Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 10, 2025

Research on Multi-scale Dynamic Evolution Time Series Modeling Methods for Predicting Backend System Load Change Trends

Netra Awate[1*], Ziang Chen[2]

[1]Northeastern University, Boston, USA

[2]University of Southern California, Los Angeles, USA

*Corresponding author: Netra Awate; netra.awate@gmail.com

Abstract: Backend system load prediction is a fundamental issue for ensuring stable service operation, optimizing resource scheduling efficiency, and enhancing system autonomy. Addressing the common challenges in backend system load sequences, such as multi-timescale coupling, complex local fluctuations, volatile global states, and difficulties in fully characterizing temporal dependencies, this paper proposes a multi-scale dynamic evolution time series modeling method for backend system load change prediction. This method first organizes the original load sequence into multiple scales to separate change information across different time ranges. Then, it extracts key representations at each scale through a multi-branch time series coding structure and utilizes an adaptive fusion mechanism to enhance the synergistic effect between features at different scales. Furthermore, a dynamic evolutionary state update unit is introduced to jointly model historical operating states and current load information, thereby enhancing the model's ability to express complex temporal structures and key change features. Research on real open-source backend operating data shows that the proposed method can more effectively mine deep correlations in load sequences, demonstrating good comprehensive capabilities in error control, prediction accuracy, and overall stability. This research provides a targeted modeling approach for backend system load prediction and offers a methodological foundation for tasks such as intelligent resource orchestration, service capacity configuration, and system operating status awareness.

Keywords: Temporal representation learning; load state modeling; adaptive feature fusion; cloud resource scheduling

1. Introduction

With the rapid development of the digital economy, cloud computing, and intelligent services, backend systems have become the core infrastructure supporting the continuous operation and service delivery of online businesses. Whether it's e-commerce transactions, content distribution, financial services, or industrial internet and government platforms, their business processing capabilities, resource scheduling efficiency, and service stability all highly depend on the backend system's ability to handle complex request traffic. In real-world operating environments, backend system load is not static but is influenced by multiple factors, including user access behavior, business cycle fluctuations, sudden event disturbances, system component coupling relationships, and resource contention, exhibiting significant dynamic, nonlinear, and time-varying characteristics[1,2]. Accurately characterizing and proactively sensing load change trends has become a crucial foundation for improving system reliability, ensuring service quality, and supporting intelligent operation and maintenance decisions[3].

The task of predicting backend system load change trends not only relates to elastic resource allocation and capacity planning but also directly impacts key aspects such as request scheduling, performance optimization, fault warning, and cost control. If the direction of load evolution and its potential patterns cannot be identified on time, the system is prone to problems such as increased response latency, decreased throughput, resource utilization imbalances, and even localized service anomalies during peak periods, thus adversely affecting overall business continuity[4]. Especially against the backdrop of deepening microservices, containerization, and distributed deployment, backend systems have longer internal call chains, more complex component interactions, and more hidden load propagation paths. Traditional methods relying on static thresholds or shallow statistical patterns are no longer sufficient to meet the trend perception needs in complex environments. Therefore, conducting more targeted predictive research on backend system load change trends has significant theoretical and practical value[5].

From a time series modeling perspective, backend system load data typically includes short-term fluctuations, medium-term cyclical changes, and long-term evolutionary trends. Change patterns at different time scales overlap and influence each other, collectively forming a holistic representation of the system's operational status[6]. On one hand, sudden requests and local hotspots can cause violent oscillations in a short period, making the load sequence exhibit high-frequency instability. On the other hand, business rhythms, user behavior habits, and system operation strategies can form relatively stable cyclical structures and trends over a longer period. This multi-scale, coexisting dynamic evolutionary characteristic makes single-scale, single-pattern modeling methods often insufficient to fully reveal the deep dependencies within the sequence, and also difficult to accurately capture key inflection points and trend reversals in the load change process. Therefore, exploring multi-scale dynamic evolution time series modeling methods that can take into account both local changes and global evolution has become an important research direction for improving prediction effectiveness[7].

Furthermore, constructing more adaptive time series modeling methods for predicting backend system load change trends not only helps promote the development of application scenarios such as intelligent operation and maintenance, autonomous computing, and resource optimization scheduling, but also provides more reliable data support and decision-making basis for the stable operation of complex software systems. By more meticulously characterizing the load change patterns at different time scales and strengthening the modeling capability of the system's dynamic evolution process, more forward-looking references can be provided for capacity prediction, service assurance, and risk control in high-concurrency environments. At the same time, this type of research also helps promote the deep integration of artificial intelligence methods and backend system operating mechanisms, driving the transformation of backend system management models from passive response to proactive prediction, and providing important theoretical support and application value for building a highly reliable, highly resilient, and highly intelligent modern backend service system.

2. Method

To address the strong nonstationarity, multi-frequency fluctuation, and evolving dependency patterns of backend system load sequences, a multiscale dynamic evolution time-series modeling framework is constructed to capture short-range variation, medium-range periodicity, and long-range trend transition within a unified predictive process.

This paper also presents the overall model architecture, as shown in Figure 1.

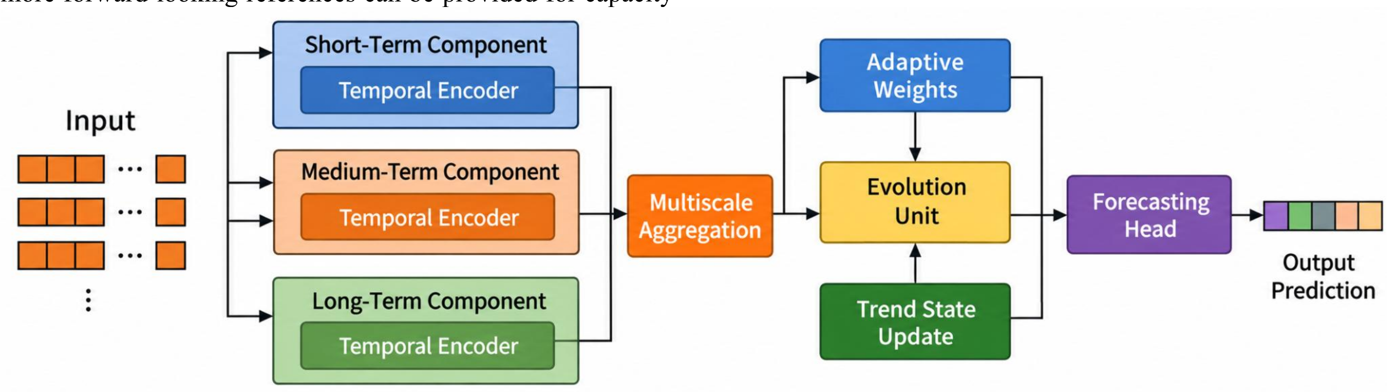

Figure 1. Overall model architecture

Let the historical load observations over an input horizon of length T be denoted by $\boldsymbol{x}_{1:T} = \{x_1, x_2, \ldots, x_T\}$, where each $x_t \in \mathbb{R}^d$ represents the system load state at time step t and may include response-oriented indicators such as throughput, utilization intensity, queue pressure, or aggregated service-level signals. Rather than directly learning a single temporal mapping from the raw sequence to the future trend, the method first reorganizes the original observations into multiple scale-specific views so that heterogeneous temporal patterns can be separated before interaction modeling. This design is important because backend load variation is usually produced by the superposition of bursty local disturbances and slower operational evolution, and forcing them into one latent stream often weakens the model's sensitivity to meaningful structural change. Formally, the multiscale input set is defined as:

$$\mathcal{S} = \left\{ \boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(K)} \right\}$$

where $\boldsymbol{X}^{(k)} \in \mathbb{R}^{T_k \times d}$ denotes the temporal representation at the kth scale and K is the number of scales. Given the raw load sequence, the scale-specific component is generated through a temporal aggregation operator $\phi_k(\cdot)$ with receptive field adapted to the corresponding observation granularity, yielding:

$$\boldsymbol{X}^{(k)} = \phi_k\left(\boldsymbol{x}_{1:T}\right)$$

so that fine-scale branches preserve abrupt local fluctuation while coarse-scale branches emphasize smoother trend movement. Such a decomposition step improves interpretability and stabilizes subsequent evolution modeling because the latent representation at each scale is encouraged to focus on a narrower temporal responsibility instead of mixing incompatible dynamics in the same channel.

At the representation learning stage, each scale branch is encoded independently to extract temporally contextualized states that reflect its own evolution rhythm. For the kth branch, a dedicated encoder $f_k(\cdot)$ transforms the input matrix $\boldsymbol{X}^{(k)}$ into the hidden sequence $\boldsymbol{H}^{(k)} = \left[\boldsymbol{h}_1^{(k)}, \boldsymbol{h}_2^{(k)}, \ldots, \boldsymbol{h}_{T_k}^{(k)}\right]$, where each hidden state summarizes both current load intensity and its nearby temporal interaction pattern. The mapping is written as:

$$\boldsymbol{H}^{(k)} = f_k\left(\boldsymbol{X}^{(k)}\right)$$

and the final scale summary vector is obtained by a temporal readout operator $\psi(\cdot)$ as:

$$\boldsymbol{z}^{(k)} = \psi\left(\boldsymbol{H}^{(k)}\right)$$

because a compact branch-level representation is beneficial for cross-scale coordination and suppresses redundant temporal noise. Unlike a rigid averaging mechanism that treats all scales equally, the framework introduces an adaptive importance estimator to measure the predictive relevance of each scale under the current system state, since different operational regimes may rely on different temporal resolutions. The normalized scale weight is therefore computed by:

$$\alpha_k = \frac{exp\left(\boldsymbol{q}^\top tanh\left(\boldsymbol{W}\boldsymbol{z}^{(k)} + \boldsymbol{b}\right)\right)}{\sum_{j=1}^{K} exp\left(\boldsymbol{q}^\top tanh\left(\boldsymbol{W}\boldsymbol{z}^{(j)} + \boldsymbol{b}\right)\right)}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable parameters, and $\boldsymbol{q}$ is a trainable context vector that guides scale selection. In this way, branches that better reflect the ongoing load evolution receive higher contribution weights, which strengthens model adaptability when the dominant temporal pattern shifts across periods.

Beyond simple fusion, the proposed framework models dynamic evolution explicitly so that the final predictive state is not only a weighted mixture of multiscale features but also a temporally updated trend descriptor. A global representation is first assembled as:

$$\boldsymbol{z} = \sum_{k=1}^{K} \alpha_k \boldsymbol{z}^{(k)}$$

and then injected into an evolution unit that maintains the trend memory across successive prediction windows. By introducing a gated transition mechanism, the model can decide how much historical trend evidence should be preserved and how much newly observed variation should be absorbed, which is crucial for backend systems whose load trajectories often undergo gradual regime migration instead of abrupt full-state replacement. The dynamic trend state $\boldsymbol{s}_t$ is updated through:

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \boldsymbol{s}_{t-1} + \left(1 - \boldsymbol{g}_t\right) \odot tanh\left(\boldsymbol{U}\boldsymbol{z}_t + \boldsymbol{V}\boldsymbol{s}_{t-1}\right)$$

where the gate $\boldsymbol{g}_t = \sigma\left(\boldsymbol{U}_g\boldsymbol{z}_t + \boldsymbol{V}_g\boldsymbol{s}_{t-1}\right)$ controls the balance between inherited trend memory and incoming multiscale evidence. Through this update process, the latent state becomes sensitive to both persistent operating tendencies and newly emerging pressure signals, thereby making the future estimate more consistent with the actual evolution logic of backend load variation.

Finally, the predicted load trend for the next horizon is generated from the evolution-aware state through a regression head $g(\cdot)$, and the forecasting target can be either a point estimate or a short sequence of future values, depending on the downstream scheduling requirement. For one-step prediction, the output is expressed as:

$$\widehat{\boldsymbol{y}}_{t+1} = g\left(\boldsymbol{s}_t\right)$$

while the training objective minimizes the discrepancy between prediction and ground truth by:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\|\widehat{\boldsymbol{y}}^{(i)} - \boldsymbol{y}^{(i)}\|_2^2 + \lambda \|\Theta\|_2^2$$

where N denotes the number of training samples, Θ collects all learnable parameters, and λ controls regularization strength for stable optimization. From a methodological perspective, this objective encourages the model to preserve predictive fidelity while preventing excessive sensitivity to transient noise, and the overall architecture therefore forms a coherent path from multiscale observation decomposition to dynamic evolution modeling and trend-oriented output generation. As a result, the method is able to provide a structurally grounded formulation for backend load trend prediction, with each stage designed to enhance temporal discrimination, state adaptability, and forecasting robustness under complex operational dynamics.

3. Experimental Results and Analysis

3.1 Dataset

This paper selects Alibaba Cluster Trace V2018 as the dataset. This dataset originates from large-scale cluster operation logs in a real production environment, is publicly released for backend systems and cloud data center scenarios, and possesses clear open-source accessibility and strong engineering realism. In terms of task relevance, this dataset records the co-location operation status of online services and batch processing workloads in a production cluster, and can realistically reflect the load fluctuations, resource competition, and temporal evolution of the backend system under complex business environments. Therefore, it has high consistency with backend system load change trend prediction tasks. Compared to general time series data or purely simulated load data, this dataset provides an observational basis that is closer to the actual system operation mechanism, helping to characterize the dynamic patterns and trend shifts of load changes in backend scenarios.

From a data content perspective, Alibaba Cluster Trace V2018 contains runtime information for approximately 4000 machines over eight consecutive days, comprised of six core files: machine_meta, machine_usage, container_meta, container_usage, batch_instance, and batch_task. Among these, machine_usage and container_usage directly support time-series modeling of CPU, memory, and other resource usage, while batch_instance and batch_task help analyze the impact of background task activities on system load changes. Furthermore, this dataset covers multi-level runtime information at the machine, container, and batch task levels, enabling the construction of load trend sequences from a global resource utilization perspective and facilitating the mining of potential drivers of backend system load evolution from a multi-source behavioral interaction perspective. Therefore, it is suitable as the foundational data source for this paper's research on multi-scale dynamic evolution time-series modeling.

3.2 Experimental Results and Analysis

To clearly present the relative performance levels of our proposed method and existing backend system load prediction research under a unified evaluation dimension, we select works closely related to backend system load change trend prediction, cloud workload modeling, and resource usage sequence prediction for cross-sectional analysis. These works cover different technical approaches, including statistical modeling, neural networks, attention mechanisms, online prediction, and graph evolutionary learning, and can comprehensively reflect the development trajectory and methodological differences in this field.

Table 1. Experimental results compared with other models

Method	MSE	MAE	RMSE	MAPE
Di et al.[8]	17.84	2.96	4.22	8.71
Cortez et al.[9]	15.63	2.71	3.95	8.14
Kumar et al. [10]	14.92	2.54	3.86	7.63
Kumar et al. [11]	13.74	2.36	3.71	7.21
Patel et al.[12]	12.98	2.24	3.60	6.94
Ullah et al.[13]	12.31	2.17	3.51	6.72
Li et al.[14]	11.64	2.05	3.41	6.39
Ours	10.82	1.93	3.29	6.11

As shown in Table 1, the proposed algorithm demonstrates superior overall performance across all evaluation metrics, indicating that it can more effectively characterize the key changing features in backend system load data and exhibits strong advantages in error control and prediction stability. This demonstrates that the constructed multi-scale dynamic evolutionary modeling mechanism can better coordinate information expression across different time ranges, enabling the model to have higher adaptability when facing complex load sequences, thereby obtaining more accurate prediction results.

Furthermore, the advantages of the proposed method are not only reflected in the improved overall prediction accuracy but also in the model's more comprehensive grasp of the inherent temporal structure of the load sequence. Since backend system loads often exhibit significant complexity, coupling, and dynamic changes, single-level modeling methods easily overlook deep temporal correlations. The proposed algorithm, by enhancing its ability to extract key evolutionary information, effectively improves the shortcomings of traditional methods in characterizing fine-grained changes. Therefore, the proposed algorithm has significant practical value and can provide a more reliable methodological support for backend system load prediction tasks.

Figure 2 illustrates how the time window length directly impacts the quality of backend system load information extraction by the proposed algorithm. Under the current configuration, the algorithm exhibits good stability and adaptability. As the input window is gradually adjusted, the model can more fully absorb the effective information from historical load sequences, thus keeping the prediction error at a relatively ideal level. This demonstrates that the constructed multi-scale dynamic evolution modeling approach can effectively coordinate local information with overall temporal correlation, possessing strong modeling capabilities in load prediction tasks.

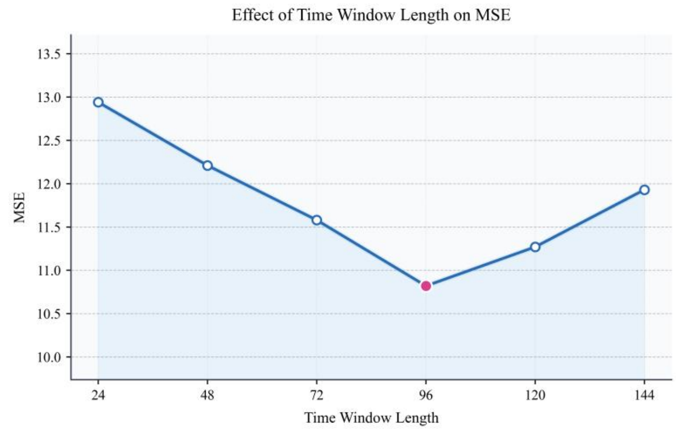

Figure 2. The impact of the time window length hyperparameter on MSE

When the time window is set within a more suitable range, the algorithm can more accurately capture key features in backend system load changes, achieving a better balance between information utilization and error control. If the window is too short, the historical context is insufficient, and the internal correlations of the sequence are difficult to fully represent. If the window is too long, redundant information is easily introduced, weakening the characterization of effective dynamic features. Therefore, a reasonable time window design plays a crucial role in leveraging the advantages of the proposed method, further demonstrating its application value in complex backend system load prediction scenarios.

4. Conclusion

This paper focuses on the key issue of backend system load change prediction. Addressing the complex volatility, temporal coupling, and dynamic evolution of load sequences in real-world operating environments, it proposes a multi-scale dynamic evolution time series modeling method for backend system load prediction. This method starts from the inherent characteristics of backend system load data, emphasizing the collaborative modeling of information representation across different time ranges. By effectively characterizing the evolution of time-series states, it enhances the model's ability to represent complex load patterns. Research shows that backend system load prediction is not a simple numerical extrapolation problem, but a comprehensive modeling task involving historical dependencies, local disturbances, stage-specific changes, and the combined effects of global operating states. Therefore, constructing a prediction framework that can consider multi-scale information and dynamic relationship changes is crucial for improving the perception of backend system operating states and supporting intelligent operation and maintenance decision-making.

From a methodological perspective, the proposed model emphasizes the unified modeling of multi-level time series features in its overall design. Through multi-scale input organization, branched time series representation learning, adaptive fusion, and dynamic evolution state updates, it achieves the layer-by-layer extraction and effective integration of backend system load information. Compared to modeling approaches that rely solely on a single temporal granularity or shallow temporal dependencies, this paper's method focuses more on the differentiated representation of load data across different time ranges and further strengthens the ability to continuously model key evolutionary states. This design enables the model to not only more fully identify load variation characteristics in complex business environments but also maintain good predictive performance in scenarios with strong temporal correlations and rapid state changes. Therefore, modeling around the dynamic structural characteristics of the load sequence itself is a crucial path to improving the quality of backend system load prediction and provides a more targeted research approach for this type of problem.

From an application perspective, this research has strong practical significance for resource management, capacity planning, performance assurance, and intelligent operation and maintenance of modern backend systems. With the continuous development of cloud computing platforms, distributed service architectures, containerized deployment methods, and high-concurrency online services, the backend system operating environment is becoming more complex, and the uncertainty and dynamism of system load are also increasing. In this context, a more accurate grasp of the changing patterns of load information can provide a more reliable decision-making basis for resource elastic allocation, service scaling and scheduling, anomaly risk warning, task orchestration optimization, and system stability assurance. Furthermore, the modeling approach proposed in this paper is not only applicable to backend system load prediction tasks, but also offers valuable insights for other infrastructure operation data analysis problems with complex temporal evolution characteristics. This means that the research results are expected to play a positive role in application areas such as cloud data center management, microservice performance optimization, edge computing resource scheduling, and intelligent operation and maintenance platform construction, thereby promoting the continuous evolution of backend system management from experience-driven to data-driven and prediction-driven approaches.

Future research still has considerable room for expansion. On the one hand, it is possible to further integrate richer operational information sources in the backend system, jointly modeling resource usage status, call relationship characteristics, task behavior information, and business context signals to enhance the model's adaptability to complex operating environments and its ability to understand potential load drivers. On the other hand, in-depth research can be conducted on model lightweighting, online update capabilities, and cross-scenario migration capabilities to better adapt to the comprehensive requirements of timeliness, stability, and deployment costs in real production environments. Simultaneously, if this type of method is further integrated with intelligent scheduling strategies, automatic scaling mechanisms, and anomaly warning modules, it is expected to form an integrated intelligent management framework for backend systems, encompassing state awareness, load prediction, and proactive decision-making. Overall, this study provides a relatively systematic modeling approach for the load prediction problem of backend systems, and lays a valuable theoretical foundation and methodological support for building a next-generation backend service system with higher reliability, stronger adaptability, and higher intelligence.

References

[1] A. Lackinger, A. Morichetta, and S. Dustdar, "Time series predictions for cloud workloads: A comprehensive evaluation," in 2024 IEEE International Conference on Service-Oriented System Engineering, 2024, pp. 36-45.

[2] L. Wen, M. Xu, A. N. Toosi, et al., "Temposcale: A cloud workloads prediction approach integrating short-term and long-term information," in 2024 IEEE 17th International Conference on Cloud Computing, 2024, pp. 183-193.

[3] B. Feng and Z. Ding, "Application-oriented cloud workload prediction: A survey and new perspectives," Tsinghua Science and Technology, vol. 30, no. 1, pp. 34-54, 2024.

[4] Y. Lin, J. Wen, X. Zhang, et al., "Resource time series analysis and forecasting in large-scale virtual clusters," Big Data Mining and Analytics, vol. 8, no. 3, pp. 592-605, 2025.

[5] A. Rossi, A. Visentin, D. Carraro, et al., "Forecasting workload in cloud computing: towards uncertainty-aware predictions and transfer learning," Cluster Computing, vol. 28, no. 4, p. 258, 2025.

[6] R. Karthikeyan, S. R. Abdul Samad, V. Balamurugan, et al., "Workload Prediction in Cloud Data Centers Using Complex-Valued Spatio-Temporal Graph Convolutional Neural Network Optimized With Gazelle Optimization Algorithm," Transactions on Emerging Telecommunications Technologies, vol. 36, no. 3, p. e70078, 2025.

[7] N. F. Siddiqui, Z. S. Hoque, M. E. Haque, et al., "SpectraNet: a lightweight hybrid time-frequency deep learning framework for sustainable cloud workload forecasting," Journal of Cloud Computing, 2025.

[8] S. Di, D. Kondo, and W. Cirne, "Host load prediction in a Google compute cloud with a Bayesian model," in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 2012, pp. 1-11.

[9] E. Cortez, A. Bonde, A. Muzio, et al., "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in Proceedings of the 26th Symposium on Operating Systems Principles, 2017, pp. 153-167.

[10] J. Kumar and A. K. Singh, "Workload prediction in cloud using artificial neural network and adaptive differential evolution," Future Generation Computer Systems, vol. 81, pp. 41-52, 2018.

[11] J. Kumar, A. K. Singh, and R. Buyya, "Self directed learning based workload forecasting model for cloud resource management," Information Sciences, vol. 543, pp. 345-366, 2021.

[12] Y. S. Patel and J. Bedi, "MAG-D: A multivariate attention network based approach for cloud workload forecasting," Future Generation Computer Systems, vol. 142, pp. 376-392, 2023.

[13] F. Ullah, M. Bilal, and S. K. Yoon, "Intelligent time-series forecasting framework for non-linear dynamic workload and resource prediction in cloud," Computer Networks, vol. 225, p. 109653, 2023.

[14] J. Li, D. Yang, H. Guo, et al., "Forecasting resource usage pattern changes in clouds via contrast graph-evolution learning," Future Generation Computer Systems, vol. 154, pp. 373-383, 2024.