Hybrid Deep Learning Model for Time Series Anomaly Detection

Chan Sik Han
Dept. of Computer Science
Chungbuk National University
Cheongju, Korea
chatterboy@chungbuk.ac.kr

Keon Myung Lee
Dept. of Computer Science
Chungbuk National University
Cheongju, Korea
kmlee@ chungbuk.ac.kr

ABSTRACT

Multivariate time series anomaly detection is a fundamental challenge in real-world applications such as industry and business1. To address this issue, numerous models with diverse structures have been proposed. Each model leverages its unique structural characteristics to extract crucial features for time series anomaly detection. Our objective is to assess whether the performance of multivariate time series anomaly detection can be improved by employing a combination of models with different structures. In this paper, we propose a hybrid model for multivariate time series anomaly detection. The proposed hybrid model comprises two sub-models, each with a unique structure, and a simple layer. Each sub-model is designed to extract significant features from input time series. The simple layer combines the extracted features from both sub-models to generate the final output. Experimental results demonstrate that the proposed hybrid model outperforms single models.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → Operations research → Industry and manufacturing.

KEYWORDS

Multivariate time series, Time series anomaly detection, Hybrid deep learning model

ACM Reference format:

Chan Sik Han and Keon Myung Lee. 2023. Hybrid Deep Learning Model for Multivariate Time Series Anomaly Detection. In *International Conference on Research in Adaptive and Convergent Systems (RACS '23), August 6-10, 2023, Gdansk, Poland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3599957.3606232

 ACM ISBN 979-8-4007-0228-0/23/08…$15.00
https://doi.org/10.1145/3599957.3606232

1 INTRODUCTION

Widespread deployment of IoT devices has resulted in the generation of abundant time-series data across various industrial and business sectors. Despite the significant advancements in deep learning techniques, which have found numerous successful applications in areas such as image, video, speech, and natural language processing, similar success in the time-series domain remains relatively limited. This is largely due to the inherent complexities of time series data, such as intricate dependencies among data elements, flexibility in data acquisition, and diversity in data collections and tasks. Nevertheless, time series data processing has been garnering increased attention due to its potential benefits in both industrial and business sectors. This type of data is generated continuously and often requires analysis and monitoring, sometimes in real time.

Time series data is a collection of observations indexed in chronological order, typically recorded at uniform intervals. Each successive observation depends on past values. Each observation can either be a single value (univariate) or multiple values (multivariate). When the statistical properties of time series data remain constant over time, the series is deemed stationary. However, many real-world time series data exhibit non-stationary behaviors, including seasonality, concept drift, and change points. Since time series data are often collected using field-deployed sensors, they are susceptible to noise, which can sometimes be indistinguishable from anomalies.

The increasing interest from industrial and business sectors has driven researchers and practitioners to invest significant effort into various tasks involving time series data. Time series classification aims to assign a class label to time series data by training a model with labeled time series data. Various techniques have been developed for this purpose, including traditional machine learning algorithms, dynamic time warping-based methods, transform-based approaches, and deep learning-based methods. Time series prediction, also known as time series forecasting, involves predicting future values based on previously observed data. There are several methods used for time series prediction, such as statistical methods like ARIMA (Autoregressive Integrated Moving Average), traditional machine learning methods like regression models and SVM, deep learning methods like RNN-based methods[1], temporal CNN models[2], and Transformer-based methods[3]. Additionally, ensemble models can be used to combine the predictions of multiple models. Time series data clustering groups time series data based on their similarity. This allows for the discovery of natural substructures within the data, which can be utilized for further analysis. Anomaly detection in time series data involves identifying unusual or unexpected patterns or data points that deviate from expected behavior. This has been an actively researched area, particularly because time series data analysis is often used to monitor the state of various systems.

This paper focuses on anomaly detection in time series data. At the time of writing, TimesNet[4] stands out as an exemplary model for this task. This deep learning model offers a general framework for the aforementioned time series tasks. It preprocesses input time series data into multiple 2D tensors, which are then sliced and stacked at prominent frequencies. The model uses convolutions in a hierarchical manner to extract features from these 2D tensors. This approach allows for the extraction both inter-period and intra-period features from the time series data.

This paper proposes a hybrid model that combines models with different structures. The hybrid model combines TimesNet and Transformer[5]. To effectively combine the different models, each model uses separate input embeddings and learnable positional embeddings. A simple linear layer is used to compute the final output from the results of each model. We demonstrated that by effectively combining models with different structures, the proposed hybrid model outperforms single ones.

2 RELATED WORK

2.1 Multivariate Times Series Anomaly Detection

Three distinct types of anomalies are typically observed in time series data: point anomalies, contextual anomalies, and collective anomalies[6]. A point anomaly refers to a single data point that significantly deviates from the remainder of the data. A contextual anomaly, on the other hand, is characterized by a data point that diverges significantly from the rest of the data, but only within a specific context. Lastly, collective anomalies occur when a group of related data points collectively deviate from the norm in relation to the entire dataset, even though the individual data points may not be considered anomalous in isolation.

Anomaly detection techniques for time series data primarily utilize machine learning-based approaches. These can be broadly divided into statistical methods, rule-based methods, clustering-based methods, spectral methods, and deep learning-based methods. Statistical methods construct a model of what's deemed normal, using historical data, and subsequently identify data points that significantly deviate from this model as anomalies. Techniques within this category include control charts, ARIMA, and its variants, among others. Rule-based methods establish specific rules or thresholds against which data points are compared to detect anomalies. These rules can be automatically determined by rule discovery algorithms. Clustering-based methods involve forming clusters of data points and detecting as anomalies those data points that reside in sparse regions. During clustering, a time series data is transformed into subsequences using sliding window sampling or non-overlapping partitioning. Spectral methods transform the time series into the frequency domain or another domain, identifying anomalies based on their spectral properties. These methods employ transformation techniques such as the Fourier transform, Wavelet transform, spectral residual transform, and singular spectrum analysis. Deep learning-based methods capitalize on the feature extraction and end-to-end training capabilities of deep learning models for time series data anomaly detection.

Deep learning-based methods for anomaly detection have been categorized into five groups: CNN-based methods, autoencoder-based methods, GAN-based methods, RNN-based methods, and transformer-based methods. CNN-based methods apply to 1D time series data or its transformed 2D tensors[4, 7, 8, 9]. These methods utilize dilated convolutions and causal padding to extract hierarchical features. The trained models predict future time steps, flagging significant discrepancies between the predicted and actual values as anomalies. Autoencoder-based methods compress time series data into a compact representation and then reconstruct the time series from this representation[10, 11]. Data points with large discrepancies between the reconstructed and actual values are considered anomalies. GAN-based methods are trained on normal data and then used to generate new data[12, 13]. The confidence level of the discriminator, indicating whether a new data point is real, can be used as a measure of anomaly. RNN-based methods[14, 15, 16] involve training RNN-based models, like LSTM, on normal data, and then using the trained model to predict future time steps. Any significant discrepancies are marked as anomalies. The THOC[17] model, an RNN-based model, uses multiple dilated RNN layers with multi-resolution recurrent skip connections. It forms multiple cluster centers of normal instances hierarchically and measures the difference between features and cluster centers at the last layer for an anomaly score. Transformer-based methods have been actively studied for anomaly detection[18, 19]. Here, transformers are tasked with extracting association information among near and far dependencies for data points.

2.2 Transformer and TimesNet

The Transformer[5] model was initially developed to tackle natural language processing tasks. Since its inception, its application domains have expanded to include computer vision, signal and speech processing, name a few. Transformer models employ multiple layered blocks to perform self-attention amongst data elements in parallel. This is accomplished by generating query, key, and value vectors for each data point, and subsequently generating features with weighted value vectors. The weights are determined by the similarity score between the corresponding query and keys. A Transformer model consists of an encoder and a decoder, and it is trained in an end-to-end manner. Certain Transformer-based models use either the encoder or the decoder exclusively, while others use both. When Transformer-based models are utilized for time series analysis,

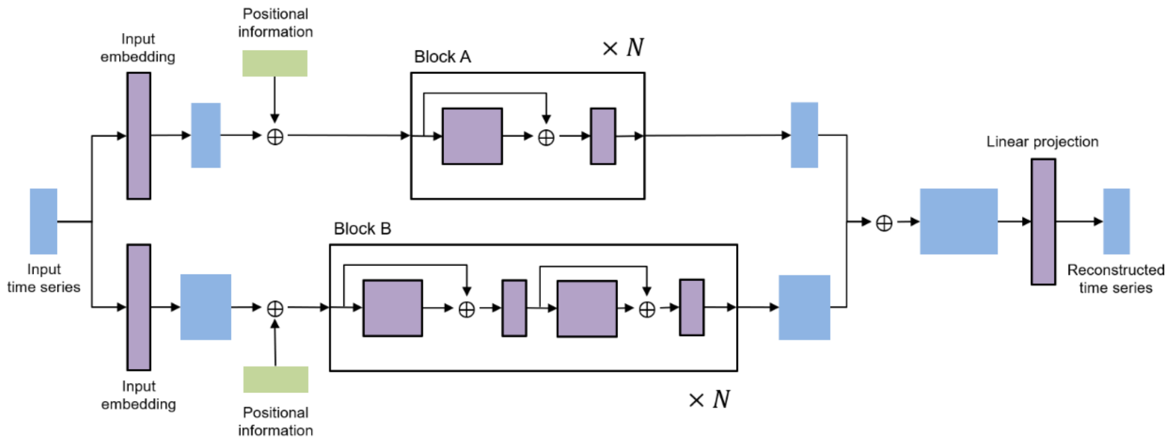

Figure 1. The overview of the proposed hybrid model. Final prediction, reconstructed time series, is computed by integrating the two outputs from the different sub-models. Each sub-model is equipped with its own input embedding and learnable position embedding to enhance flexibility and capability.

the transformation of the input time series data can significantly influence their performance.

TimesNet[4], a CNN-based model, provides a framework for time series data analysis by extracting useful features from time series data. TimesNet transforms 1D time series data into a 2D space by selecting the top-k amplitude values in its Fourier transform and obtaining the most significant frequencies. In accordance with each selected frequency, the input time series data is reshaped into 2D tensors. Each 2D tensor is processed by a unit called TimesBlock, which captures temporal 2D-variations and adaptively aggregates representations with an Inception block that performs multiple convolutions to extract features. However, TimesBlock uses convolutions to extract features and thus inherently struggles with long-term dependencies in intra-period and inter-period terms. To address this issue, we propose a hybrid model that combines the Transformer and TimesNet to enhance anomaly detection in time series data. proposed hybrid model outperforms single ones.

2 THE PROPOSED METHOD

Here, we introduce the proposed hybrid model. The hybrid model aims to leverage the strengths of sub-models with different structures and utilize the features extracted from them. This section begins with an explanation of the preprocessing technique. Then, we describe the structure of the proposed hybrid model. Lastly, we explain the procedure of conducting multivariate time series time series anomaly detection.

3.1 Preprocessing

In multivariate time series anomaly detection, it is vital to extract significant features from multivariate time series. However, it is difficult to extract important features directly from multivariate time series due to its inherent complexity and noise.

To reduce complexity and noise in multivariate time series, we preprocess multivariate time series. The preprocessing consists of two steps – resampling and normalization. Resampling not only is allowed to reduce the size of multivariate time series but also can keep important features[11]. Given a multivariate time series $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, resampling can be defined as follows:

$$\mathbf{x}^{resampled} = \{\mathbf{x}_\tau, \mathbf{x}_{2\tau}, \dots, \mathbf{x}_{k\tau}\} \tag{1}$$

where τ is a sampling period and $\mathbf{x}$ is resampled until $k\tau \leq T$. τ depends on the characteristics of multivariate time series. Multivariate time series involves different variables that have different scales or distributions. These factors make a model difficult to learn important features from multivariate time series. Normalization can solve this challenge. Normalization can be defined as follows:

$$x_{ij}^{normalized} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{2}$$

where μ_j is a mean and σ_j is a standard deviation for j-th variable.

Preprocessing plays a crucial role in reducing the inherent complexity and noise in multivariate time series. Hence, this process allows a model to capture important feature better. Furthermore, it facilitates more efficient learning by reducing the size of multivariate time series.

3.2 Model Architecture

Fig. 1 shows the structure of the proposed hybrid model. The proposed hybrid model comprises two different sub-models and a simple layer. Each sub-model involve its own input imbedding and learnable position embedding for flexibility and capability.

We realize two sub-models with TimesNet and Transformer. A simple layer plays role of integrating the outputs of two sub-models.

To feed multivariate time series, it is necessary to match the dimensions in a model with those of input time series. However, it often limits the flexibility and capability of a model. The appropriate dimensions in a model can vary based on several factors such as model type and dataset characteristics. To address this issue, we employ input embedding to transform the dimensions of input time series before it is fed into the model. By introducing input embedding, the model gains the flexibility and capability effectively. In the proposed model that consists of two distinct sub-models, each sub-model may necessitate different dimensions. Hence, we employ two different input embeddings to accommodate these variations.

Position embedding is widely adopted in transformer-based models to incorporate positional information into input time series or features. For the proposed model that combines a CNN-based model with a transformer-based model, we utilize two separate learnable position embeddings. Because we assume that models with different structures can learn unique positional information even at the same position.

The proposed hybrid model is built upon two different models – TimesNet and Transformer. TimesNet is a CNN-based model that transforms 1D times series into its 2D variants using its frequency components and then leverages convolutional layers to extract features from these transformed features. TimesNet demonstrated state-of-the-art performance across various time series tasks. On the other hand, Transformer has gained widespread adoption in diverse domains thanks to its ability to capture global dependencies. While Transformer-based models are typically used for 1D time series in time series domain, they consistently deliver state-of-the-art results. The proposed hybrid model combine TimesNet with Transformer to leverage their strengths: TimesNet focuses on learning features from local patterns, whereas Transformer is allowed to capture global dependencies. The proposed hybrid model is allowed to capture important features for time series anomaly detection by leveraging these effective models with different structures.

After input time series is fed into the two different models, the models output their own features. To make a final prediction, we use a simple layer, which consists of a concatenation and a linear layer to integrate the two different outputs.

3.3 Anomaly Detection

To conduct unsupervised time series anomaly detection, a model is trained using a training dataset without labels. Given an input time series $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ and its corresponding reconstructed time series $\hat{\mathbf{x}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T\}$ generated by a model, we can define a reconstruction loss function as follows:

$$\ell(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_p^p \tag{3}$$

where p represents p-norm, with values of 1 or 2 commonly used.

In time series anomaly detection, it is essential to have methods for computing anomaly scores and selecting an appropriate threshold. Computing an anomaly score can be defined as follows:

To conduct time series anomaly detection, we need a way to compute anomaly scores and we need a way to select a threshold. Computing anomaly scores can be defined as follows:

$$\mathrm{s}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 \tag{4}$$

where anomaly score $s(\cdot,\cdot)$ is similar with reconstruction loss. For threshold selection, we follow the threshold selection strategy by [4].

4 EXPERIMENTAL RESULTS

We conducted experiments to evaluate the effectiveness of the proposed hybrid model. We adopted widely used benchmark datasets in multivariate time series anomaly detection, namely SMD, SWaT, and PSM. SMD[15] includes 28 datasets, each consisting of multivariate time series with 38 variables. Each dataset has a train set and a test set. The total length of all train sets is 708,377 and the total length of all test sets is 708,392. SWaT[20] has a multivariate time series with 51 variables and involves a train set and a test set. The train set has a length of 495,000 and the test set has a length of 449,919. PSM[21] has a multivariate time series with 25 variables and includes a train set and a test set. The train set has a length of 132,481 and the test set has a length of 87,841. For all benchmarks, we split their train set into a train set and a validation set using a split ratio of 70:30. In case of SMD, which consists of multiple datasets, we split each dataset separately. We followed the training settings provided by [4], including optimization, learning rate, and so on. F1 score is used as the evaluation metric to evaluate the performance of a model. We trained a model on each benchmark for a total of 3 epochs.

Table 1 presents the experimental results on the benchmark datasets. The proposed method outperformed other models on SMD and SWaT and it achieved the second-best performance on PSM. Furthermore, we observed that for some benchmark datasets, the proposed model outperformed larger models, even when its size was significantly smaller. Therefore, combining the extracted features by two models with different structures can improve performance.

Table 1: Performance comparison on the benchmark datasets. 'I' indicates the use of the Inception block and 'R' denotes the use of the block of ResNeXt. Bold indicates the best results and underlined indicates the second-based results.

Method	SMD	SWaT	PSM
Transformer	0.7956	0.8037	0.7607
TimesNet + I	0.8512	<u>0.9210</u>	0.9521
TimesNet + R	<u>0.8581</u>	0.9174	**0.9747**
Ours + I	**0.8739**	**0.9296**	<u>0.9727</u>

5 CONCLUSIONS

This paper proposes a hybrid model for multivariate time series anomaly detection. The proposed hybrid model is composed of two sub-models and a simple layer. Sub-models adopt different structural models from each other. In addition, each sub-model has its own input embedding and learnable position embedding for flexibility and capability. These sub-models play an important role in extracting significant features from input time series. The simple layer is responsible for integrating the extracted features from both models to generate the final output. Experimental results demonstrate that the proposed hybrid model outperforms TimesNet, confirming its effectiveness in multivariate time series anomaly detection. Hence, the hybrid model effectively leverages unique features extracted from models with distinct structures for multivariate time series anomaly detection.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986) and this work was carried out with the support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ016247012022)" Rural Development Administration, Republic of Korea.

REFERENCES

[1] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting 36, 3 (2020), 1181–1191.
[2] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. 2017. Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691 (2017).
[3] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems 34 (2021), 22419–22430.
[4] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 (2022).
[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017).
[6] Kukjin Choi, Jihun Yi, Changhwa Park, and Sungroh Yoon. 2021. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access 9 (2021), 120043–120065.
[7] Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. 2021. Neural contextual anomaly detection for time series. arXiv preprint arXiv:2107.07702 (2021).
[8] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 8980–8987.
[9] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. Advances in Neural Information Processing Systems 35 (2022), 3988–4003.
[10] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148 (2016).
[11] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria AZuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 3395–3404.
[12] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In International conference on artificial neural networks. Springer, 703–716.
[13] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2020. Tadgan: Time series anomaly detection using generative adversarial networks. In 2020 IEEE International Conference on Big Data (Big Data). IEEE, 33–43.
[14] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 387–395.
[15] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2828–2837.
[16] Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T Kwok. 2021. Time series anomaly detection with multiresolution ensemble decoding. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 9567–9575.
[17] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. Advances in Neural Information Processing Systems 33 (2020), 13016–13026.
[18] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642 (2021).
[19] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284 (2022).
[20] Jun Inoue, Yoriyuki Yamagata, Yuqi Chen, Christopher M Poskitt, and Jun Sun. 2017. Anomaly detection for a water treatment system using unsupervised machine learning. In 2017 IEEE international conference on data mining workshops (ICDMW). IEEE, 1058–1065.
[21] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2485–2494.