International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n2.19
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Investigating Semi-supervised Time Series Anomaly Detection Using AW-SNBAE Model

Mingyuan Guo

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China

ABSTRACT

In time series anomaly detection, most of the existing algorithms have difficulty in balancing the precision and dependence on labeled data. To solve this problem, a semi-supervised model called Adaptive Wavelet Sparse N-Beats AutoEncoder (AW-SNBAE) is proposed. This model has two main innovations. Firstly, an adaptive wavelet transform module is designed, which adaptively learns the wavelet function based on the global information of the time series, removes noise, and captures features of short-term and long-term changes. Secondly, the proposed Sparse N-Beats Auto Encoder (SNBAE) uses N-Beats as a base, and then, combines residual enhancement and sparse regularization strategies to construct the encoder and the decoder, respectively. Experimental results show that the AW-SNBAE model improves F1 scores by 8% in univariate time series anomaly detection and by 2% on average in multivariate detection. These results demonstrate that the model is able to exhibit excellent detection capability despite limited labeled data.

KEYWORDS

Anomaly detection; Time series analysis; Adaptive wavelet transform; N-Beats

1. INTRODUCTION

The core purpose of time series anomaly detection is to identify and label abnormal behavior patterns from time-varying data, so as to obtain valuable information for practical applications. Therefore, building an efficient anomaly detection algorithm has become the key. However, building an efficient anomaly detection algorithm often faces three challenges: the training data is unbalanced, and the normal samples are far more than the abnormal samples; The anomaly pattern is unknown, and the new data may contain abnormal types that have not appeared before. For large-scale labeled data, the cost of labeled samples is too high.

In addressing these challenges, anomaly detection algorithms based on supervised learning, such as autoregressive integral moving average (ARIMA [1]) and support vector machines (SVM [2]), which perform well in univariate time series, and algorithms such as graph convolutional neural networks (GCN [3]), which are suitable for multivariate time series, perform well in some cases. However, it is often difficult to capture unknown anomalies and requires a large amount of labeled data, so it is difficult to completely overcome the above challenges. Therefore, in order to deal with these challenges more effectively, unsupervised and semi-supervised learning methods become more optimal solutions.

Among them, unsupervised learning methods have been widely used in the field of anomaly detection. For example, Zong et al. [4] proposed a deep AutoEncoder model, which automatically learns the characteristics of normal data and constructs a reconstruction model, and uses the reconstruction error to identify unknown anomalies. Mao et al. [5] proposed a clustering-based anomaly detection algorithm (CBFS), which simplifies the feature subset while clustering the feature vectors, and uses a classifier based on decision tree to detect abnormal traffic on the simplified feature subset. In addition, Deng et al. [6] introduced an anomaly detection method based on graph neural network, which combines structure learning and graph neural network, captures the complex relationship and interaction between data through the attention mechanism, and can effectively locate anomalies in time series data. While these unsupervised methods provide flexible and powerful solutions, there are still challenges in clearly defining the boundary between abnormal and normal behavior. To this end, semi-supervised learning combines the advantages of supervised and unsupervised learning to further improve the accuracy of the model.

In the study of semi-supervised learning for time series anomaly detection, Pelati et al. [7] proposed a model combining semi-supervised method and Recurrent Neural Network (RNN) to learn the context of time series and extract key features in the data. Abnormal events were automatically identified and their duration was estimated through posterior analysis and real-time detection. Yu et al. [8] proposed a Mixture of Experts Convolutional Variational AutoEncoder Ensemble Model (MEx-CVAEC). This model combined two Convolutional Variational AutoEncoder (CVAE) and convolutional networks to learn the multi-manifold relationship of data, and used a gated network based on Linear AutoEncoder (CAE) to optimize the expert structure, and realized anomaly detection in the Encoder-Decode-Recode (EDE) pipeline. Kumagai et al. [9] proposed a method based on Graph Convolutional Network (GCN), which maps the nodes in the graph to the hidden space, and achieves anomaly detection by constructing a hypersphere to surround normal nodes and placing abnormal nodes outside the sphere. However, the above model design is relatively complex and computationally expensive when dealing with large-scale data. Therefore, the selection of a more basic and efficient model has become a more practical choice.

Although traditional AE [10] have efficiency advantages in anomaly detection tasks, they have obvious limitations in capturing long-term dependent features and multi-scale trends in complex time series. To overcome this limitation while ensuring the simplicity of this model, the N-Beats model can be used as the encoder and decoder of AE. The N-Beats model was proposed by Oreshkin et al. [11] for time series prediction tasks. Through neural basis expansion analysis, this model effectively captures and models long-term trends and seasonal components. With its parameter-free design and ability to directly predict future values, this model can flexibly deal with various complex time series signals. Therefore, by combining N-Beats with AE, it not only retains the efficiency and simplicity of the AE, but also greatly improves the precision and robustness of the model in complex time series anomaly detection.

In order to further optimize this above model, wavelet transform technology can be introduced in the data preprocessing stage [12]. In recent years, wavelet transform has been widely used in time series analysis by virtue of its multi-scale analysis ability, and it performs well in dealing with non-stationary signals [13]. However, fixed wavelet basis functions are often difficult to adapt to the complexity of different data characteristics, which may limit the performance of wavelet transform in anomaly detection. To solve this problem, in this paper, adaptive wavelet transform is used to achieve more effective noise removal and key information enhancement by dynamically adjusting the wavelet basis functions according to the data characteristics.

Based on the above analysis, this paper proposes the AW-SNBAE model based on semi-supervised learning, which consists of two core modules: an adaptive wavelet transform module and an anomaly detection module. In the adaptive wavelet transform module, the adaptive wavelet learning model (AWL) dynamically adjusts wavelet basis functions according to data characteristics to capture multi-scale features. In the anomaly detection module, the SNBAE model is proposed, combining sparse regularization and a residual enhancement strategy to effectively extract deep patterns and long-term dependencies in time series. This design not only improves the precision of anomaly detection but also avoids the high computational cost of complex models. The specific innovations of this model are as follows.

The Transformer [14] encoder is introduced to implement the adaptive wavelet transform module based on global sequence information. In this module, the AWL model maps the output of the Transformer encoder into a probability distribution of wavelet bases and series. By selecting the wavelet basis and series with the highest probability, the corresponding wavelet function is generated and used to perform the wavelet transform.

The SNBAE model is proposed to enable lightweight processing of N-Beats by omitting the nesting of blocks into stacks. This streamlined N-Beats is applied to the AE, with a residual-enhanced N-Beats structure used in the encoding phase to strengthen key feature extraction. In the decoding phase, a sparsity-regularized N-Beats structure is employed to optimize data reconstruction and select important features.

2. THE AW-SNBAE MODEL

The AW-SNBAE model consists of three modules: feature engineering module, adaptive wavelet transform module, and anomaly detection module. The three modules work together to form a complete anomaly detection process.

2.1. Feature Engineering Module

In the feature engineering module, for univariate time series data, Kalman filter is applied to smooth the data and reduce the influence of noise. At the same time, trend and seasonal analysis are carried out to reveal the internal structure and dynamic change law of the data, which provides key information for anomaly detection. Taking the KPI dataset as an example, Fig. 1 shows the specific results of the seasonal analysis of trends, where the residual component plot in Fig. 1 (d) can be used to visually identify outliers, noise, and other unexplained fluctuations in the data.

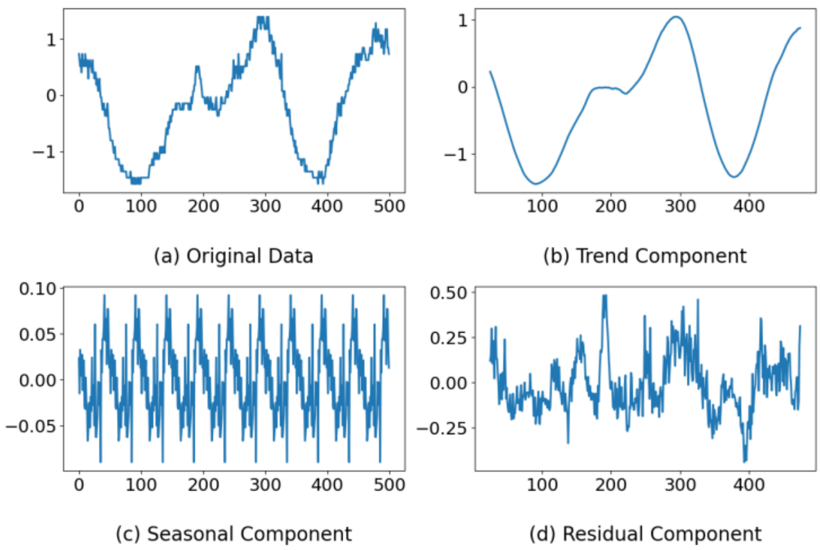


Figure 1. Trend seasonal analysis

In order to deal with the curse of dimensionality effectively when dealing with multivariate data, a feature optimization strategy is introduced. This strategy utilizes MI [15] to evaluate the interdependence between each feature and the anomaly, and screens out the features that contribute more to the anomaly detection task. For the features with small contribution, PCA [16] method is used to transform the original data into a new feature space, so as to extract the main components of the data and reduce the dimension, maintain the main information of the original data, and finally merge to generate a reduced feature set containing high-information feature fusion features.

2.2. Adaptive Wavelet Transform Module

2.2.1. The AWL model

By introducing the Transformer encoder, the AWL model uses its self-attention mechanism to comprehensively analyze the global information of the input sequence, and then realizes a new dynamic wavelet function selection process. The process is as follows.

Feature sequence coding process: Process each individual feature data by introducing Transformer encoder. In this process, Transformer uses its own self-attention mechanism to understand and analyze the sequence dependencies of the input data, capture the complex relationships between the various parts of the sequence, and finally obtain a comprehensive encoding of the input sequence features.

Adaptive wavelet learning: The sequence representation is obtained from the feature sequence coding process, and these representations are converted into the wavelet basis and its series scores through the fully connected layer. Sigmoid function is used to convert these scores into probability values, and the optimal wavelet function is learned by selecting the wavelet base and series with the highest probability value.

Optimization strategy for parallel processing: For multivariate time series, a parallel optimization strategy is proposed. Through the parallel deconstruction of feature dimensions, this strategy allows each feature to perform sequence coding and wavelet basis selection independently, which effectively improves the overall processing efficiency and ensures the processing quality.

Through the above steps, the AWL model improves the flexibility and efficiency of this model while also enhancing its performance when dealing with complex data. The AWL model structure is shown in Fig. 2.

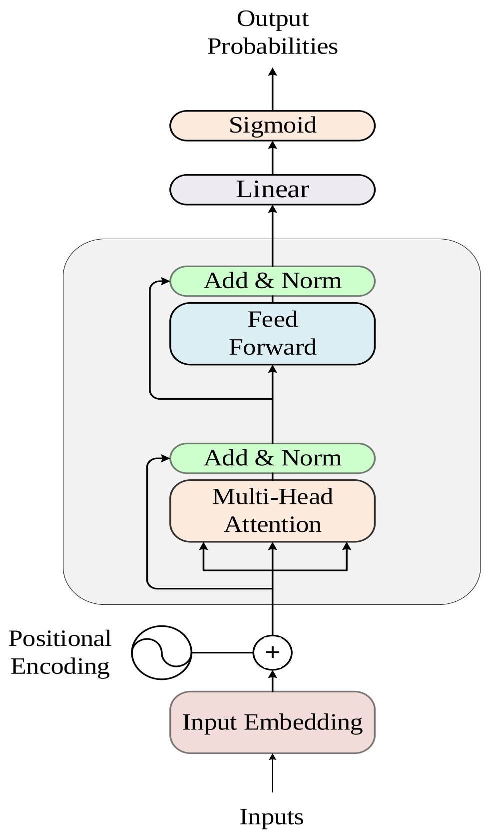


Figure 2. The AWL model structure diagram

Fig. 2 describes the specific process of adaptive wavelet function learning for univariate time series in detail. For multivariate time series, the AWL model is used to realize the parallel learning of wavelet function on each feature dimension, so as to obtain the wavelet function corresponding to each feature independently.

2.2.2. Wavelet transform

In this part, the wavelet transform will be performed using the wavelet function dynamically learned by the AWL model. The process is as follows.

Multivariate data preprocessing: For multivariate time series data, the primary step is to implement multi-channel wavelet transform to divide the input data into multiple independent channels in parallel, so as to prepare for the subsequent unified operation with univariate time series data.

Feature decomposition and parallel processing: for univariate and multivariate time series each characteristics of time sequence, to understand its internal structure and the dynamic characteristics of the AWL model learning to decomposition, the wavelet function approximation coefficients and detail coefficients. The approximate coefficients reflect the long-term trend of the time series. While the detail coefficients reveal short-term variations or noise properties.

Noise processing and reconstruction: The denoising threshold is determined based on the first-level wavelet detail coefficients by applying the standard estimation method based on the Gaussian noise model. Soft threshold processing is implemented to dynamically subtract the noise threshold to reduce the noise. This process is crucial to ensure the true dynamic range and key characteristics of the signal while reducing the noise.

Taking PSM (Pooled Server Metrics) dataset as an example, in this study, the wavelet transform technique is applied with integrated denoising function to transform the data and eliminate the noise. Fig. 3 shows the comparison of four key features (features 1-4) in the PSM dataset before and after processing by the adaptive wavelet transform module. Each subfigure corresponds to each of the four features. Feature 1 to feature 3 are the features that make outstanding contributions to anomaly detection in the process of feature extraction, while feature 4 is a comprehensive feature obtained by fusing other contributing features. In Fig. 3, the blue lines represent the original data, while the orange lines represent the reconstructed data after wavelet transform.

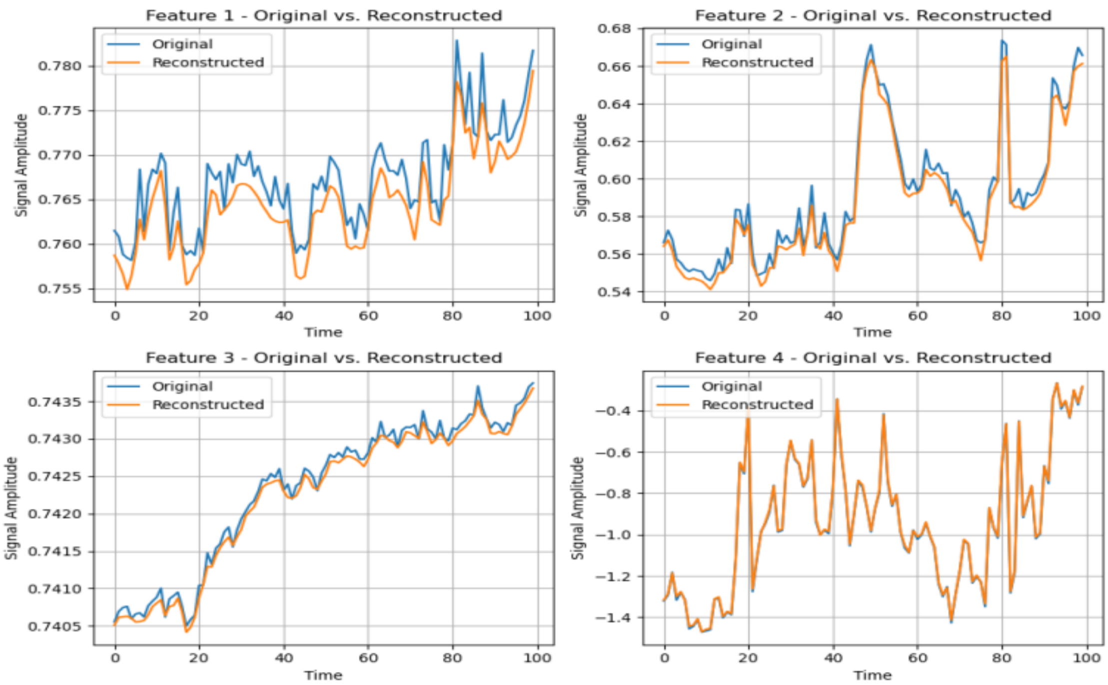


Figure 3. Wavelet transform before and after comparison

2.3. Anomaly Detection Module

The anomaly detection module combines the lightweight N-Beats block structure with the AE to construct the SNBAE model. The core of this model consists of the residual-enhanced N-Beats encoder and the sparse-optimized N-Beats decoder, both of which are built on the lightweight N-Beats model.

Therefore, this chapter will be divided into three parts: the first part introduces the improved lightweight N-Beats model; The second part introduces the construction of SNBAE model. The third part introduces the process of anomaly detection by the SNBAE model.

2.3.1. The Lightweight N-Beats model

In the traditional N-Beats model, time series anomaly detection is handled by concatenating multiple stacks, each consisting of multiple blocks in series. Although powerful, this structure is computationally expensive and not suitable for resource-constrained environments. In order to improve the computational efficiency and simplify the structure of the model, a lightweight N-Beats model is proposed. The lightweight operation is as follows.

The further nesting of blocks into Stack is eliminated, and multiple blocks are directly concatenated for sequence processing, which aims to simplify this model structure.

Each Block retains multiple fully connected layers and ReLU activation functions, but the number of layers and parameters are reduced, aiming to simplify the calculation process while maintaining the performance of the model.

The lightweight N-Beats model architecture is composed of multiple blocks in series, as shown in Fig. 4, where the left half is the specific structure in each Block, and the right half is the approximate framework of lightweight N-Beats. The input of the first Block is the original input sequence, the output is the reconstructed value of the Block input, and the input of the other blocks is the residual of the reconstructed value of the last Block input minus the output of the last Block. On the one hand, the next Block is a supplement to the previous Block, and the residual information that the previous Block does not fit is constantly fitted. On the other hand, this process can be regarded as the decomposition of time series information.

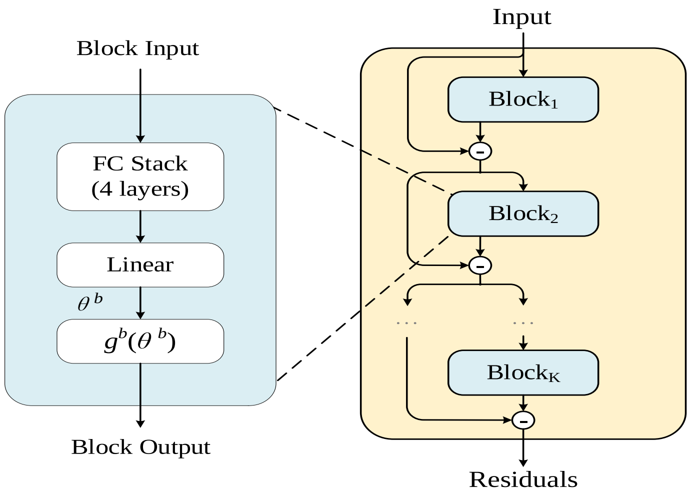


Figure 4. The lightweight N-Beats model

For each Block, the specific structure is shown in the left half of Fig. 4, which consists of four FC Stack layers (composed of a fully connected layer and a RELU activation function) and a simple linear layer. The four FC Stack layers perform the following operations.

$$\mathrm{h}_{l,1} = \mathrm{FC}_{l,1}(x_l) \tag{1}$$

$$\mathrm{h}_{l,2} = \mathrm{FC}_{l,2}(\mathrm{h}_{l,1}) \tag{2}$$

$$\mathrm{h}_{l,3} = \mathrm{FC}_{l,3}(\mathrm{h}_{l,2}) \tag{3}$$

$$\mathrm{h}_{l,4} = \mathrm{FC}_{l,4}(\mathrm{h}_{l,3}) \tag{4}$$

The data obtained from the FC Stack layer is fed into the Linear layer to realize a simple linear projection. The specific operation is as follows.

$$\theta_l^b = W_l^b \, \mathrm{h}_{l,4} \tag{5}$$

Where, $\mathrm{h}_{l,4}$ is the output of the last layer of the FC Stack, Wlb is the linear projection matrix that $\mathrm{h}_{l,4}$ will map to θ_l^b.

After eliminating the redundant information in the input, Block retains the low-dimensional vector $\hat{x}_l$ with information closely related to the prediction target or through the mapping function $g^b(\theta^b)$. In order to adapt to time series information, this paper designs the mapping function $g^b(\theta^b)$ into a specific functional form, and forces the Block to learn specific properties of time series to enhance the interpretability of N-Beats. The specific formula is given below.

$$\hat{x}_l = \sum_{i=0}^{[H/2-1]} \theta_{l,i}^b \cos(2\pi it) \; + \theta_{l,i+[H/2]}^b \sin(2\pi it) \tag{6}$$

H/2-1 is the number of terms of the series, and both $\theta_{l,i}^b$ and $\theta_{l,i+[H/2]}^b$ are the projection results of linear projections onto different terms.

2.3.2. The SNBAE model

(i) The residual-enhanced N-Beats encoder

In this part, the lightweight N-Beats are used as the encoder, and the output of each block is passed to the next block through the residual connection. The encoder processes the data as follows.

1) In the encoder structure, the input data first goes through a series of blocks. Each Block consists of multiple layers of FC (Fully connected) networks equipped with ReLU activation functions after each layer to introduce nonlinearities. After the data flows through these layers, it finally passes through the linear layer for dynamic output adjustment.

2) Introducing residual connection between Block mechanism. The output of each Block contains corrections (predicted values) and residuals for specific characteristics of the current input data. The current Block passes the residual to the next Block, ensuring that the next Block takes into account errors that the previous Block failed to capture at input, thus gradually reducing the overall prediction error.

3) The final output of the encoder consists of two parts: the residual and the corrected output. The residual is the part of the error that the model fails to fully capture throughout the processing pipeline, while the correction output is the prediction of the last Block and is used as input to the decoder.

The residual-enhanced N-Beats encoder architecture is suitable for complex application scenarios that require fine control of each data processing stage and deep mining of complex patterns in data.

It not only enhances the learning ability of the model, but also ensures the precision and depth of data encoding.

(ii) The Sparse Optimized N-Beats decoder

This part in N-Beats sparse optimization was applied in the process of decoding, make the model by multiple Block correction step by step and restore data characteristics, but more likely to choose a few key characteristics, so as to improve the quality of data reconstruction and explanation of this model. Fig. 5 shows this improved decoder structure for N-Beats in detail.

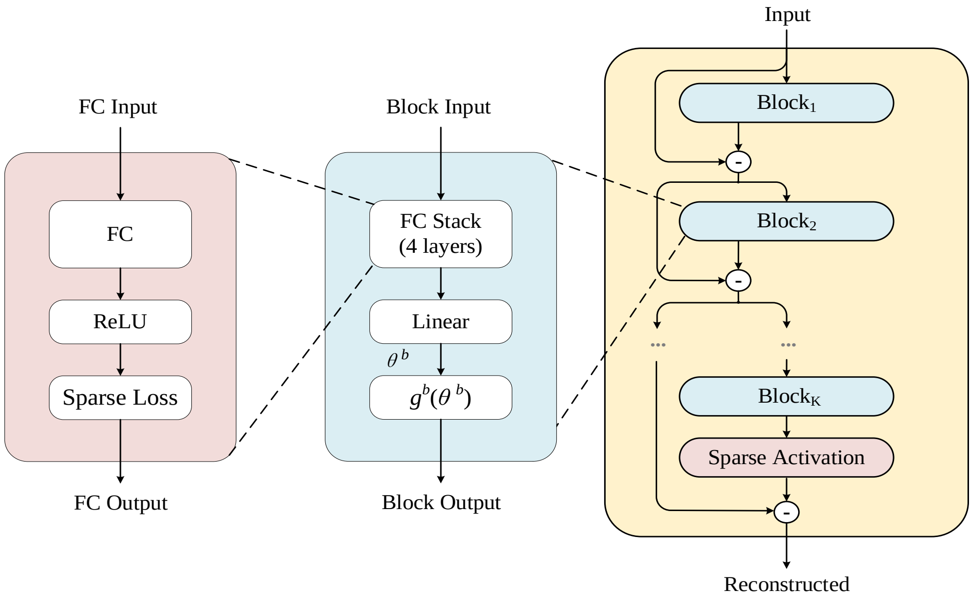


Figure 5. The Sparse Optimized N-Beats decoder

Combined with Fig. 5, the innovation of introducing the sparse optimization strategy in the decoder design based on the lightweight N-Beats model is deeply analyzed. The details are as follows.

1) Hierarchical sparsity Regularization: In the FC Stack shown in Fig. 5, not only the standard ReLU activation function is applied after each FC layer, but also the sparsity constraint based on L1 regularization is additionally introduced. This constraint acts directly on the output of the ReLU activation layer to encourage the network to generate sparse activation patterns during learning by penalize the sum of absolute values of nonzero activations. The sparse activation function (L1 regularization) is formulated as follows.

$$\text{SparsityLoss} = \lambda \sum |h| \tag{7}$$

Where ρ is the target sparsity and $\hat{\rho}_i$ is the average activation value of a neuron.

2) Dynamic sparsity adjustment: Sparse regularization is integrated into the loss function, and the regularization coefficient and sparsity parameter are adjusted through experiments and validation to achieve dynamic sparsity control.

2.3.3. Anomaly detection

In the process of anomaly detection of SNBAE model, finding the optimal threshold is a key step to achieve efficient anomaly detection. This model uses semi-supervised learning method to train on normal data to learn the normal pattern of the data, and uses the validation set to adjust and determine the most suitable anomaly detection threshold. The process consists of the following steps.

Reconstruction Error calculation: For each data point in the validation set, the model calculates the error between it and the reconstructed data. This error combines the reconstruction error and the sparsity penalty to preserve the sparsity of the encoding as much as possible.

Adaptive threshold selection: This model adaptively finds the optimal threshold by optimizing the F1 score. Specifically, this model tries different thresholds on the validation set and calculates the F1 score corresponding to each threshold to determine the threshold that maximizes the F1 score. This ensures that this model achieves as much precision as possible while maintaining a high recall, thus effectively balancing the number of false positives and false negatives.

This model evaluates whether each point in the test set is abnormal according to the best threshold, and calculates various performance indicators on the test set according to the gap between the evaluation results and the true value, such as precision, recall, F1 score, to comprehensively evaluate the overall performance of this model. Through these indicators, the anomaly detection effect of SNBAE model on different data sets can be verified, which proves its robustness and reliability in practical applications.

3. EXPERIMENT AND RESULT ANALYSIS

3.1. Experimental Setup

3.1.1. Dataset

The experiments are carried out on three public datasets, including KPI [17] dataset, SKAB(Skoltech anomaly benchmark library) and PSM [18], where KPI dataset is a univariate time series dataset, SKAB and PSM are multivariate time series datasets.

The KPI dataset consists of KPI (key performance index) time series data and labels from many real scenarios of Internet companies. A data set is a univariate time series data set containing one variable per set of data.

The SKAB (v0.9) [19] benchmark library contains 34 multivariate time series datasets, covering single point and collective anomaly labels, and the data are derived from real sensor tests. The PSM dataset is collected from multiple application server nodes of eBay and includes a multivariate time series dataset with 26 dimensions.

The details of the above three datasets are shown in Table 1.

Table 1. Dataset information

Dataset	Number of features	The number in test dataset	
		Normal	Anomaly
KPI	1	417945	8359
SKAB	34	21374	5596
PSM	26	51679	4651

3.1.2. Experimental parameter configuration

Pytorch1.12.0 deep learning framework was used for this model training, and the parameter configuration during training is shown in Table 2.

Table 2. Hyperparameter Settings

Training parameters	Parameter values
batch size	5
step length	50
iterations	300
initial learning rate	0.001

3.1.3. Evaluation metrics

The AW-SNBAE model performance is measured by three metrics: precision, recall, and F1 score. The following is the formula for calculating these metrics.

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1\_score = 2 \times \frac{precision \cdot recall}{precision + recall} \tag{10}$$

TP is the number of normal samples correctly identified by this model; FP is the number of anomalies that this model mistakenly identifies as normal; TN is the number of abnormal samples correctly identified by this model; FN is the number of normal that this model mistakenly identifies as abnormal.

3.2. Analysis of Experimental

3.2.1. Ablation experiment

In ablation experiments, the impact of each improvement on the overall performance of the model is comprehensively evaluated using the F1 score by progressively removing components from this model.

In this paper, four sets of ablation experiments were performed, and the details and experimental results are listed in Table 3. The detailed experimental design is as follows.

Experiment 1: The AWL model was removed and the time series were wavelet transformed using a fixed wavelet basis and wavelet series.

Experiment 2: Remove the complete adaptive wavelet transform module and directly feed the data into the SNBAE model for anomaly detection.

Experiment 3: The N-Beats model is removed in the anomaly detection module, and the ordinary AE is used for anomaly detection.

Experiment 4: Remove all sparse regularization terms from the SNBAE model and use lightweight N-Beats with AE for anomaly detection.

In addition, Experiment 5 verifies the proposed AW-SNBAE model.

Table 3. Ablation experiment analysis (All the results, the best ones are shown in bold and the second best ones are underlined.)

Experiment number	AWL	Wavelet Transform	N-Beats	Sparse Regularization	AE	KPI (F1)	SKAB (F1)	PSM (F1)
1	—	√	√	√	√	0.82	0.80	0.78
2	—	—	√	√	√	0.75	0.82	0.77
3	√	√	—	√	√	0.81	<u>0.88</u>	0.85
4	√	√	√	—	√	<u>0.86</u>	0.86	<u>0.90</u>
5	√	√	√	√	√	0.89	0.96	0.94

3.2.2. Comparative experiment

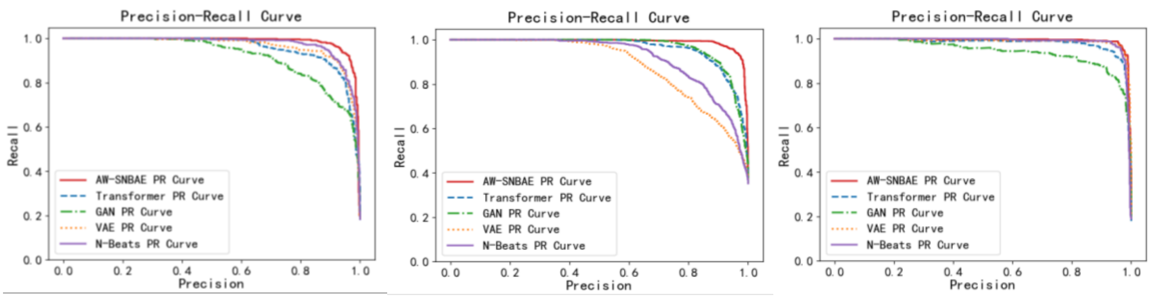
In order to comprehensively evaluate the performance of the AW-SNBAE model, this paper compares the AW-SNBAE model with four related algorithms in depth. The comparison algorithms include Generative adversarial network (GAN [20]), Transformer, Variational AutoeEncoder (VAE [21]), and N-Beats algorithm.

Under the same experimental environment, the four algorithms are tested on KPI, SKAB and PSM data sets respectively, and the experimental results are summarized in Table 4. In order to comprehensively evaluate the performance of the algorithm in anomaly detection tasks, especially considering the balance between precision and recall, F1 score is selected as the comprehensive evaluation index in this part. By analyzing the F1 score in Table 4, it can be seen that the AW-SNBAE model improves the KPI of the univariate time series data set by 8% compared with the optimal comparison algorithm, and improves the multivariable time series data set SKAB and PSM by 2% on average. The detection effect of the model in multivariate time series SKAB is not as good as the optimal model, but the model proposed in this paper has the most balanced effect by integrating the anomaly detection effect of univariate time series and multivariate time series.

Table 4. Comparative experimental results (All the results, the best ones are shown in bold and the second best ones are underlined.)

Dataset	KPI			SKAB			PSM		
Metric	P	R	F1	P	R	F1	P	R	F1
VAE [21]	0.76	0.83	0.79	0.95	0.97	0.96	0.95	0.78	0.89
GAN [20]	0.79	0.83	0.81	0.93	0.83	0.86	0.85	0.78	0.81
N-Beats [11]	0.80	0.78	0.77	0.94	0.97	0.95	0.93	0.87	0.90
Transformer [14]	0.67	0.85	0.75	0.92	0.92	0.92	0.82	0.94	0.87
AW-SNBAE (Our model)	0.92	0.88	0.89	0.93	0.98	0.96	0.92	0.95	0.94

In order to visually show the influence of different components on model performance, Precision-Recall (PR) curves of each comparison experiment in different data sets are plotted in this paper, as shown in Fig. 6. By comparing Fig. 6 (a), (b) and (c), it can be seen that although the PR curve of the AW-SNBAE model is not always at the top, the effect of the AW-SNBAE model is the smoothest and most balanced for data sets of different dimensions, indicating the effectiveness of the proposed model in the field of univariate time series and multivariate time series anomaly detection.



(a) KPI (b) SKAB (c) PSM

Figure 6. PR plot of the comparison algorithms for different datasets

4. CONCLUSIONS

This paper proposes the AW-SNBAE model based on semi-supervised learning, which aims to balance the precision and cost of the algorithm in time series anomaly detection, especially in the case of limited labeled data. By using partial labels, this model reduces the dependence on large-scale labeled data and achieves high precision in anomaly detection in both univariate and multivariate time series data. This model consists of two core components: the AWL model and the SNBAE model. The AWL model uses the global information of the sequence to adaptively learn the wavelet basis and wavelet series to achieve flexible wavelet transform. The SNBAE model simplifies the N-Beats architecture to improve the efficiency of feature extraction, and combines sparse regularization to optimize data reconstruction, so as to further improve the precision of anomaly detection.

The experimental results show that the AW-SNBAE model shows good anomaly detection accuracy on both univariate and multivariate time series data, especially in the case of reducing the use of labels, it still maintains good performance.

REFERENCES

[1] Kozitsin V, Katser I, Lakontsev D. Online forecasting and anomaly detection based on the ARIMA model [J]. Applied Sciences, 2021, 11(7): 3194.

[2] Hosseinzadeh M, Rahmani A M, Vo B, et al. Improving security using SVM-based anomaly detection: issues and challenges [J]. Soft Computing, 2021, 25(4): 3195-3223.

[3] Jiang J, Chen J, Gu T, et al. Anomaly detection with graph convolutional networks for insider threat and fraud detection[C]//MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM). IEEE, 2019: 109-114.

[4] Zong B, Song Q, Min M R, et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection[C]//International conference on learning representations. 2018.

[5] Mao J, Hu Y, Jiang D, et al. CBFS: a clustering-based feature selection mechanism for network anomaly detection [J]. IEEE Access, 2020, 8: 116216-116225.

[6] Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4027-4035.

[7] Pelati A, Meo M, Dini P. Traffic anomaly detection using deep semi-supervised learning at the mobile edge [J]. IEEE Transactions on Vehicular Technology, 2022, 71(8): 8919-8932.

[8] Yu Q, Kavitha M S, Kurita T. Mixture of experts with convolutional and variational autoencoders for anomaly detection [J]. Applied Intelligence, 2021, 51(6): 3241-3254.

[9] Kumagai A, Iwata T, Fujiwara Y. Semi-supervised anomaly detection on attributed graphs[C]//2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021: 1-8.

[10] Li P, Pei Y, Li J. A comprehensive survey on design and application of autoencoder in deep learning [J]. Applied Soft Computing, 2023, 138: 110176.

[11] Oreshkin B N, Carpov D, Chapados N, et al. N-BEATS: Neural basis expansion analysis for interpreTable time series forecasting [J]. arxiv preprint arxiv:1905.10437, 2019.

[12] Zhang D, Zhang D. Wavelet transform [J]. Fundamentals of image data mining: Analysis, Features, Classification and Retrieval, 2019: 35-44.

[13] Liu X, Liu H, Guo Q, et al. Adaptive wavelet transform model for time series data prediction [J]. Soft Computing, 2020, 24(8): 5877-5884.

[14] Kim J, Kang H, Kang P. Time-series anomaly detection with stacked Transformer representations and 1D convolutional network [J]. Engineering Applications of Artificial Intelligence, 2023, 120: 105964.

[15] Belghazi M I, Baratin A, Rajeshwar S, et al. Mutual information neural estimation[C]//International conference on machine learning. PMLR, 2018: 531-540.

[16] Greenacre M, Groenen P J F, Hastie T, et al. Principal component analysis [J]. Nature Reviews Methods Primers, 2022, 2(1): 100.

[17] Sauvanaud C, Kaâniche M, Kanoun K, et al. Anomaly detection and diagnosis for cloud services: Practical experiments and lessons learned [J]. Journal of Systems and Software, 2018, 139: 84-106.

[18] Li W, Feng C, Chen T, et al. Robust learning of deep time series anomaly detection models with contaminated training data [J]. arxiv preprint arxiv:2208.01841, 2022.

[19] Skolkovo Institute of Science and Technology. (n.d.). SKAB: Skoltech anomaly benchmark. GitHub. https://github.com/waico/SKAB

[20] **a X, Pan X, Li N, et al. GAN-based anomaly detection: A review [J]. Neurocomputing, 2022, 493: 497-535.

[21] Gouda W, Tahir S, Alanazi S, et al. Unsupervised outlier detection in IOT using deep VAE [J]. Sensors, 2022, 22(17): 6617.