Transactions on Computational and Scientific Methods | Vo. 4, No. 10, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Mamba-Based Temporal Feature Learning for Anomaly Detection in Distributed Microservice Environments

Ke Wu
Northeastern University, Boston, USA
wikiwu24@gmail.com

Abstract: To address the challenges of identifying abnormal behavior, strong temporal correlations in runtime states, and complex dynamic changes in multidimensional monitoring data within backend microservice systems, this paper proposes an anomaly detection method based on the Mamba state-space model. This method focuses on the continuously generated monitoring sequences during microservice operation, constructing a unified detection framework around temporal state modeling and anomaly representation learning to enhance the ability to identify complex anomaly patterns. First, the raw monitoring data is serialized and feature-mapped, transforming heterogeneous runtime signals into a compact representation suitable for temporal modeling. Then, a gating modulation mechanism is introduced to adaptively filter key information in the input features, enhancing the expression intensity of anomaly-related patterns in the feature space. Based on this, a selective state transition mechanism is used to model the evolution of hidden states under continuous input, thereby more effectively preserving long-range temporal dependencies and perceiving local anomaly disturbances. Simultaneously, a residual enhancement strategy is combined to stabilize the deep representation propagation process, further improving the model's ability to characterize complex runtime states and discriminate anomalies. This research focuses on anomaly data in open-source microservices and evaluates the proposed method against several related approaches. The results show that the proposed method achieves superior performance in terms of precision, recall, F1 score, and AUC, enabling more accurate identification of potential anomalies in backend microservice scenarios. The study demonstrates that the state-space modeling-based anomaly detection framework provides a more effective technical path for anomaly identification in backend microservice systems and offers methodological support for intelligent monitoring and anomaly analysis in complex distributed software environments.

Keywords: Microservice anomaly detection; state-space modeling; temporal feature learning; intelligent operation and maintenance.

1. Introduction

With the rapid development of cloud computing, container orchestration, and continuous delivery technologies, backend systems are accelerating their evolution towards microservice architectures[1]. Compared to traditional monolithic systems, microservices improve system maintainability, scalability, and business iteration efficiency through service decomposition, independent deployment, and elastic scaling, becoming an important paradigm for building large-scale distributed applications. However, the continuous growth in the number of services, the continuous extension of call chains, and the highly dynamic changes in

the operating environment have also made the internal dependencies of the system more complex, and the operating state exhibits stronger time-varying and uncertainties. In this context, abnormal behavior is often no longer limited to a single node or a single indicator, but appears in the form of cross-service propagation, cascading amplification, and covert evolution, significantly increasing the system's operational risks and the difficulty of operation and maintenance.

Anomaly detection in backend microservice systems is a critical link in ensuring service stability, improving resource utilization efficiency, and reducing the scope of failure impact. If anomalies are not identified promptly, they may further lead to problems such as increased interface latency, resource imbalance, link blockage, or even service avalanche, thereby affecting the availability of the entire platform and user experience. Because microservice systems typically exhibit high concurrency, high coupling, heterogeneity, and strong dynamism, traditional methods relying on manual rules, fixed thresholds, or shallow feature modeling often struggle to adapt to the diversity and evolution of anomaly patterns in complex scenarios. Especially with the continuous generation of massive amounts of monitoring data, accurately capturing potential anomalies from multi-dimensional time-series signals and improving the timeliness and stability of anomaly identification has become a crucial issue in backend intelligent operations and maintenance research[2,3].

From a methodological development perspective, the core of backend microservice anomaly detection lies in the unified modeling of long-term dependencies, local mutation characteristics, and multivariate correlation structures. The large amounts of logs, metrics, and call status information generated during microservice operation not only have significant temporal correlations but also exhibit dynamic changes at different time scales. If the model cannot effectively characterize the long-term state evolution process, it may easily overlook slowly accumulating anomaly trends; conversely, if it only focuses on global information and lacks sensitivity to key local patterns, it may miss weak precursors before anomaly outbreaks. Therefore, a novel method is urgently needed that balances long-range dependency modeling capabilities, temporal state representation capabilities, and computational efficiency to meet the comprehensive requirements of accuracy, robustness, and real-time performance for anomaly detection tasks in complex backend scenarios[4].

State-space modeling provides a new research approach for complex temporal analysis, with the Mamba state-space model-based modeling mechanism showing significant potential in long-sequence feature extraction and dynamic information compression. Introducing it into backend microservice anomaly detection tasks helps extract more stable and discriminative state representations from continuously evolving system observations, enhancing the ability to perceive complex operating patterns and hidden anomaly signals. Research on backend microservice anomaly detection methods based on the Mamba state-space model can not only provide new technical support for distributed system fault early warning and intelligent operation and maintenance, but also help promote the in-depth application of temporal modeling methods in cloud-native application scenarios, which has important theoretical significance and practical value for improving the reliability, security, and autonomous operation and maintenance level of modern software systems.

2. Related work

In recent years, research on anomaly detection in backend microservices has primarily focused on intelligent analysis methods based on logs, metrics, call chains, and multi-source observation data[5]. Early methods typically relied on statistical feature extraction, rule matching, or threshold discrimination, identifying anomalies by monitoring operational signals such as response time, resource consumption, and error rate. These methods are relatively inexpensive to implement and suitable for systems with relatively simple structures and stable fluctuation patterns. However, in complex microservice environments, anomalies often exhibit a complex combination of cross-service propagation, localized bursts, and temporal coupling. Traditional methods have significant limitations in feature representation and pattern generalization capabilities. With the development of deep learning methods, research has gradually shifted towards utilizing

recurrent networks, convolutional networks, attention mechanisms, and graph structure modeling techniques to jointly characterize temporal dependencies, topological relationships, and multivariate interactions in microservice systems, thereby improving the anomaly detection's ability to perceive and adapt to complex patterns[6].

In the area of temporal modeling, existing research increasingly focuses on balancing representational capability and computational efficiency under long-sequence conditions. Methods based on recurrent structures have certain advantages in continuous state modeling, but they remain limited in terms of long-distance dependency propagation and parallel computation[7]. Attention-based methods can enhance global information capture capabilities, but they often face high computational costs and insufficient local dynamic characterization as sequence length continues to increase. In contrast, state-space models, by explicitly describing the evolution of input signals and hidden states, provide a new technical path for efficient modeling of complex time series. Especially in backend microservice scenarios, the system's operating state is characterized by continuous evolution, frequent local disturbances, and significant multi-scale changes. State-space modeling is more conducive to extracting stable temporal representations from dynamic observations. Therefore, research on anomaly detection in backend microservices based on state-space models is not only a natural extension of existing time series analysis methods in complex distributed system scenarios, but also an important direction for improving the accuracy, robustness, and practical value of anomaly identification.

3. Methodology

Modern backend microservice environments generate multivariate execution signals whose abnormal behaviors are often weak at the local level but progressively amplified along temporal evolution, making anomaly detection fundamentally dependent on the quality of sequential state modeling. Rather than treating each observation as an isolated event, the proposed method organizes the monitoring stream into a continuous latent transition process so that transient perturbations, slow drifts, and abrupt disruptions can be represented within a unified temporal framework.

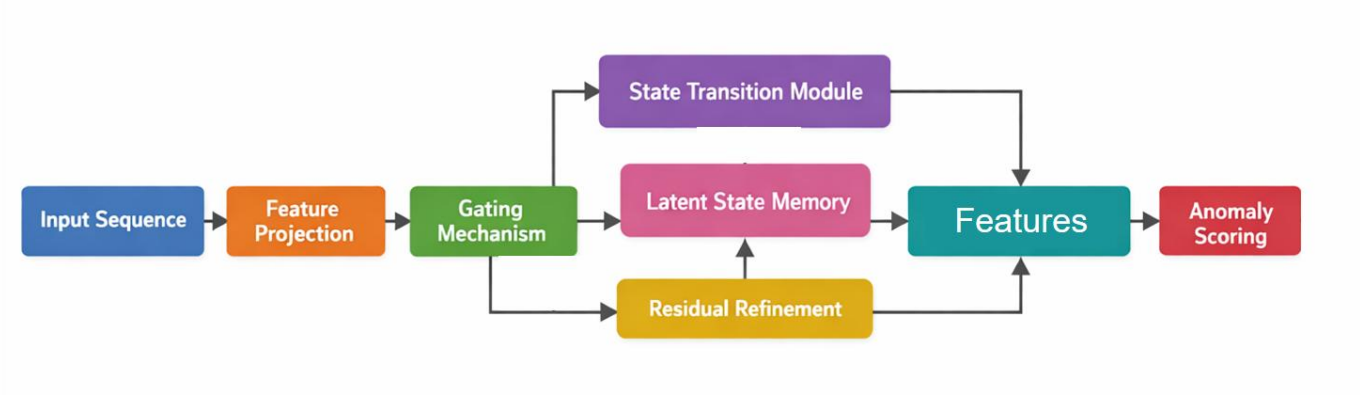

Figure 1. Overall model architecture

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{T \times D}$ denote the input sequence collected from a backend microservice system, where T is the sequence length and D is the number of monitored attributes. A linear projection is first introduced to map heterogeneous raw signals into a compact feature space, because direct modeling in the original domain tends to preserve noise and scale inconsistency across variables:

$$\boldsymbol{z}_t = \boldsymbol{W}_e \boldsymbol{x}_t + \boldsymbol{b}_e$$

Here, $\boldsymbol{W}_e \in \mathbb{R}^{d \times D}$ and $\boldsymbol{b}_e \in \mathbb{R}^d$ define the embedding transformation, while $\boldsymbol{z}_t \in \mathbb{R}^d$ provides a temporally aligned representation that is more suitable for subsequent state evolution. To strengthen

sensitivity to structural changes hidden in neighboring observations, the embedded sequence is further modulated by a gating mechanism that adaptively reweights informative channels under dynamic operating conditions:

$$\tilde{\boldsymbol{z}}_t = \sigma\left(\boldsymbol{W}_g \boldsymbol{z}_t + \boldsymbol{b}_g\right) \odot \boldsymbol{z}_t$$

Such a design suppresses unstable fluctuations and preserves salient precursors of anomalous behavior, with $\sigma(\cdot)$ denoting the sigmoid function and $\odot$ representing elementwise multiplication.

A selective state space transition is then employed to capture long-range dependencies without incurring the high redundancy commonly introduced by exhaustive pairwise interactions. Distinct from static sequence encoders, this component updates the latent state according to the current input context, allowing the model to retain useful historical information while remaining responsive to newly emerging irregularities. The hidden dynamics are formulated as:

$$\boldsymbol{h}_t = \boldsymbol{A}_t \boldsymbol{h}_{t-1} + \boldsymbol{B}_t \tilde{\boldsymbol{z}}_t$$

where $\boldsymbol{h}_t \in \mathbb{R}^m$ is the latent state at time step t, and the input-dependent transition matrices $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ are generated through learnable selective mappings so that the state evolution can adapt to varying service workloads and interaction patterns. The corresponding output response is defined by:

$$\boldsymbol{y}_t = \boldsymbol{C}_t \boldsymbol{h}_t + \boldsymbol{D}_t \tilde{\boldsymbol{z}}_t$$

Through this formulation, the model integrates accumulated temporal memory with immediate signal evidence, where $\boldsymbol{C}_t$ controls the projection from latent dynamics to observable features and $\boldsymbol{D}_t$ preserves the direct influence of current inputs. This combination is important because microservice anomalies may arise either from gradual state deviation or from sudden local shocks, and both sources should contribute to the detection process.

Beyond temporal dependency alone, reliable anomaly identification also requires emphasizing intervals whose semantic variation is disproportionate to normal operational rhythms. A residual refinement branch is therefore introduced to stabilize feature propagation and prevent critical anomaly cues from being diluted during recurrent state updates:

$$\boldsymbol{r}_t = \boldsymbol{y}_t + \boldsymbol{W}_r \tilde{\boldsymbol{z}}_t$$

In this expression, $\boldsymbol{W}_r$ is a trainable matrix used to inject shallow information into the deeper sequential response, thereby improving robustness when the latent transition encounters highly volatile system behavior. Since abnormal events are ultimately manifested as deviations from learned normality, the refined representation is aggregated into an anomaly score through a nonlinear decision function:

$$s_t = \sigma\left(\boldsymbol{w}_o^{\top} \boldsymbol{r}_t + b_o\right)$$

The scalar $s_t \in (0,1)$ quantifies the abnormal tendency of the current observation, with larger values indicating stronger inconsistency with regular temporal patterns. In practical backend scenarios, this scoring strategy is meaningful because it transforms complex sequential states into interpretable decision signals while preserving sufficient flexibility for diverse anomaly manifestations.

Training is guided by the principle that the learned state trajectory should distinguish normal and abnormal behaviors at the sequence level while maintaining temporal coherence inside the latent space. For a labeled

dataset $\mathcal{D} = \{(\boldsymbol{X}^{(n)}, \boldsymbol{q}^{(n)})\}_{n=1}^{N}$, where $\boldsymbol{q}^{(n)} = [q_1^{(n)}, q_2^{(n)}, \ldots, q_T^{(n)}]$ denotes the binary annotation sequence, the objective function is defined as:

$$\mathcal{L} = -\frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\left[q_t^{(n)} \log s_t^{(n)} + \left(1 - q_t^{(n)}\right) \log\left(1 - s_t^{(n)}\right)\right] + \lambda \sum_{n=1}^{N}\sum_{t=2}^{T} \left\| \boldsymbol{h}_t^{(n)} - \boldsymbol{h}_{t-1}^{(n)} \right\|_2^2$$

The first term drives discriminative anomaly prediction, whereas the second term regularizes abrupt and unnecessary latent oscillations through the coefficient λ, which helps the model preserve physically plausible temporal transitions in backend service observations. Once optimization converges, the resulting framework can characterize sequential evolution, highlight informative disturbances, and convert hidden state deviations into actionable anomaly judgments, thereby offering a methodologically coherent solution for backend microservice anomaly detection based on the Mamba state space model.

4. Experimental Results and Analysis

4.1 Dataset

This paper selects an open-source anomaly monitoring dataset built on the Train Ticket microservice system as the research object. This dataset, publicly released on Zenodo, is designed for anomaly analysis tasks in microservice architectures. It contains ten sets of monitoring data, covering logs, Jaeger call chain tracing data, and Prometheus metric data, and provides corresponding explanations for anomalies. Because the data simultaneously retains cross-service call information and internal service operation observation information during request execution, it can effectively reflect the characteristics of anomaly propagation, state fluctuations, and link disturbances in backend microservice scenarios, exhibiting strong task matching and suitability for research on backend microservice anomaly detection methods. Furthermore, this dataset originates from a publicly available microservice benchmark system, with a clear data organization, facilitating sequence modeling and multi-source observation analysis.

From the perspective of suitability to the paper's theme, this dataset can provide a suitable data foundation for temporal anomaly detection based on the Mamba state-space model. On the one hand, logs, metrics, and tracking data themselves have significant temporal correlation and state evolution characteristics, which are conducive to constructing continuous sequence inputs and characterizing the dynamic operation process of microservice systems. On the other hand, Train Tickets have been used in microservice anomaly detection research. Their call chains and operation records can more realistically reflect the dependencies and abnormal behavior patterns of backend services, thereby enhancing the consistency between research scenarios and actual backend microservice systems.

4.2 Experimental setup

To ensure that the backend microservice anomaly detection method based on the Mamba state-space model can be evaluated under unified conditions, the experimental environment revolves around four aspects: data preprocessing, sequence construction, model training, and parameter setting. First, logs, call chain tracing information, and operational metrics from open-source microservice anomaly data are uniformly organized. Monitoring data from different sources is aligned chronologically, and continuous input sequences are constructed based on fixed time windows to preserve the dynamic evolution of the system's operational state. Subsequently, input features are standardized to reduce the dimensional differences between different monitoring dimensions and enhance the model's learning stability for temporal fluctuation patterns. During the training phase, supervised learning is used to optimize the sequence samples, enabling the model to identify potential anomalies from continuous state changes and develop a more stable anomaly discrimination capability within the temporal context. To avoid parameter oscillations and local overfitting during training, the learning rate, batch size, regularization term, and Dropout ratio are uniformly controlled in the experiment, with specific settings shown in Table 1.

Table 1: Hyperparameter settings

Parameter Category	Parameter Name	Value
Data Settings	Time Window Length	50
Data Settings	Sliding Step Size	10
Data Settings	Input Feature Processing	Standardization
Training Settings	Batch Size	32
Training Settings	Epochs	100
Training Settings	Optimizer	AdamW
Training Settings	Initial Learning Rate	1e-4
Training Settings	Weight Decay	1e-5
Training Settings	Dropout	0.1
Model Settings	Embedding Dimension	128
Model Settings	Hidden State Dimension	128
Model Settings	Number of Mamba Layers	4
Model Settings	State Space Channels	32
Output Settings	Anomaly Decision Method	Sigmoid Binary Classification Output

Table 1 presents the core configuration parameters of the experiment, including time window length, sliding step size, optimizer type, initial learning rate, hidden layer dimension, and state space-related settings. These parameters collectively determine the model's ability to model long-range dependencies, local anomalies, and hidden state evolution. Specifically, the time window length controls the temporal range covered by a single input sequence, the sliding step size balances the number of samples and sequence overlap, and the hidden layer dimension and state space dimension affect the model's ability to express complex operating patterns. Meanwhile, the optimizer and weight decay parameters help improve the model's convergence stability, while Dropout enhances the model's generalization ability. Through the above experimental settings, consistent and reliable experimental conditions can be provided for subsequent method performance analysis while ensuring the standardization of the training process.

4.3 Experimental Results and Analysis

This section compares the proposed method with representative studies closely related to backend microservice anomaly detection. These methods cover major technical approaches such as log analysis, distributed tracing modeling, graph-based representation, and multi-source monitoring fusion. Based on this,

Table 2 summarizes the performance of different methods under common evaluation metrics, thus supporting a unified comparison between the proposed method and existing research.

Table 2: Experimental results compared with other models

Method	Precision	Recall	F1	AUC
Zhang et al.[8]	88.34	87.29	87.81	91.42
Pawar et al.[9]	89.17	88.05	88.61	92.08
Kohyarnejadfard et al.[10]	87.46	86.71	87.08	90.64
Khanahmadi et al.[11]	88.92	88.13	88.52	91.87
Jiang et al.[12]	89.41	88.76	89.08	92.35
Xie et al.[13]	90.08	89.24	89.66	93.11
Panahandeh et al.[14]	90.37	89.58	89.97	93.46
Ours	92.14	91.63	91.88	95.27

Compared to existing methods, the state-space modeling framework constructed in this paper not only enhances the preservation of long-term temporal context information but also improves the perception of local anomalies, enabling the model to obtain more stable and discriminative feature representations during anomaly identification. This demonstrates that introducing the Mamba state-space model into backend microservice anomaly detection tasks is reasonable and effective, and can better adapt to the practical characteristics of the continuous evolution of runtime states, hidden anomaly forms, and complex propagation paths in microservice scenarios.

Furthermore, the proposed method achieves satisfactory results in terms of accuracy, completeness, and overall discriminative ability, reflecting that the model can not only reduce the misclassification of normal samples as anomalies but also more fully identify genuine anomaly samples, thereby improving the reliability and practicality of anomaly detection results. This result indicates that the designed feature embedding, gating modulation, state transition modeling, and residual enhancement mechanisms form a good synergy, enabling the model to extract more representative anomaly clues from complex monitoring sequences. For backend microservice systems, this method helps improve the stability and robustness of anomaly detection, providing more reliable methodological support for subsequent fault warning and intelligent operation and maintenance.

The results of the ablation experiments are also presented in this paper, as shown in Table 3.

Table 3: Ablation study results of different module settings

Ablation Setting	Precision	Recall	F1	AUC
w/o Gating Mechanism	90.42	89.76	90.09	93.54
w/o Selective State Transition	89.87	89.13	89.50	92.91

w/o Residual Refinement	90.76	90.05	90.40	94.08
Full Model	92.14	91.63	91.88	95.27

The ablation results show that the proposed complete method performs best in overall performance, indicating that each component plays an irreplaceable role in the anomaly detection process. The gating mechanism enhances the model's ability to filter key temporal information, enabling more complete preservation of anomaly-related features in complex monitoring sequences. The selective state transition module improves the modeling ability for long-range dependencies and dynamic state evolution processes, allowing the model to more accurately capture potential anomaly patterns during microservice operation. The residual enhancement module further stabilizes the feature transfer process, improving the completeness and discriminativeness of anomaly representation. Therefore, the proposed method achieves effective synergy across feature modulation, state modeling, and representation enhancement, ensuring good anomaly detection performance and strong model robustness.

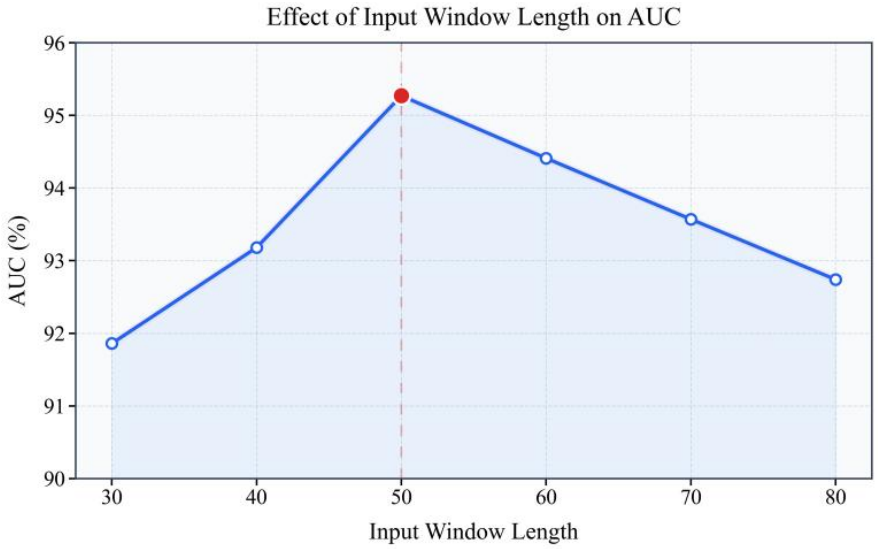

Figure 2. The impact of input time window length on AUC

Figure 2 shows that the proposed algorithm achieves good anomaly detection results under this time window setting, indicating that the constructed state-space modeling mechanism can adequately characterize the key dynamic information in the backend microservice monitoring sequence. By organizing the input sequence into temporal segments with continuous semantic association, the model can more effectively preserve anomaly-related features during hidden state updates and form stable discriminative representations under complex operating backgrounds. This demonstrates that the designed method has strong applicability in microservice anomaly detection tasks and can provide reliable temporal representation support for anomaly identification.

From the perspective of the method itself, the results further illustrate that the proposed model possesses good state perception and anomaly characterization capabilities when processing multidimensional monitoring data in backend microservice scenarios. The synergistic effect of modules such as gating modulation, selective state transition, and residual enhancement enables the model to extract more representative anomaly clues from continuous inputs and enhances the stability of identifying complex system operating states. Therefore, the anomaly detection framework built based on the Mamba state-space model not only has good feature representation capabilities but also provides an effective methodological support for subsequent fault warning and intelligent operation and maintenance of microservice systems.

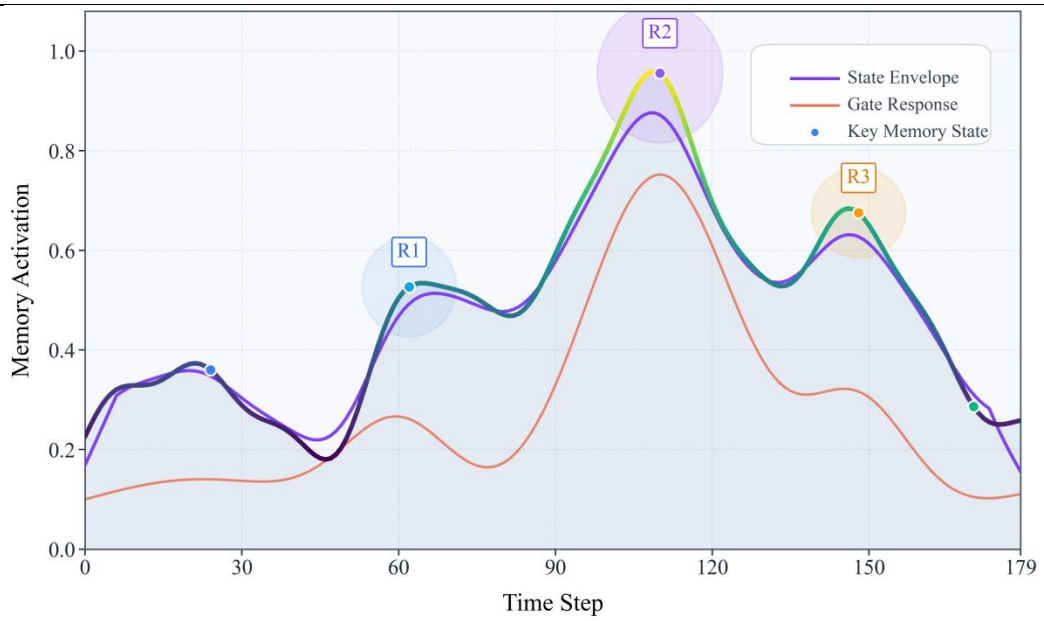

Figure 3. Visualization Experiment of Dynamic Memory Response in Mamba State Transition Process

As shown in Figure 3, the proposed algorithm can form clear and stable state activation responses in the backend microservice monitoring sequence, indicating that the model has a strong continuous modeling capability for key temporal information. During the continuous evolution of complex inputs, the latent state representation does not exhibit disordered fluctuations but rather forms a relatively concentrated memory reinforcement effect around anomaly-related regions. This demonstrates that the designed selective state transition mechanism can effectively preserve core semantic information related to anomaly detection. For backend microservice scenarios, this dynamic memory modeling approach helps improve the model's perception depth of potential anomaly clues, making anomaly representations more continuous, complete, and interpretable.

The dynamic memory response, state envelope, and gating response in the figure maintain a good synergistic relationship, indicating that the proposed method constructs a relatively natural linkage mechanism between feature selection, state update, and representation enhancement. Through this synergistic cooperation in its internal structure, the model can more effectively extract key anomaly information from multidimensional monitoring data and stably map it into the latent state space, thereby enhancing the discriminative ability and robustness in the anomaly identification process. This also demonstrates that the backend microservice anomaly detection method based on the Mamba state space model not only possesses excellent temporal modeling capabilities but also exhibits strong structural rationality and methodological effectiveness in its internal response mechanism.

5. Conclusion

This paper addresses the challenges of concealed abnormal behavior, complex propagation paths, and insufficient characterization of temporal dependencies in backend microservice systems. It proposes an anomaly detection method based on the Mamba state-space model. To overcome the shortcomings of traditional temporal modeling methods in maintaining long-sequence states, detecting local anomalies, and representing complex dynamic processes, this paper constructs a unified detection framework for backend microservice monitoring sequences from the perspective of continuous state evolution modeling. This method jointly models the long-term dependency information and key anomaly clues contained in multi-dimensional monitoring signals during system operation through feature embedding, gating modulation, selective state transitions, and residual enhancement, thereby improving the representation and discrimination

capabilities in the anomaly identification process. Overall, this work provides a new approach with strong temporal modeling characteristics for anomaly detection in complex distributed software systems and expands the research path for the application of state-space models in backend intelligent operation and maintenance scenarios.

From a methodological perspective, the detection framework constructed in this paper emphasizes the dynamic characterization of continuous temporal states, elevating the anomaly detection task in backend microservice systems from static feature matching to the level of hidden state evolution modeling. Because microservice environments typically involve frequent changes in service instances, dynamic adjustments to dependencies, and continuous fluctuations in workload, anomaly signals often do not directly manifest as drastic shifts in a single metric. Instead, they are more likely to reflect the accumulation of subtle perturbations and the spread of anomaly patterns within the state space. Based on this, the method proposed in this paper can more effectively perceive local key changes while preserving global temporal semantics and form a more stable anomaly representation during continuous input. This research demonstrates that introducing state-space modeling mechanisms into backend microservice anomaly detection is not only theoretically reasonable but also provides more natural technical support for understanding the operational state and identifying anomalies in complex software systems.

From an application perspective, this research has significant practical implications for cloud-native platform operation and maintenance, distributed service governance, online business stability assurance, and intelligent fault early warning. With the increasing prevalence of large-scale service-oriented systems, modern backend platforms' requirements for anomaly detection methods are no longer limited to simply identifying faults, but rather focus more on the timeliness of anomaly perception, the stability of state representation, and adaptability to complex business environments. The research and construction of the proposed method in a backend microservice scenario helps improve the automation and intelligence of system operation and maintenance, reduce manual troubleshooting costs, shorten anomaly location and fault response time, and provide a more solid model foundation for platform reliability construction. Simultaneously, this research also plays a positive role in promoting the integrated application of time-series modeling methods in interdisciplinary fields such as software engineering, cloud computing operation and maintenance, and service computing, and can, to a certain extent, facilitate the further transformation of related fields from experience-driven to data-driven and model-driven approaches.

Future research can be further deepened in several directions. On the one hand, it can further explore deep collaborative modeling mechanisms for multi-source heterogeneous observation data in more complex real-world production environments, more tightly integrating indicators, logs, call chains, and service topology relationships into a unified state representation framework, thereby enhancing the model's ability to understand complex abnormal behaviors. On the other hand, it can combine online learning, adaptive updates, and continuous evolutionary modeling ideas to improve the model's long-term adaptability in dynamic business scenarios, enabling it to better cope with practical problems such as service structure changes, workload migration, and anomaly pattern evolution. Furthermore, the interpretable expression of anomaly detection results, the characterization of anomaly propagation paths, and the integrated modeling of detection and diagnosis also have high research value. With the continuous development of intelligent operation and maintenance technology, the backend microservice anomaly detection method based on a state space model is expected to serve the construction of highly reliable software systems on a larger scale and provide continuous support for the safe and stable operation of complex digital infrastructure in the future.

References

[1] X. Yang, "Trend-Fluctuation Decomposition with Deep Residual Networks for System Forecasting," 2024.

[2] H. Chen, P. Chen, B. Wang, et al., "Graph neural network based robust anomaly detection at service level in SDN driven microservice system," Computer Networks, vol. 239, p. 110135, 2024.

[3] M. Panahandeh, A. Hamou-Lhadj, M. Hamdaqa and J. Miller, "ServiceAnomaly: An anomaly detection approach in microservices using distributed traces and profiling metrics," Journal of Systems and Software, vol. 209, p. 111917, 2024.

[4] S. Jacob, Y. Qiao, Y. Ye and B. Lee, "Anomalous distributed traffic: Detecting cyber security attacks amongst microservices using graph convolutional networks," Computers & Security, vol. 118, p. 102728, 2022.

[5] K. Shi, J. Li, Y. Liu, et al., "BSDG: Anomaly Detection of Microservice Trace Based on Dual Graph Convolutional Neural Network," in International Conference on Service-Oriented Computing, Cham: Springer Nature Switzerland, 2022, pp. 171-185.

[6] C. Zhao, M. Ma, Z. Zhong, et al., "Robust multimodal failure detection for microservice systems," in Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5639-5649.

[7] J. Huang, Y. Yang, H. Yu, et al., "Twin graph-based anomaly detection via attentive multi-modal learning for microservice system," in 2023 38th IEEE/ACM International Conference on Automated Software Engineering, IEEE, 2023, pp. 66-78.

[8] C. Zhang, X. Peng, C. Sha, et al., "Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning," in Proceedings of the 44th International Conference on Software Engineering, 2022, pp. 623-634.

[9] P. P. Pawar and C. Hartung, "TraceGra: A trace-based anomaly detection for microservices using graph neural networks," Computer Communications, vol. 204, pp. 109-117, 2023.

[10] Z. Wang, "Federated Multi-Scale Representation Learning for Privacy-Aware Log Anomaly Detection in Distributed Cloud Environments," 2024.

[11] M. Khanahmadi, A. Shameli-Sendi, M. Jabbarifar, et al., "Detection of microservice-based software anomalies based on OpenTracing in cloud," Software: Practice and Experience, vol. 53, no. 8, pp. 1681-1699, 2023.

[12] Y. Wang, "Semantic-Driven Large Model Scheduling for Distributed Systems via Unified Representation and Policy Generation," 2024.

[13] Z. Xie, H. Xu, W. Chen, et al., "Unsupervised anomaly detection on microservice traces through graph VAE," in Proceedings of the ACM Web Conference 2023, 2023, pp. 2874-2884.

[14] F. Chen, "AI-Augmented Anomaly Detection via Generative Distribution Modeling and Uncertainty Quantification in Cloud Systems," 2024.