Markov State Transition Modeling in Complex High-Dimensional State Space Based on Fuzzy Integral

Jinhan Guo[1,2,3], Kai Li[1], Hanhui Li[1,2,3], Wenxiang Liu[4], Zeming Zhuang[1,2,3], Yong Zhou[1], and Yang Yang[1]

[1]School of Information Science and Technology, ShanghaiTech University
[2]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]National University of Defense Technology
{guojh1, likai, lihh1, zhuangzm, zhouyong, yangyang}@shanghaitech.edu.cn, liuwenxiang8888@163.com

2022 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-5975-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCWkshps56602.2022.10008684

***Abstract*—The extremely high business volume of the financial industry brings unaffordable operating pressure to the back-end data system of financial companies. Recently, data-driven deep learning algorithms have achieved breakthroughs in analyzing and predicting system anomalies. However, in the case of high-dimensional data, deep learning faces the problems of long training time, lack of explainability and transferability. In this paper, we propose a model based on fuzzy integral for observing and modeling the state of the system. Firstly, the fuzzy integral algorithm has lower complexity, which is more suitable for the time-sensitive financial industry. Then, based on the fuzzy integral, the vector composed of the fuzzy measures of all the features is used to represent the state of the system. It is proved that the system constructed by this modeling method has the Markov property. Moreover, compared with deep learning, fuzzy integral-based methods are not only more computationally efficient but also explainable and transferable. Experimentally, we use the actual data of securities companies and have better results in the systematic anomaly analysis.**

***Index Terms*—Choquet fuzzy integral, anomaly detection, Markov-chains-based state space.**

I. Introduction

As a vertical industry where information technology (IT) is primarily used, IT has critical application scenarios in improving service quality, reducing operating costs, and ensuring business security. Cloud computing provides a platform for the application of big data and machine learning in the financial industry [1]. The application of cloud computing in IT systems improves the efficiency of system operation and management, but it also brings some risks. Reference [2] proposes three possible risks of cloud computing architectures: stability risk, availability risk and preservation risk. Simultaneously, large IT platforms have experienced downtime. Because cloud computing services have a sizable market, outages could result in significant economic losses. Particularly in a time-sensitive industry such as finance, an anomaly interruption of the IT system may lead to unquantifiable economic losses.

The IT platform is a highly dynamic and extremely complex network system. From the external factors of the IT platform, user business distribution is complex, and user activities are unpredictable. From the perspective of internal factors, IT systems have nonlinear characteristics. The network topology, software and hardware of the system will be upgraded, which makes the system time-varying. The dynamic deployment of resources such as microservice modules, middleware, and network switching equipment makes the system dynamic and complex. These factors bring challenges to the operation and maintenance of IT systems.

Much of the studies on system anomaly detection are based on deep learning methods. Reference [3] proposes an unsupervised and sequential autoencoder ensembles based anomaly detection framework, which trains KPI sequences through RSC-RNN, which maximizes the preservation of sequence information. Reference [4] proposes an LSTM-based VAE unsupervised anomaly detection algorithm with anomaly scoring by local outlier factor. Reference [5] extracts the features of the KPI sequence based on CNN, and judges the abnormality by analyzing the difference between the predicted value and the actual value. Besides deep learning method, references [6]–[8] use the data of the microservice system to reflect the system architecture, and combine the method of big data statistics to obtain the system graph structure to analyze the system anomaly situation through the obtained graph structure. Reference [9] compares several distance-based anomaly detection algorithms. The problem with distance-based algorithms is that the anomaly threshold is fixed. However, the operating state of the system is not fixed, and the threshold needs to change with the system state. Reference [10] proposes KELOS to estimate data density, which improves the problem of immutable thresholds in local anomaly detection based on distance. The main feature of the black-box approach is that the model needs to learn historical anomaly data. The limitations of this approach are as follows:

- Insufficient explainability. Explainability is defined as whether a parameter in a model can be explained in physical sense. The black-box method is based on phenomena and does not analyze the interaction relationship between

features.

- Insufficient transferability. Transferability is a capability that is reusable. Since the basic features are stable, once constructed, there is no need to retrain. Nevertheless, when the microservice architecture varies, data collection and training need to be re-implemented.
- Insufficient systematicness. Our model can analyze the state of the system macroscopically across function nodes and networks. The analysis granularity of black-box model is single, which means that the model can only analyze the problem from one perspective.

The lack of explainability reduces the trust and verifiability of decisions in deep learning-based systems [11], [12]. To address this problem, [13], [14] propose explainability of deep learning models. For the transferability problem of deep learning, reference [15] surveyed some studies on transfer learning. Transfer learning can solve the problem of insufficient transferability to a certain extent. This paper is based on back-end data from securities company A. The sensitivity of data in the financial industry and the time-varying nature of high-dimensional data systems require models to have explainability, transferability and systematicness at the same time. In order to solve the above problems, our contributions include:

- We build a Markov state transition model based on fuzzy measure vectors for each host. Models based on fuzzy measures are explainable, transferable and systematic.
- The anomaly detection based on the above model is achieved. At the same time, the running time of the algorithm is tens of milliseconds in case of 10-20 input features, which is much lower than the 10 seconds required by the industry.

Section II will introduce the fuzzy integral theory and anomaly localization and analysis algorithm based on the fuzzy measure. Section III will verify the fuzzy measures to describe the rationality of the system and the experimental results of the anomaly detection and localization algorithm. Section IV is the conclusion of this paper.

II. System Model

A. Choquet Fuzzy Integral

Sugeno developed the concept of fuzzy measure to solve the multi-attribute decision-making problem, where there is correlation between attributes but no additivity. The Fuzzy measure can effectively characterize the relationship between many attributes and evaluate the overall significance of one or more attributes [16]. As a result, the importance model built on fuzzy measure attributes has the potential to capture the significance of each feature. There are two main types of fuzzy integral theories based on fuzzy measures, namely the Sugeno fuzzy integral proposed by Sugeno and the Choquet fuzzy integral. Among them, Choquet fuzzy integral is widely used. In some studies, it is combined with AI to enhance the accuracy and explainability of AI models [17], [18].

Since the KPI of an IT system is a concept affected by multi-dimensional factors, the contribution of each factor to the KPI of the observed object can be reflected through fuzzy measures. We can observe m features of a host. The set $\mathbf{X} = \{x_1, x_2, ..., x_m\}$ represents the feature set. At the same time, we can get the KPI value y_i in the corresponding observation period of the host. With n observations, for the above observation process, we can abstract a function f. The i-th observation for the j-th feature is f_{ij}, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Then, we can get Table 1.

TABLE I
Feature observations and system KPIs

x_1	x_2	...	x_m	y_i
f_{11}	f_{12}	...	f_{1m}	y_1
f_{21}	f_{22}	...	f_{2m}	y_2
...	...	...	...	...
f_{n1}	f_{n2}	...	f_{nm}	y_n

In the i-th observation, the contribution of the combination of different features can be modeled as parameter combination fuzzy measure (PCFM), and the combination of different features is an element $\mathbf{e}$ in the power set of $\mathbf{X}$. The above calculation process can be abstracted as a mapping $\boldsymbol{\mu}$, satisfying $\boldsymbol{\mu}(\mathbf{e}) \to \mathbb{R}$ and $\boldsymbol{\mu}(\phi) = 0$.

Equation (1) is the computational expression of the choquet fuzzy integral,

$$y = \int_{(C)} f d\boldsymbol{\mu} = \int_{0^-}^{\infty} (F_T)\, dT, \tag{1}$$

where $F_T = \{x \mid f(x) \leq T, T \in \mathbb{R}\}$. Reference [19] discusses the use of the non-additivity method to solve fuzzy integral. This method has high complexity and is not suitable for the real-time financial field. Therefore, in order to reduce the complexity of solving equation (1), the f_{ij} are rearranged according to their value to satisfy (2),

$$f(x_{(0)}) \leq f(x_{(1)}) \leq f(x_{(2)}) \leq ... \leq f(x_{(m)}), \tag{2}$$

where $f(x_{(0)}) = 0$ and $x_{(1)}, x_{(2)}, x_{(3)}, ..., x_{(m)}$ is rearranging result of $x_1, x_2, x_3, ..., x_m$ according (2). Based on the rearrangement of (2), equation (1) can be transformed into the sum form of equation (3),

$$\int_{(C)} f d\boldsymbol{\mu} = \sum_{i=1}^{m} \left(f\left(x_{(i)}\right) - f\left(x_{(i-1)}\right)\right) \boldsymbol{\mu}\left(\boldsymbol{\Theta}_{\mathbf{i}}\right), \tag{3}$$

where $\boldsymbol{\Theta}_{\mathbf{i}} = \{x_{(i)}, x_{(i+1)}, ..., x_{(m)}\}$.

Reference [19] proposes an algorithm for computing PCFM, and we improve on it. In order to get the contribution of each feature to the system KPI, algorithm 1 shows a PCFM-based method to compute parameter additive fuzzy measure (PAFM). In algorithm 1, $\boldsymbol{\zeta}_i^T = \left\{f_i\left(x_{(1)}\right) - f_i\left(x_{(0)}\right), \ldots, f_i\left(x_{(m)}\right) - f_i\left(x_{(m-1)}\right)\right\}$, $\mathrm{bit}(k, i)$ represents the value of the i-th bit in the binary number corresponding to the decimal number k.

Algorithm 1: Calculate the PAFM algorithm

Input: Pre-processed $\boldsymbol{\zeta}_i^T$, host KPI sequence $\mathbf{y}$, hyper-parameter ϵ
Output: PAFM vector $\boldsymbol{\mu}_{\text{PAFM}}$
Initialization:
$\boldsymbol{\mu}_{\text{PAFM}} = [0, ..., 0]$,
$\boldsymbol{L}$ is a vector where $\boldsymbol{L}_i$ represents the original index of each parameter.
for $i = 1, ..., n$ **do**
  $\mathbf{f}_{i,sort} = \{f_i(x_{(0)}), f_i(x_{(1)}), ..., f_i(x_{(m)})\}$,
  $\boldsymbol{L}_{i,sort}$ represents the index after sorting,
  $\boldsymbol{\mu}_{\text{PCFM}} = \text{PCFM}(\boldsymbol{\zeta}_i,\ y_i,\ \epsilon,\ \boldsymbol{L}_{i,sort})$.
for $i = 1, ..., m$ **do**
  $\boldsymbol{\mu}_{\text{PAFM}}[i] = \sum_{k=1}^{2^m-1} \boldsymbol{\mu}_{\text{PCFM}}[k] \times \text{bit}(k, i)$
return $\boldsymbol{\mu}_{\text{PAFM}}$

def PCFM($\boldsymbol{\zeta}_i,\ y_i,\ \epsilon,\ \boldsymbol{L}_{i,sort}$)
  $\boldsymbol{\mu}_{\boldsymbol{\zeta}}^T = \boldsymbol{\zeta}_i \left(\boldsymbol{\zeta}_i^T \boldsymbol{\zeta}_i\right)^{-1} y_i$
  for $j = 1, ..., m$ **do**
    compare $\boldsymbol{L}$ and $\boldsymbol{L}_{i,sort}$ and find $\boldsymbol{\zeta}_{i,j}$ corresponding parameter combination,
    update it with $\boldsymbol{\mu} = \epsilon\boldsymbol{\mu} + (\mathbf{1} - \boldsymbol{\mu})\boldsymbol{\mu}_{\boldsymbol{\zeta}}$.
  return $\boldsymbol{\mu}$

B. Anomaly Detection and Localization Algorithm

Algorithms based on state transition have achieved good results in the field of industrial control systems [20]. However in previous studies, the definitions of system states and transition relationships were specified manually, so the construction of the state transition graph model requires much manual work. As the system architecture evolves, prior knowledge about the system becomes outdated. For complex IT systems, the number of possible states of the system is enormous, so these states need to be defined automatically.

In this paper, the discretized mapping of the PAFM vector calculated in a specific time period is used as the method of dividing the state. The reason for discretization is that the elements in the PAFM vector calculated by algorithm 1 are continuous values, so discretization processing is required to perform state division. After the PAFM is normalized according to $x_{norm} = (x - x_{min})/(x_{max} - x_{min})$, the values of $0-1$ are evenly mapped to the set $\{0, 1, 2, ..., L\}$. The larger the value, the greater the contribution to the KPI. From the above fuzzy integral theory, PAFM is the contribution of each feature to KPI. Therefore, the discretized PAFM vector directly reflects the contribution level of the features of the system to the KPI in the current state. The effectiveness of this method will be illustrated in the first part of Section III. When a piece of data appears and directly affects the discretized PAFM, it means that the state of the system has shifted.

For anomaly detection, we add average KPIs to states determined by discretely PAFM vectors. The reason for this is that when the PAFM vectors are similar, it indicates that the contributions of each feature to the KPI are in a similar state. The average KPI at this time is also similar. As a new record is entered, we can update the average KPI with $Avg_{n+1} = (n \times Avg_n + K_{n+1})/(n + 1)$, where Avg_n represents the average KPI value at the previous moment, K_{n+1} represents the KPI value of the input data and n represents the number of times this state occurs.

The basis for our judgment of anomaly is that when the KPI in the latest record exceeds the average KPI Δ times of the corresponding state, it is judged that the new input data is anomalous. Furthermore, comparing the PAFM vector calculated in this time period with the PAFM vector of the previous time period, the feature with the largest increase in the fuzzy measure of the two vectors contributes the most to the anomaly, so it is used as the result of root cause localization.

Algorithm 2: Anomaly detection algorithm

Input: System observations $\boldsymbol{D}[N]$ in N time periods,
State space vector $\boldsymbol{s}$,
Output: anomaly boolean sequence $\boldsymbol{a}$
Initialization:
State calling times vector $\boldsymbol{s}_n = [0, .., 0]$,
State KPI vector $\boldsymbol{s}_{\text{K}} = [0, ..., 0]$,
anomaly boolean vector $\boldsymbol{a} = [0, \ldots, 0]$,
for $i = 1, ..., len(\boldsymbol{D})$ **do**
  According to algorithm 1, compute $\boldsymbol{\mu}_{\text{PAFM}}$ corresponding to $\boldsymbol{D}[i]$,
  map $\boldsymbol{\mu}_{\text{PAFM}}$ to discrete $\boldsymbol{\mu}_{\text{d,PAFM}}$,
  $j = \boldsymbol{s}.\text{index}(\boldsymbol{\mu}_{\text{d,PAFM}})$
  update $\boldsymbol{s}_{\text{k}}[j]$,
  update $\boldsymbol{s}_n[j]$,
  Get the KPI value k of the latest record,
  if $k > \Delta \times \boldsymbol{s}_K[j]$ **then**
    $\boldsymbol{a}[i] = 1$
    Comparing the $\boldsymbol{\mu}_{\text{PAFM}}$ based on $\boldsymbol{D}[i-1]$, the feature corresponding to the element with the largest contribution increment is the root cause of this anomaly
return $\boldsymbol{a}$

III. Data Processing and Theory Verification

A. Microservice Architecture of Experimental IT System

The data used in this paper comes from the data center of securities company A. The experimental IT system consists of four function nodes, and each function node contains several hosts. An example of its business flow is shown in figure 1. From figure 1, we can see that the business flow generated by the system will pass through function node A, so the analysis result of function node A can reflect the condition of the entire system. The data we obtained includes software calling and hardware monitoring of function nodes B, C, and D, so these three modules can be analyzed from two perspectives.

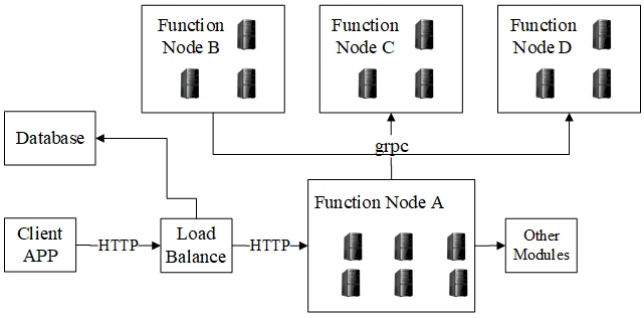

Fig. 1. IT system architecture

B. Feature Engineering

The original data outputs by the IT system contain fields including timestamp, function name and corresponding delay. Combined with the algorithm described above, we define the function name as the feature of the system and the delay as the system KPI. For subsequent analysis needs, we need to construct $\boldsymbol{P}$ from the original data, and record the average KPI of each feature of the host within an observing interval. First, we divide the entire observation period into several observation intervals, the length of which is the parameter d. Then, in original data, we perform the following statistics process in the i-th observation interval. $\boldsymbol{P}_n[i,j]$ represents the number of times the $j-$th feature is called, and $\boldsymbol{P}_t[i,j]$ represents the sum of the delays of the j-th feature. We define $\boldsymbol{P}[i,j] = \boldsymbol{P}_t[i,j]/\boldsymbol{P}_n[i,j]$. Each row in $\boldsymbol{P}$ is called as a *record*. We also define $y_i = \sum_{j=1}^{m} \boldsymbol{P}_t[i,j] / \sum_{j=1}^{m} \boldsymbol{P}_n[i,j]$ as the average KPI of the system in i-th observation. Table 2 shows the structure of the $\boldsymbol{P}$. The horizontal axis represents the index of the observation interval, and the vertical axis represents the index of the feature. KPI_{ij} represents the average KPI of the j-th feature in the i-th observation interval. The last column $\boldsymbol{y}$ of the table represents the system KPI.

TABLE II
STRUCTURE OF $\boldsymbol{P}$

	t_1	t_2	...	t_N	$\mathbf{y}$
$func_1$	KPI_{11}	KPI_{12}	...	KPI_{1N}	y_1
$func_2$	KPI_{21}	KPI_{22}	...	KPI_{2N}	y_2
⋮	⋮	⋮	⋮	⋮	⋮
$func_M$	KPI_{M1}	KPI_{M2}	...	KPI_{MN}	y_n

In order to observe the changes of the system over time, we perform sliding window processing on the $\boldsymbol{P}$ data to obtain a windowed data set. The sliding window length is w_l, and the sliding step size is s_l. That is, every time window slides, s_l new records enter the end of the window, and s_l old records move out from the head of the window at the same time. Finally, through algorithm 1, each group of window data obtains a $\boldsymbol{\mu}_{\text{PAFM}}$ vector, which becomes a state of the system after discretization processing. After obtaining the process of system state transition, we can analyze the system anomaly through algorithm 2.

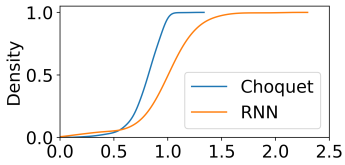

Fig. 2. Prediction error CDF

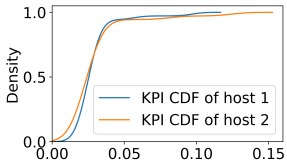

Fig. 3. KPI CDF of two hosts

C. Verification of Fuzzy Integral Results

To verify whether the fuzzy integral-based method can effectively characterize the system state, we design a set of comparative experiments. We use the fuzzy integral method and the RNN model to train and predict the KPI of the system respectively, and compare and predict the relative error calculated by

$$l = \begin{cases} \frac{y_p[i]-y_l[i]}{y_l[i]}, y_l[i] \neq 0 \\ \frac{y_p[i]}{0.01}, y_l[i] = 0 \end{cases}, \tag{4}$$

where y_p denotes the prediction value and y_l denotes the label value. We selected data within a period of time for training and prediction and obtained the relative error cumulative distribution function (CDF) as shown in figure 2. From the results, the prediction error obtained by the fuzzy integration method is lower and more stable. By comparison, the algorithm based on fuzzy integral is better than the algorithm based on RNN in predicting KPI results. Therefore, the PAFM vector can accurately model the state of the system.

D. Verification of explainability and transferability

In order to verify the effect of dividing the system state based on PAFM, we verify it based on part of the actual data, and take $L = 4$. That is, the continuous values of PAFM are evenly mapped to the set $\{0, 1, 2, 3, 4\}$. The state transition diagram we obtained is shown in figure 4, where the information of each node contains a discrete PAFM vector containing 7 features and the average KPI in this state. The directed edge represents the state transition, and the transition frequency under the experimental data is marked. We can see that a state can be transferred to another state in the next window or remain in this state, and some states appear more frequently, which can represent the normal operating state of the system.

1) explainability: Based on algorithm 2, we obtain the above state transition model. Furthermore, we get the output anomaly boolean sequence of algorithm 2, as shown in figure 5. In order to verify the explainability of our model, we arbitrarily select an anomaly point from the results. Figure 5 shows $\boldsymbol{\mu}_{\text{PAFM}}$ before and after the anomaly. Among them, the horizontal axis Ftr$i$ represents the index of the feature, and the vertical axis represents the value of PAFM. As can be seen from figure 5, Ftr2 has the largest fuzzy measure increment before and after this anomaly, that is, the KPI increases, and Ftr2 contributes the most to it. Therefore, this feature will be determined as the root cause of this anomaly. To sum up, the parameters in our model have practical meaning.

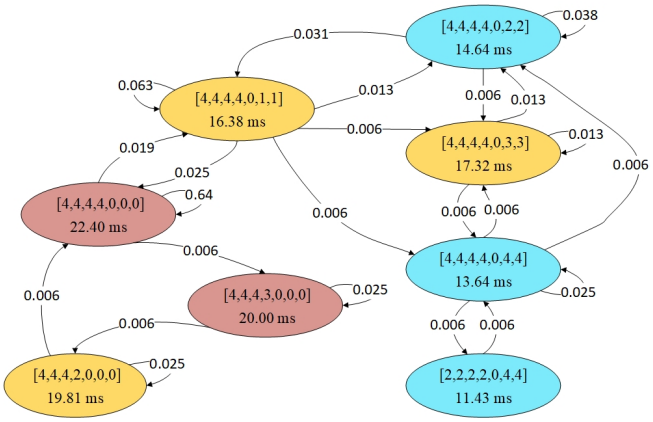

Fig. 4. State transition example

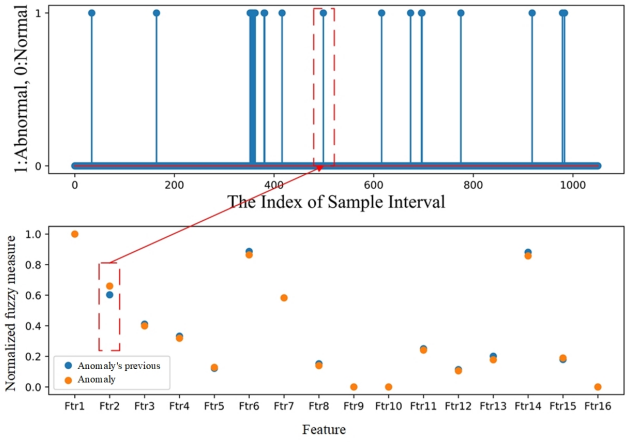

Fig. 5. PAFM corresponding to anomaly

2) transferability: We select two hosts in the same function node, and use the data of the two hosts to transfer the model to the two states through algorithm 2. Because the average KPI value of each state is an important criterion for us to judge anomalies. We calculate the CDF of the average KPI under the same state of the two sets of data, as shown in figure 3, and calculate the KL divergence of the two CDF lists through

$$D_{\mathrm{KL}}(\boldsymbol{CDF}_1||\boldsymbol{CDF}_2) = \sum_{i=1}^{n} p(x) log \frac{p(x)}{q(x)}. \tag{5}$$

The final result is $D_{\mathrm{KL}} = 0.03$, which shows that when the resource configuration and input business models are similar, the same state in the state space can be reused.

E. System Memoryless Verification

In the actual IT system, the system is memoryless. That is, the system has the Markov property. We use the chi-square test method to verify the Markov property of the above system description method. Assuming that the state set in the system contains p states, the element c_{ij} in the transition matrix represents the frequency of transition from state i to state j. According to the state transition matrix $\mathbf{C}$, the marginal probability can be obtained by (6),

$$p_{\cdot j} = \frac{\sum_{i=1}^{m} c_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}}. \tag{6}$$

Then, given the significance level $\alpha = 0.05$, the statistic $\chi^2 = 2\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}|log\frac{c_{ij}}{p_{\cdot j}}|$ satisfies the limit distribution with a chi-square distribution with degrees of freedom $(p-1)^2$. If $\chi^2 > \chi^2_\alpha((p-1)^2)$ is satisfied, the state transition process of the system can be regarded as having Markov property. We use the built-in function of matlab to find $\chi^2_\alpha((p-1)^2)$ as the benchmark. We perform the above chi-square test based on the data of function node A, and the verification results are shown in figure 6. The horizontal axis represents the index of the host, and the vertical axis represents the comparison between the logarithmic calculated value and the benchmark value. Through verification, the hosts of each function node have Markov properties.

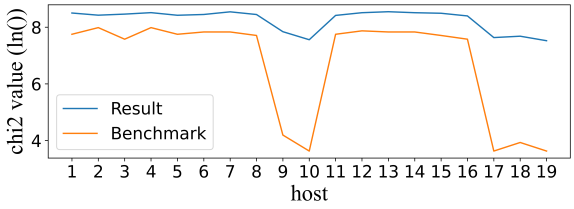

Fig. 6. PAFM corresponding to anomaly

IV. Experiments and Results

A. Analysis results of a single host

In the experimental data set, there are 19 hosts in function node A. The time period we observed was 8:30-11:30. In the statistics of $\boldsymbol{P}$, the length of the observation interval is 10s. That is, there are a total of 1080 observation intervals. The total number of features is $m = 16$. The definition of hyperparameters is as follows: $w_l = 30 \times 10s$. That is, each window contains 30 pieces of record ($n = 30$ in algorithm 1), and the window sliding step size $s_l = 1$.

Substituting the above dataset and hyperparameter settings into the algorithm, we are able to obtain features in function node A that reflect anomalies. The output results of our algorithm for selecting two hosts in function node A are shown in figure 7. The figure shows a sequence of boolean values where True value indicates that an anomaly was found in this time window. The analysis results of other function nodes will be reflected in the overall analysis of function nodes, and their anomaly sequences will not be repeated in this part.

B. Joint analysis of function nodes

In order to reveal the effect of algorithm 2, this part integrates the analysis results of function node A, function node B, C, D software calling data and function node B, C, D hardware monitoring data. We define $\boldsymbol{r}_i$ as the ratio sequence of hosts in a normal state and $\boldsymbol{r}_i$ reflects the health state of

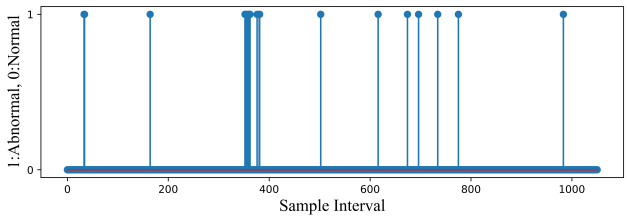

Fig. 7. Experimental results of host of node A

the function node. In order to analyze the correlation between the health status of each function node and the overall health status of the system, we use equation (7) to calculate the cross-correlation sequence between $\boldsymbol{r}_A$ sequence of function node A and $\boldsymbol{r}_{oth}$ sequences of other function nodes respectively. Figure 8 shows the calculation results of the cross-correlation sequence. It can be seen that the hardware monitoring score

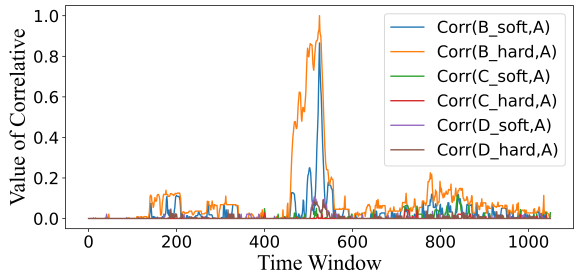

Fig. 8. PAFM corresponding to anomaly

sequence and software calling score sequence of function node B are strongly correlated with the overall score health sequence of the system. Therefore, we can conclude that the overall health of the system is greatly affected by the function node B (software, hardware), and the anomaly of the system can be prioritized for troubleshooting from the function node B. Combined with the characteristics of the financial securities business, UTC+8 9:00-10:00 is the opening time of the Chinese stock market, and a large number of users will send buying, selling or querying requests at this time. The above analysis results reflect the impact on the system of the function node B that undertakes user services during this time period.

$$\boldsymbol{R}[i] = \sum_{j=0}^{n-1} \boldsymbol{r}_A[j]\boldsymbol{r}_{oth}[j+i] \tag{7}$$

V. CONCLUSION

In this paper, we propose a Markov state-space model for complex high-dimensional data systems based on fuzzy integrals. Based on this model, we solved the problem of anomaly detection and root cause location of the back-end data system. On the basis of solving the problem, we first verify the physical meaning of the fuzzy measure in the system, and solve the problem of explainability. Then we verify that state transitions under similar business models are reusable. Finally, our experimental results show that the results with the host as the analysis granularity reflects the correlation between the upper-level modules. In actual operation scenarios, our analysis results predict the performance bottlenecks of the system during operation and improve the efficiency of troubleshooting when anomalies occur.

REFERENCES

[1] J. Huttunen, J. Jauhiainen, L. Lehti, A. Nylund, M. Martikainen and O. M. Lehner, "Big Data cloud computing and data science applications in finance and accounting", *ACRN Journal of Finance and Risk Perspectives*, vol. 8, pp. 16-30, 2019.

[2] B. Ford, "Icebergs in the clouds: The other risks of cloud computing," in *Proceedings of the USENIX HotCloud*, Boston, MA, June 2012, pp. 453–466.

[3] N. Zhao, B. Han, Y. Cai and J. Su, "SeqAD: An Unsupervised and Sequential Autoencoder Ensembles based Anomaly Detection Framework for KPI," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp.1-6, 2021.

[4] S Guo, Z Jin, Q Chen et al., "Visual Anomaly Detection in Event Sequence Data", in *2019 IEEE International Conference on Big Data (Big Data)*, pp.1125–1130, 2019.

[5] M. Munir, S. A. Siddiqui, A. Dengel and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series", in *IEEE Access*, vol. 7, pp. 1991-2005, 2018.

[6] Y. Meng, S. Zhang, Y. Sun, R. Zhang, Z. Hu, Y. Zhang, et al., "Localizing failure root causes in a microservice through causality inference", in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pp. 1-10, 2020.

[7] X. Guo, X. Peng, H. Wang, W. Li, H. Jiang, D. Ding, et al., "Graph-based trace analysis for microservice architecture understanding and problem diagnosis", in *ESEC/SIGSOFT FSE*, pp. 1387-1397, 2020.

[8] M. Ma, W. Lin, D. Pan, and P. Wang, "MS-rank: Multi-metric and selfadaptive root cause diagnosis for microservice applications," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, pp. 60–67, Jul. 2019.

[9] L. Tran, L. Fan and C. Shahabi, "Distance-based outlier detection in data streams", in *Proc. VLDB Endowment (PVLDB)*, vol. 9, no. 12, pp. 1089-1100, Aug. 2016.

[10] X. Qin, L. Cao, A. E. Rundensteiner and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams", in *Proc. 22nd Int. Conf. Extending Database Technol. (EDBT)*, pp. 421-432, Mar. 2019.

[11] A. Holzinger, G. Langs, H. Denk, K. Zatloukal and H. Müller, "Causability and explainability of artificial intelligence in medicine", *WIREs Data Mining Knowl. Discovery*, vol. 9, no. 4, pp. e1312, Jul. 2019.

[12] W. Samek, T. Wiegand and K.-R. Müller, "Explainable artificial intelligence: Understanding visualizing and interpreting deep learning models", in *ITU J. ICT Discoveries*, vol. 1, no. 1, pp. 39-48, 2018.

[13] A. Rai, "Explainable ai: from black-box to glass box," in *Journal of the Academy of Marketing Science*, vol. 48, no. 1, pp. 137–141, Jan 2020.

[14] Yang G, Ye Q, Xia J. "Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond", in *Information Fusion*, vol. 77, pp. 29-52, 2022.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning", in *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[16] A. R. Sambucini, "he Choquet integral with respect to fuzzy measures and applications," in *Math. Slovaca*, vol. 67, no. 6, pp. 1427–1450, 2017.

[17] T. C. Havens and D. T. Anderson, "Machine learning of Choquet integral regression with respect to a bounded capacity (or non-monotonic fuzzy measure)", in *Proc. IEEE Int. Conf. Fuzzy Syst.*, pp. 1-6, 2019.

[18] M. Siami, M. Naderpour and J. Lu, "A choquet fuzzy integral vertical bagging classifier for mobile telematics data analysis", in *Proc. IEEE Int. Conf. Fuzzy Syst.*, pp. 1-6, Jun. 2019.

[19] Z. Lin, K. Li, Y. Yang, F. Sun, L. Wu, P. Shi, et al., "Dresia: Deep reinforcement learning-enabled gray box approach for large-scale dynamic cyber-twin system simulation", in *IEEE Open Journal of the Computer Society*, vol. 2, pp. 321-333, 2021.

[20] F. A. Novikov, A. V. Veinmeister, E. V. Druyan, G. V. Belskii and A. R. Muzalevskiy, "Application of state transition graphs in control system for solenoid valve testing," in *2017 IEEE II International Conference on Control in Technical Systems (CTS)*, pp. 196-198, 2017.