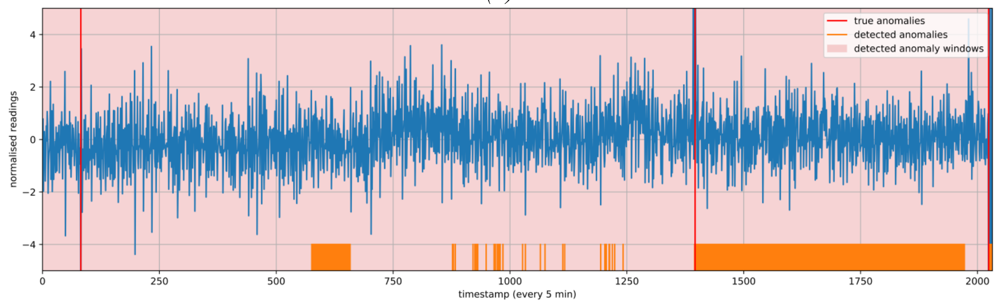
Figure 3 illustrates representative examples of anomaly detection results produced by our proposed model across five real-world time series datasets. In each subplot, the blue curve denotes the raw input data, red vertical lines indicate the ground truth anomaly points, orange vertical lines mark the anomalies detected by our model, and the red-shaded areas highlight the anomalous regions over time. Specifically, Figure 3(a) corresponds to ambient temperature data, Figure 3(b) shows the results for CPU utilization from AWS, Figure 3(c) presents detection results on the EC2 CPU request dataset, Figure 3(d) illustrates anomaly detection for industrial machine temperature, and Figure 3(e) shows results for the NYC taxi passenger dataset. Our model demonstrates the capability to identify anomalies with high accuracy, particularly by effectively capturing the contextual relationships surrounding anomalous points.



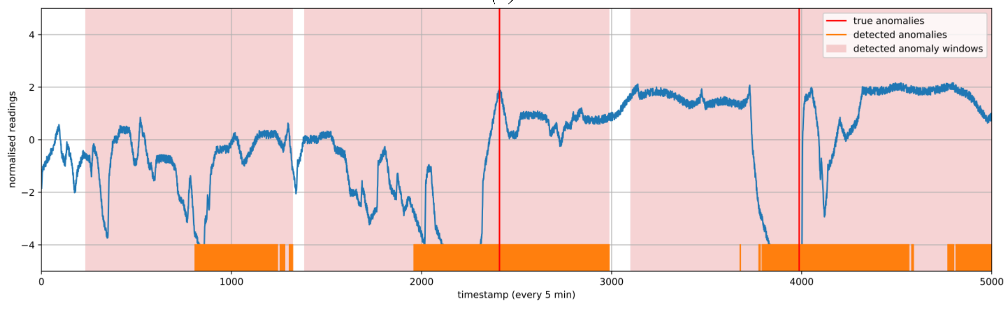
(a)



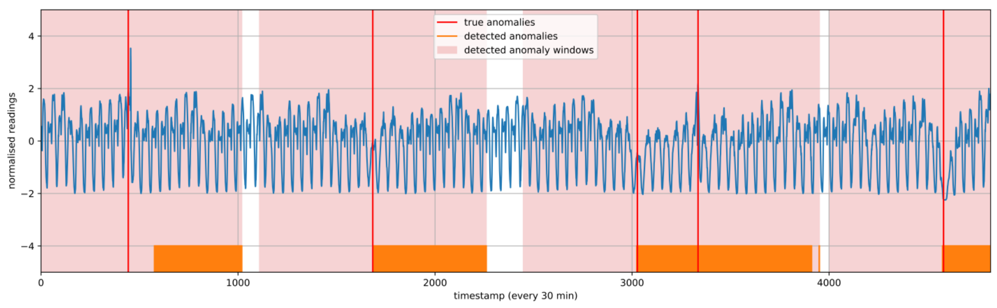
(b)



(c)



(d)



(e)

Figure 3 Anomalies detected by our model.

5. Conclusion

In this work, we proposed the Hreg-VAE-LSTM model to address key challenges in time series anomaly detection. As an unsupervised learning framework, our model integrates three components: a Variational Autoencoder (VAE) for extracting robust local features within short sliding windows, a novel Hreg regularization technique to mitigate the effects of imbalanced feature distributions, and a Long Short-Term Memory (LSTM) network to capture long-term dependencies across time. This design enables the model to detect anomalies occurring across multiple temporal scales. Extensive experiments conducted on five real-world datasets demonstrate the effectiveness and generalizability of our approach. The proposed model consistently outperforms several widely-used baseline methods, confirming its superiority in both detection accuracy and robustness.

Acknowledgements

This work is supported by natural science research projects of Wuhu Institute of Technology (wzyzr202424, wzyzr202423), Anhui Province universities natural science research project (2023AH052393, 2023AH052378, 2022AH052198) and Anhui Province Higher Education Quality Engineering Teaching Research Project (2021jyxm1695).

References

[1] Gong D, Liu L, Le V, et al. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1705-1714.
[2] Vanem E, Brandsæter A. Unsupervised anomaly detection based on clustering methods and sensor data on a marine diesel engine[J]. Journal of Marine Engineering & Technology, 2021, 20(4): 217-234.
[3] Sánchez-Fernández A, Baldan F J, Sainz-Palmero G I, et al. Fault detection based on time series modeling and multivariate statistical process control[J]. Chemometrics and Intelligent Laboratory Systems, 2018, 182:57-69.
[4] Li D, Chen D, Jin B, et al. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks[C]//International Conference on Artificial Neural Networks. Springer, Cham, 2019:703-716.
[5] Jiang W, Hong Y, Zhou B, et al. A GAN-based anomaly detection approach for imbalanced industrial time series[J]. IEEE Access, 2019, 7:143608-143619.
[6] Sun Y, Yu W, Chen Y, et al. Time series anomaly detection based on gan[C]//2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE, 2019: 375-382.
[7] Su Y, Zhao Y, Niu C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2828-2837.
[8] Zhang C, Song D, Chen Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 1409-1416.
[9] Ren H, Xu B, Wang Y, et al. Time-series anomaly detection service at microsoft[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 3009-3017.

[10] Naik G R, Kumar D K. Determining number of independent sources in undercomplete mixture[J]. EURASIP Journal on Advances in Signal Processing, 2009:1-5.
[11] Hou X, Zhang L. Saliency detection: A spectral residual approach[C]//2007 IEEE Conference on computer vision and pattern recognition. IEEE, 2007: 1-8.
[12] Wu Y, Dai H N, Tang H. Graph neural networks for anomaly detection in industrial internet of things[J]. IEEE Internet of Things Journal, 2021, 9(12): 9214-9231.
[13] Zhao H, Wang Y, Duan J, et al. Multivariate time-series anomaly detection via graph attention network[C]//2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020: 841-850.
[14] Zhang Y, Chen Y, Wang J, et al. Unsupervised deep anomaly detection for multi-sensor time-series signals[J]. IEEE Transactions on Knowledge and Data Engineering, 2021.
[15] Ding C, Sun S, Zhao J. MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection[J]. Information Fusion, 2023, 89: 527-536.
[16] Adepu S, Mathur A. Distributed attack detection in a water treatment plant: Method and case study[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 18(1): 86-99.