China and WTO Review (CWR) Journal
2026, Vol(1)
ISSN: 2384-4388
https://cwto.net

Research Article

Behavioral Fingerprinting for Detecting Covert DNS Tunneling Activities in Enterprise Networks

Oliver Bennett [1], Amelia Clarke [1*]

[1] Department of Computing, Faculty of Engineering, Imperial College London, London, United Kingdom
* **Corresponding Author**: a.clarke@placeholder.ac.uk

ARTICLE INFO

Accepted: 19 Aug 2026

ABSTRACT

DNS tunneling is a common covert communication technique used to bypass traditional security monitoring systems and exfiltrate data through seemingly legitimate domain name queries. Because DNS traffic is usually allowed across enterprise and institutional networks, detecting tunneling behavior requires models that can identify subtle statistical and behavioral deviations rather than rely only on blacklist matching. This study proposes an explainable ensemble learning framework for DNS tunneling detection based on query behavior and flow-level features. The proposed model combines LightGBM, CatBoost, and Extremely Randomized Trees through a rank-weighted voting strategy. SHAP analysis is applied to explain how query length, domain entropy, subdomain repetition, response size, query interval, and NXDOMAIN ratio contribute to detection decisions. Experiments are conducted on a DNS traffic dataset collected from a controlled enterprise network with 6 recursive resolvers and 3,850 client hosts over 35 days. The dataset contains 5.64 million DNS query-flow records, including normal browsing, software update traffic, cloud service access, malware-generated domains, iodine-based tunneling, dnscat2 tunneling, and low-rate data exfiltration. After feature engineering, 46 features are retained for training and interpretation. The proposed model achieves 98.73% accuracy, 97.88% F1-score, and 99.21% AUC in binary detection. In multi-class detection, it obtains a macro-F1 of 96.42% across normal DNS, malware DNS, high-throughput tunneling, and low-rate tunneling classes. Compared with standalone CatBoost, the ensemble model reduces the false positive rate from 2.31% to 1.44% and improves recall for low-rate tunneling by 4.27%. SHAP results show that subdomain entropy, repeated long-label queries, abnormal NXDOMAIN ratio, and short periodic query intervals are the strongest indicators of DNS tunneling behavior. The results demonstrate that explainable ensemble learning can provide accurate and interpretable detection of covert DNS-based network anomalies.

Keywords: DNS Tunneling Detection, Covert Channel, Network Traffic Anomaly Detection, Explainable AI; SHAP, Ensemble Learning, Data Exfiltration, DNS Security

INTRODUCTION

The Domain Name System (DNS) is a basic service in enterprise, campus, cloud, and industrial networks. It translates domain names into IP addresses and supports almost all network applications, including web access, email services, software updates, cloud platforms, authentication systems, and industrial control communication. Because DNS is open, lightweight, and usually trusted by firewalls and security gateways, it is also attractive for covert communication. DNS tunneling hides command-and-control messages or stolen data inside DNS queries and responses, allowing attackers to bypass traditional monitoring systems that focus mainly on web, email, or direct outbound connections (Yin, Huang, & Rao, 2026; Banoth & Godishala, 2026). Recent work on policy-driven vulnerability risk quantification for large-scale cloud infrastructure has emphasized that security risk should be evaluated together with exposure context, operational constraints, and the potential consequence of vulnerable services (Shao, 2026). This perspective is also important for DNS tunneling detection because DNS abuse is not only a traffic anomaly, but also a hidden channel that exploits a trusted infrastructure service(Rao, Nayak, & Poornalatha, 2026; Yang, 2026).

DNS tunneling can appear in different forms, including high-throughput data transfer, low-rate exfiltration, command-and-control beaconing, and malware-driven domain communication. In high-throughput tunneling,

attackers may generate long subdomains, high query frequency, large response sizes, and abnormal query types (Ali, Afaq, Niazi, & Behzad, 2025; Deng & Yang, 2026). In low-rate tunneling, malicious queries may be sparse, short, and mixed with normal browsing or cloud service traffic, making detection more difficult. Early detection methods mainly relied on blacklists, domain reputation, query length, entropy, query frequency, response size, query type, and NXDOMAIN ratio (Rotună, Sacală, & Alexandru, 2025; Xu, Ma, Liu, & Yue, 2026). These methods are easy to deploy and useful for known malicious domains or obvious tunneling behaviors. However, they are weak against newly registered domains, unseen tunneling tools, encrypted payloads, low-rate communication, and payloads that imitate normal domain structures. Machine learning has become an important method for DNS tunneling detection because it can learn complex relations among lexical, temporal, and flow-level features (Gautam, Sapkota, Dawadi, & Shukla, 2026; Huang, 2026). Models such as support vector machines, random forests, gradient boosting, CatBoost, LightGBM, and deep learning methods have been used to identify abnormal DNS behavior. Lexical features can describe domain length, character distribution, entropy, label count, and subdomain repetition (Huang, 2026; Pelekoudas et al., 2026). Temporal features can capture query interval, burst behavior, periodicity, and low-frequency beaconing. Flow-level features can reflect response size, query type, resolver behavior, client activity, and NXDOMAIN patterns. By combining these features, machine learning models can detect suspicious DNS activities that cannot be captured by simple threshold rules (Wang, Wen, Wu, Wang, & Cai, 2025; Alorainy, 2025). Despite these advances, several limitations remain in existing DNS tunneling studies. Many studies still rely on small public datasets or traffic generated by only one or two tunneling tools (Zhu, Yao, & Yang, 2025; Razooqi & Pekar, 2025). Such settings may not reflect real enterprise networks where normal browsing, software updates, cloud services, malware DNS, high-throughput tunneling, and low-rate exfiltration coexist. Some datasets also contain overly clean separation between normal and malicious traffic, which may inflate reported accuracy and reduce practical reliability. In real networks, DNS traffic is highly diverse, class distribution is imbalanced, and benign services may also generate long domains, repeated queries, or unusual response patterns (Xue, Zi, Qi, Gong, & Zou, 2025; Mahmood, Abbas, Amjad, Shahid, & Shah, 2025). Therefore, models trained on narrow datasets may fail when deployed in operational environments with changing user behavior and service dependencies. Another important problem is the lack of clear explanation for model decisions. Deep learning and complex ensemble models can improve detection performance, but their outputs are often difficult for security analysts to verify (Ma, 2026; Ahmed et al., 2025). In real security operations, analysts need to understand whether an alert is caused by long subdomains, high domain entropy, repeated labels, abnormal NXDOMAIN ratio, short query intervals, unusual response size, or rare query types. Without interpretable evidence, even accurate alerts may be difficult to trust, prioritize, and investigate (Gao, Gao, Lu, Gao, & Kuang, 2026; Owen, Solomon, & Peter, 2025). Recent studies on explainable intrusion detection and DNS security have shown that SHAP-based feature analysis can help connect model outputs with readable evidence and improve analyst understanding (Zhang, 2026; Li & Liu, 2026). However, most DNS tunneling studies still focus mainly on accuracy, precision, recall, F1-score, and AUC, while giving limited attention to feature-level interpretation, multi-class detection, and operational alert validation.

To address these gaps, this study proposes an explainable ensemble learning framework for DNS tunneling detection based on query behavior and flow-level features. The proposed model combines LightGBM, CatBoost, and Extremely Randomized Trees through a rank-weighted voting strategy to improve robustness across different DNS traffic patterns. It is evaluated on a large DNS traffic dataset collected from a controlled enterprise network with 6 recursive resolvers and 3,850 client hosts over 35 days. The dataset includes normal DNS, malware DNS, iodine tunneling, dnscat2 tunneling, and low-rate data exfiltration, allowing the model to be tested under both high-throughput and stealthy tunneling scenarios. SHAP analysis is used to explain the contribution of query length, domain entropy, subdomain repetition, response size, query interval, NXDOMAIN ratio, and other key features to model decisions. The purpose of this study is not only to improve DNS tunneling detection accuracy, but also to provide clear and traceable evidence for covert-channel alerts. By combining ensemble learning with SHAP-based explanation, the proposed framework aims to support practical DNS security monitoring in enterprise networks and help analysts identify suspicious DNS behavior before it develops into serious data leakage or persistent command-and-control activity.

MATERIALS AND METHODS

Sample and Network Environment Description

The study used a DNS query-flow dataset collected from a controlled enterprise network for 35 consecutive days. The network included 6 recursive DNS resolvers and 3,850 client hosts. These hosts included office computers, virtual machines, internal application servers, software update terminals, and cloud access devices. After removing incomplete logs, repeated resolver records, and malformed packet entries, 5.64 million DNS query-flow records

were retained. The samples covered normal web browsing, enterprise login traffic, cloud service access, software update traffic, malware-related domain queries, iodine-based DNS tunneling, dnscat2 tunneling, and low-rate DNS data exfiltration. Each sample was treated as a query-flow unit rather than a single packet. This design allowed both domain features and short-term traffic behavior to be measured. The study focused on domain text features, response features, query timing, and resolver-level flow patterns.

Experimental Design and Control Settings

The experiment included a binary detection task and a multi-class classification task. In the binary task, DNS tunneling samples were labeled as tunneling traffic, while normal DNS and non-tunneling service traffic were used as the control group. This task tested whether the model could separate covert DNS channels from common enterprise DNS traffic. In the multi-class task, the samples were divided into normal DNS, malware DNS, high-throughput tunneling, and low-rate tunneling. This task tested whether the model could identify different abnormal DNS patterns. The experimental group included iodine tunneling, dnscat2 tunneling, and low-rate exfiltration flows. The control group included normal browsing, software updates, internal service access, and cloud platform DNS traffic. This setting was used because benign enterprise traffic may also contain long domains, repeated requests, and short query bursts. A strict control group was therefore needed to reduce false alarms caused by complex but normal DNS behavior.

Measurement Methods and Quality Control

DNS traffic was collected at the recursive resolver layer and converted into structured query-flow records. For each record, the source host, resolver identifier, query domain, query type, response code, response size, timestamp, and short-window query statistics were extracted. Domain-level measurements included query length, label number, maximum label length, repeated subdomains, character diversity, and domain entropy. Flow-level measurements included query interval, query burst density, NXDOMAIN ratio, response-size change, and repeated-domain frequency within a fixed time window. Quality control was conducted in four steps. First, records with missing timestamps, invalid domain strings, or abnormal resolver identifiers were removed. Second, repeated logs caused by resolver forwarding were filtered using timestamp, source host, queried domain, and response code. Third, tunneling samples were checked against controlled attack periods to avoid wrong labels. Fourth, benign high-frequency services, such as software updates and cloud synchronization, were retained. This helped test the model under realistic enterprise traffic conditions.

Data Processing and Model Formulation

After preprocessing, 46 features were retained for model training and explanation. Continuous features were standardized by z-score scaling. Categorical features, such as query type and response code, were converted into numerical variables. The dataset was divided into training, validation, and test subsets by host and time window. This reduced overlap between similar query sequences. Domain entropy was used to measure the randomness of characters in a queried domain. It was calculated as (1):

$$H(d)=-\sum_{i=1}^{n} p_i \log_2(p_i) \tag{1}$$

where $H(d)$ is the entropy of domain d, p_i is the probability of the *i-th* character, and n is the number of unique characters in the domain. A higher entropy value indicates a more random domain string, which often appears in encoded tunneling payloads. Model performance was measured by accuracy, precision, recall, F1-score, and AUC. The F1-score was calculated as:

$$F1=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{2}$$

where precision is the proportion of detected tunneling samples that are true tunneling samples, and recall is the proportion of true tunneling samples correctly detected by the model.

Ensemble Learning and Model Explanation

The detection model combined LightGBM, CatBoost, and Extremely Randomized Trees through a rank-weighted voting strategy. Each base classifier produced a class probability score for each DNS query-flow record. The class ranks from the three models were then weighted according to validation performance. The class with the highest final rank score was used as the final label. LightGBM was used to learn nonlinear feature relations. CatBoost was used to handle mixed feature distributions in a stable way. Extremely Randomized Trees was used to lower variance through random split selection. The ensemble model was compared with each single classifier to test whether the combined structure improved detection stability. SHAP analysis was used to explain feature effects on model outputs. Global SHAP values were used to identify the most important indicators, while local SHAP values were used to explain individual alerts. This allowed the model to show whether a tunneling decision was mainly caused by high subdomain entropy, repeated long labels, abnormal NXDOMAIN ratio, short periodic query intervals, or unusual response-size patterns.

RESULTS AND DISCUSSION

Binary Detection Performance

The proposed ensemble model achieved stable results in binary DNS tunneling detection. The accuracy, F1-score, and AUC were 98.73%, 97.88%, and 99.21%, respectively. These values show that the model could separate DNS tunneling traffic from normal DNS traffic with high reliability. The false positive rate decreased from 2.31% for standalone CatBoost to 1.44% for the ensemble model. This result is important because normal enterprise services, such as software updates, cloud synchronization, and login systems, may also produce long domains and repeated DNS requests (Alabdulatif, 2025; Zhang, Gao, & Tong, 2026). A single classifier may mistake these benign but irregular records for tunneling traffic. By combining LightGBM, CatBoost, and Extremely Randomized Trees, the model reduced unstable decisions and captured different feature patterns. Compared with earlier DNS tunneling detection methods, this model used both query behavior and flow-level features instead of relying only on packet length or a small group of selected variables. The overall detection result is shown in **Figure 1**.

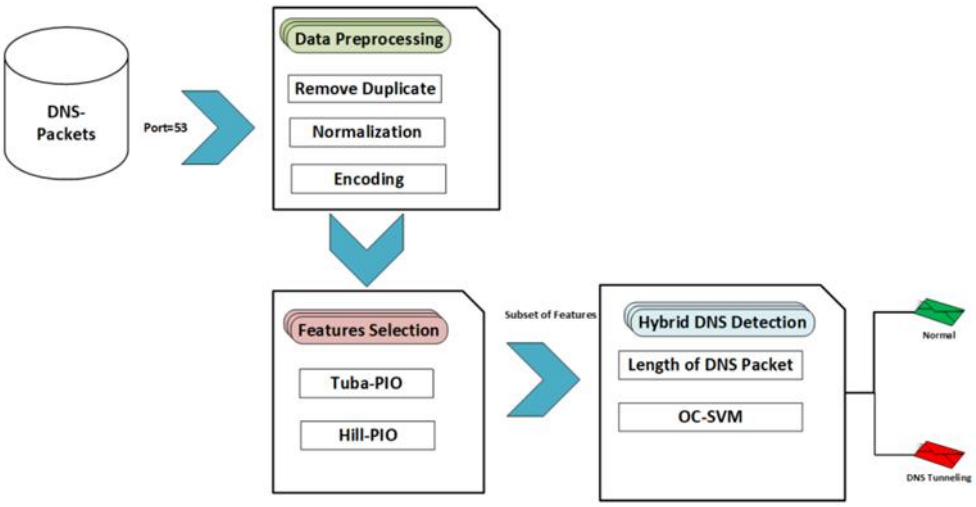

Figure 1. Performance of the Proposed Ensemble Model In Binary And Multi-Class DNS Tunneling Detection.

Multi-Class Classification Results

The multi-class task was more difficult than binary detection because the four traffic classes had partly similar DNS behavior. Normal DNS, malware DNS, high-throughput tunneling, and low-rate tunneling may all contain repeated queries or short traffic bursts. The proposed model achieved a macro-F1 of 96.42%, indicating good performance across different traffic classes. High-throughput tunneling was easier to detect because it produced longer domain labels, higher entropy, denser query bursts, and more regular request intervals. Low-rate tunneling was more difficult to identify because it reduced query frequency and payload size to imitate normal DNS traffic. Even under this condition, the ensemble model improved low-rate tunneling recall by 4.27% compared with standalone CatBoost. This result shows that rank-weighted voting helped detect weak abnormal signals (Mahalakshmi & Chandra, 2026; Qi & Qiao, 2026). Compared with previous studies that mainly focused on known

tunneling samples, this study further separated high-throughput tunneling from low-rate tunneling. This setting is closer to real network defense because low-rate exfiltration may remain hidden for a longer time.

Feature Contribution and Model Explanation

The feature analysis showed that subdomain entropy, repeated long-label queries, abnormal NXDOMAIN ratio, short periodic query intervals, and response-size variation were the main indicators of DNS tunneling. These features have clear security meanings. High entropy often appears when data are encoded into domain labels. Repeated long labels may indicate segmented payload transmission. A high NXDOMAIN ratio may be caused by generated or temporary domains that fail to resolve. Short and regular query intervals usually reflect automated communication rather than human browsing. The SHAP results showed that the model did not rely on one single feature. Instead, it made decisions based on lexical, temporal, and response-level signals together. This helps reduce false alarms because benign cloud services may also generate long or frequent DNS queries (Iqbal, Butt, Shareef, Siddique, & Browne, 2026; Su & Wu, 2026). The explanation process is shown in **Figure 2.** Compared with black-box detection models, this method provides clearer evidence for analysts when they check suspicious DNS alerts.

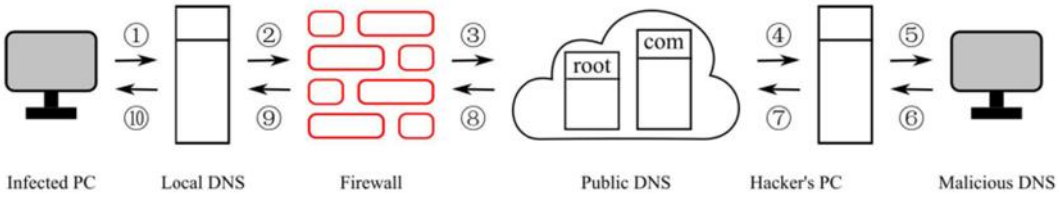

Figure 2. SHAP Analysis of Key Features Used for DNS Tunneling Detection

Comparison with Existing Studies and Practical Implications

Compared with earlier DNS tunneling detection methods, the proposed framework has practical value in three aspects. First, the dataset included normal browsing, software update traffic, cloud access, malware-related DNS, iodine tunneling, dnscat2 tunneling, and low-rate exfiltration. This setting reduced the risk of testing the model on a narrow dataset. Second, the ensemble model improved stability by combining classifiers with different learning patterns. Third, SHAP analysis linked each alert with readable DNS evidence, including entropy, label repetition, NXDOMAIN ratio, query interval, and response size. Existing machine learning methods have shown that DNS tunneling can be detected through traffic features, but many of them still focus on binary detection or limited attack types. The proposed framework extends this direction by combining binary detection, multi-class recognition, and feature-level explanation. The main limitation is that the dataset was collected from a controlled enterprise network. Future studies should test the model on DNS logs from different organizations, encrypted DNS traffic, and adaptive tunneling tools that change query timing and domain structure.

CONCLUSION

This study presented an explainable ensemble learning method for DNS tunneling detection based on query behavior and flow-level features. The model combined LightGBM, CatBoost, and Extremely Randomized Trees through rank-weighted voting, which gave more stable results than single classifiers. The model achieved 98.73% accuracy, 97.88% F1-score, and 99.21% AUC in binary detection. In the multi-class task, it reached a macro-F1 of 96.42% for normal DNS, malware DNS, high-throughput tunneling, and low-rate tunneling. It also reduced false positives and improved the recall of low-rate tunneling, which is important for finding slow and hidden data leakage. SHAP analysis showed that subdomain entropy, repeated long labels, abnormal NXDOMAIN ratio, short query intervals, and response-size changes were the main signs of DNS tunneling. These results show that the method can give accurate detection results and clear evidence for security analysts. It may be used in enterprise DNS monitoring, threat hunting, and early warning of covert channels. The main limitation is that the data were collected from a controlled enterprise network. Future work should test the method on DNS logs from different organizations, encrypted DNS traffic, and tunneling tools with changing query patterns.

REFERENCES

Ahmed, U., Nazir, M., Sarwar, A., Ali, T., Aggoune, E. H. M., Shahzad, T., & Khan, M. A. (2025). Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. *Scientific Reports, 15*(1), 1726.

Alabdulatif, A. (2025). A novel ensemble of deep learning approach for cybersecurity intrusion detection with explainable artificial intelligence. *Applied Sciences, 15*(14), 7984.

Ali, F., Afaq, M., Niazi, M., & Behzad, M. (2025). From Graphs to Gates: DNS-HyXNet, A Lightweight and Deployable Sequential Model for Real-Time DNS Tunnel Detection. *arXiv preprint arXiv:2512*.09565.

Alorainy, W. S. (2025). Echoes From the Void: Detecting DNS Tunneling With Blackhole Features in Encrypted Scenarios With High Accuracy. *IEEE Access*.

Banoth, R., & Godishala, A. K. (2026). Attacking the Foundation, Attacking What We Do, Understanding Defense. In *Building a Secure Infrastructure: The key Procedures for System Network Security* (pp. 373-415). Cham: Springer Nature Switzerland.

Deng, X., & Yang, J. (2026). Design of a Quantitative Futures Trading Model Incorporating Multimodal Feature Fusion and Tail Risk Control. Available at SSRN 6702398.

Gao, G., Gao, R., Lu, C., Gao, R., & Kuang, Y. (2026, March). Security Governance Methods and Quantitative Evaluation for Enterprise SMS and Verification Code Systems. In *2026 International Conference on Generative Artificial Intelligence and Information Security (GAIIS)* (pp. 455-458). IEEE.

Gautam, S., Sapkota, B., Dawadi, B. R., & Shukla, U. (2026). Detecting DNS tunneling-based data exfiltration using machine learning. *Journal of Innovations in Engineering Education, 9*(1), 105-119.

Huang, R. (2026). ConfSched: Confidence-Threshold Routing for Efficient Edge-Cloud Activity Recognition. *World Journal of Engineering and Technology, 14*(2), 471-486.

Iqbal, M., Butt, T., Shareef, Z., Siddique, A., & Browne, W. N. (2026). A Systematic Interaction Analysis of Distance Measures and Feature Representations in Mammography. *IEEE Access*.

Li, Y., & Liu, S. (2026, May). A Study on Dynamic Optimization of Alerting Policies and Multi-Agent Decision-Making Mechanisms in Cloud Environments. In *2026 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)* (pp. 703-706). IEEE.

Ma, Y. (2026). Industrial Financial Risk Prediction Model Based on Graph Neural Network and Knowledge Graph Inference. *Journal of Circuits, Systems and Computers, 35*(16), 2650103.

Mahalakshmi, A., & Chandra, G. (2026). *Architecting a High-Availability and Secure Three-Tier Web Infrastructure using AWS Services* (Doctoral dissertation, Dublin, National College of Ireland).

Mahmood, A., Abbas, H., Amjad, F., Shahid, W. B., & Shah, S. Q. A. (2025). Real-Time Detection and Multi-Class Classification of DGAs with HybridBERT. *IEEE Access*.

Owen, A., Solomon, M., & Peter, L. (2025). Trust and Transparency in Human-AI Collaboration: Building Reliable Real-Time Alerting Systems.

Pelekoudas, A. P., Bolis, E., Lindner, J., Kyriakidis, P., Davidsen, M., Hansen, J. T., ... & Homayoun, S. (2026). TLS Certificate and Domain Feature Analysis of Phishing Domains in the Danish. dk Namespace. *arXiv preprint arXiv:2603*.21652.

Qi, C., & Qiao, X. (2026). Using AI to Monitor AI: Automated Operations Through Log-Driven Intelligence. Available at SSRN 6795840.

Rao, K. P., Nayak, S. H., & Poornalatha, G. (2026). A Comprehensive Review of DNS Tunneling Detection Methods in Cybersecurity. *IAENG International Journal of Computer Science, 53*(3), 947.

Razooqi, Y. S., & Pekar, A. (2025). Vpn traffic analysis: A survey on detection and application identification. *IEEE Access, 13*, 132830-132848.

Rotună, C. I., Sacală, I. Ş., & Alexandru, A. (2025). Towards Proactive Domain Name Security: An Adaptive System for. ro domains Reputation Analysis. *Future Internet, 17*(10), 478.

Shao, W. (2026). Policy-Driven Vulnerability Risk Quantification framework for Large-Scale Cloud Infrastructure Data Security. *arXiv preprint arXiv:2604*.06252.

Su, D., & Wu, Y. (2026). Multilevel Growth and Social Network Modeling for Functional Communication Enhancement in Students with Intellectual Disabilities.

Wang, Y., Wen, Y., Wu, X., Wang, L., & Cai, H. (2025). Assessing the Role of Adaptive Digital Platforms in Personalized Nutrition and Chronic Disease Management.

Xu, S., Ma, Q., Liu, H., & Yue, L. (2026). Continuous Reorganization and Performance Preservation of Agent Memory Structure Under Distributed Change Environments.

Xue, Z., Zi, Y., Qi, N., Gong, M., & Zou, Y. (2025, August). Multi-level service performance forecasting via spatiotemporal graph neural networks. In *2025 5th International Conference on Intelligent Communications and Computing (ICICC)* (pp. 202-206). IEEE.

Yang, J. (2026). Stage-Coupled Computational Framework for Stratified Accessibility and Equity Analysis in Community-Based Elderly Care Services.

Yin, J., Huang, Y., & Rao, H. (2026). A Study on User Behavior Signal-Driven Predictive Models for Digital Platform Consumption Trends. Available at SSRN 6607298.

Zhang, Z. (2026). A Study on Multimodal Product Understanding and Assembly Guidance Optimization in e-commerce for Complex Consumer Goods. Available at SSRN 6734559.

Zhang, Z., Gao, Y., & Tong, Y. (2026). Semantic-Temporal Graph Learning for Demand Forecasting and Resource Allocation in Public Information Systems. Available at SSRN 6792279.

Zhu, W., Yao, Y., & Yang, J. (2025). Optimizing Financial Risk Control for Multinational Projects: A Joint Framework Based on CVaR-Robust Optimization and Panel Quantile Regression.

ETHICAL DECLARATION

Conflict of interest: No declaration required. **Financing:** No reporting required. **Peer review:** Double anonymous peer review.