GANFL: A log anomaly detection method based on collaborative optimization of federated learning and generative adversarial networks

Longxin Yao YLX15138699692@GS.ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou, China

Xuanran Li LIXUANRAN@GS.ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou, China

Mingzhe Li 1351147614@QQ.COM
School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou, China

Bo Zhang* ZHANGBO2050@ZZU.EDU.CN
School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou, China

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

With the rapid development of information technology, the amount of data is growing explosively. Enterprises and society have an increasing demand for data storage, processing and analysis. Data centers have emerged as the times require. They can centrally manage massive amounts of data, provide efficient computing and storage capabilities, meet the high requirements of different industries for data processing, and ensure data security and reliability. In data centers, numerous devices, systems and applications continuously generate a large number of logs during operation. These logs record the activities and status information at all levels of the data center, including the operating status of the server, the traffic of network devices, and the operation records of applications. Log anomalies refer to the presence of records that do not conform to normal patterns or expected content in the log files that record the operating system's own operating events. Log analysis can help developers quickly locate the source of the fault. By analyzing the log data, they can determine the device where the fault occurred and the cause of the fault. At the same time, they can also conduct advance analysis based on the existing log data to discover potential problems.In this paper, the method of co-optimization of GAN and federated learning is adopted, which not only solves the problem of data silos, but also solves the problem of insufficient data.

Keywords: Federated Learning; GAN; Log exceptions; Data silos

1. Introduction

Federated learning (FL) (Wang et al., 2021), (Wang et al., 2024) uses a distributed architecture to protect data privacy: the client calculates the model update based on the local data, and the server aggregates to generate the global model (as shown in Figure 1). In this paper, a dynamic weighted aggregation mechanism is introduced to adjust the weight of parameters according to the data quality score to improve the compatibility of heterogeneous log formats. It supports local model fine-tuning and adaptation of node characteristics to ensure detection accuracy and real-time performance in a multi-tenant environment, reduce communication costs and maintain data security.

The Generative Adversarial Network (GAN) (Ding et al., 2022) generates data through adversarial training of generators and discriminators, and its objective function optimizes the data by

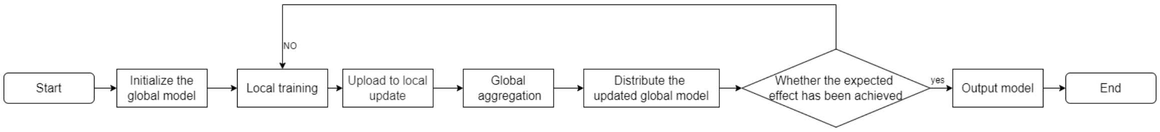

Start
Initialize the global model
Local training
Upload to local update
Global aggregation
Distribute the updated global model
Whether the expected effect has been achieved
NO
yes
Output model
End

Figure 1: Flowchart of federated learning model training.

minimizing cross-entropy loss. Traditional GANs rely on JS divergence, resulting in unstable training, and fail when the generation does not overlap with the real distribution. In this paper, the Wasserstein GAN (WGAN) (Wang et al., 2023) is used to effectively measure the non-overlapping distribution difference by using the Wasserstein distance (Earth-Mover distance), and the generator loss directly reflects the sample quality. WGAN-GP (Zhao et al., 2025) is further introduced to significantly improve the training stability and generation diversity by adding gradient penalty terms to the discriminator loss instead of weight clipping strategy.

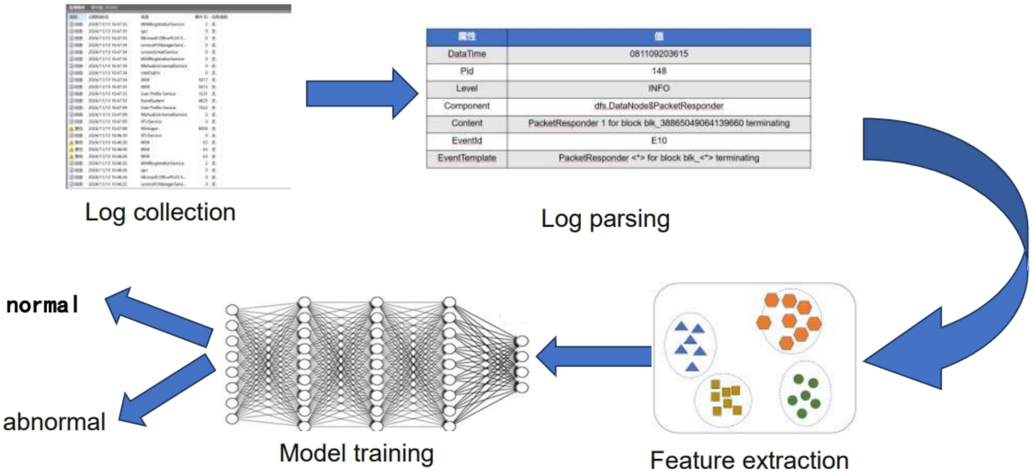
BERT (Guo et al., 2021) is a bidirectional pre-trained language model based on Transformer, which significantly improves natural language understanding ability by simultaneously learning context information. Its core architecture consists of a multi-layer Transformer encoder, which is pre-trained through a masked language model and next sentence prediction tasks to generate dynamic word vectors to solve the problem of polysemy. In log analysis scenarios, BERT can be used for structured parsing and semantic feature extraction of log texts to enhance the semantic representation ability of abnormal events.

Our framework employs federated learning for localized model training with encrypted parameter sharing, ensuring privacy and reduced communication overhead. To address data scarcity and anomaly complexity, WGAN-GP generates realistic log samples via noise injection, expanding rare anomaly datasets. Its gradient-penalized discriminator enforces Lipschitz continuity, mitigating mode collapse while reducing false positives through adversarial distribution learning.

2. Related Work

Operating system log anomaly detection is a technology that automatically identifies abnormal behavior by analyzing system logs. Logs record user activities, service status, and security events such as login failures, resource anomalies, or malicious processes, aiming to discover potential attacks, system failures, or performance issues. Its process includes log collection, parsing, feature extraction, and distinguishing anomalies based on rules, statistics, or machine learning models.

The log anomaly detection process can be summarized as follows as shown in Figure 2: First, collect the original logs from the system or application, clean the redundant information through preprocessing and unify the format; then enter the structured parsing stage, disassemble the key attributes, and convert the unstructured text into computer-recognizable labeled data. Next, extract multidimensional features from the parsing results through feature extraction, encode them into feature vectors and input them into the model. The model training stage uses neural networks to learn the distribution law of normal logs and optimize parameters through multi-user data on the server side. Finally, after the new log is processed by the same process, the model makes anomaly judgments based on the learned patterns (outputs normal/abnormal labels) to help quickly locate potential failures or security threats. The entire process is connected in series with modular design, and the arrows point to clear logical links, taking into account both efficiency and explainability.



属性	值
DataTime	081109203615
Pid	148
Level	INFO
Component	dfs.DataNode$PacketResponder
Content	PacketResponder 1 for block blk_38865049064139660 terminating
EventId	E10
EventTemplate	PacketResponder <*> for block blk_<*> terminating

Figure 2: The whole process of log anomaly detection is conducted.

Anomaly detection of logs through an unsupervised approach (Farzad and Gulliver, 2020) combining dual deep autoencoders and isolated forest techniques. After the preprocessing of the original log, the dual deep autoencoder is used for unsupervised feature learning: the encoding layer (L1 regularization) extracts the latent features, and the decoding layer reconstructs the input to capture the key patterns. The feature input isolated forest screens high-confidence positive samples, strengthens the normal log representation through iterative training, and finally dynamically sets the anomaly threshold based on the feature standard deviation to achieve efficient unsupervised detection.

Through semantic clustering and sentiment analysis, log anomaly detection is optimized (Liu et al., 2025), and a pre-trained semantic embedding model and sentiment classification model are used to construct a mixed feature space for spectral clustering, and the original log sequence is compressed into a clustered label sequence and then inputted LSTM modeling, so that its false positive rate is reduced.

Traditional centralized machine learning faces the risk of data privacy leakage and single point of failure (Ning et al., 2024). In this paper, the federated learning architecture is used to realize multi-agency privacy collaborative modeling through encrypted parameter transmission, and the adversarial samples are generated by WGAN-GP to solve the problems of data heterogeneity and annotation scarcity.

3. Method

The pre-trained BERT model is utilized to semantically embed log text and capture contextual information. Log text is input and transformed into structured feature vectors through multimodal feature fusion. A global model and client-specific sub-models are trained using federated learning methodology, as depicted in Figure 3.

The global anomaly detection model is initialized on the central server with the parameter θglobal(θ) and distributed to all clients. Each client k trains the model using a local dataset with the goal of minimizing the loss function. Each server generates abnormal data based on the architecture of the generative adversarial network. The JS divergence of the original GAN can easily lead

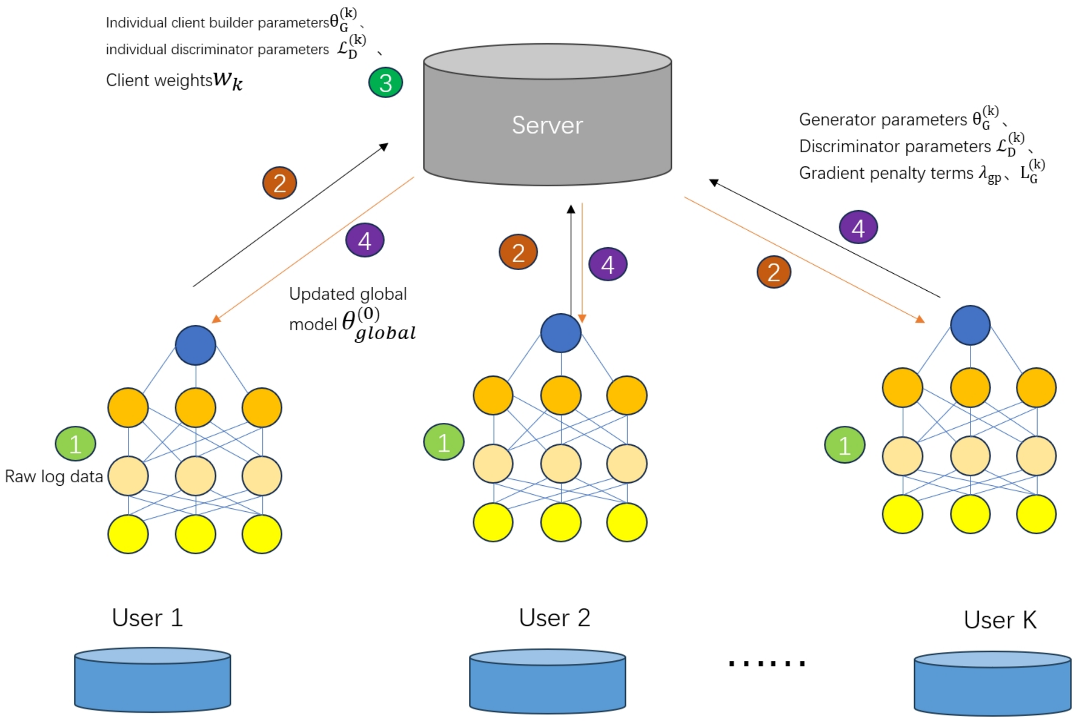

Individual client builder parameters$\theta_G^{(k)}$、
individual discriminator parameters $\mathcal{L}_D^{(k)}$ 、
Client weightsw_k
3
Server
Generator parameters $\theta_G^{(k)}$、
Discriminator parameters $\mathcal{L}_D^{(k)}$、
Gradient penalty terms λ_{gp}、 $L_G^{(k)}$
2
4
Updated global model $\theta_{global}^{(0)}$
2
4
4
2
1
Raw log data
1
1
User 1
User 2
......
User K

Figure 3: Federated Learning Parameter Transfer Diagram.

to gradient disappearance or schema crash, especially in the federated scenario of heterogeneous data, it is difficult for the generator to cover the real data distribution of all clients. So replacing the generator loss function with the Wasserstein GAN (WGAN) form and adding the gradient penalty term is:

$$L_G^{(k)} = \mathbb{E}_{z \sim P_z}[-D(G(z; \theta_G^{(k)}))] \tag{1}$$

In Eq.(1), $L_G^{(k)}$: The generator loss at the k-th client. Indicates the training objective of the generator at client k, which should be minimized. $\mathbb{E}_{z \sim p_z}$: Expectation over the noise variable z following distribution p_z (typically the standard normal distribution $N(0, I)$), used to generate random noise inputs. $G(z; \theta_G^{(k)})$: Generation process of generator G, taking noise z as input and outputting adversarial samples $\widetilde{x}$, with parameters $\theta_G^{(k)}$ (generator parameters at the k-th client). $D(\cdot)$: Probability output by the discriminator, representing the likelihood of an input sample being real log data (values in $[0, 1]$).

To prevent discriminator overfitting, a gradient penalty term is introduced to enforce the discriminator's gradient norm at interpolated samples to be close to 1, thereby satisfying the Lipschitz continuity condition of Wasserstein GAN (WGAN) and improving training stability. This significantly enhances the diversity of generated samples and training stability. Discriminator loss function:

$$L_G^{(k)} = \mathbb{E}_{x \sim D_k}[-D(x)] + \mathbb{E}_{z \sim p_z}[D(G(z; \theta_G^{(k)}))] + \lambda_{\text{gp}} \mathbb{E}_{\hat{x} \sim \hat{p}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{2}$$

In equation 2, $L_D^{(k)}$: The discriminator loss at the k-th client. Minimize this loss to optimize the discriminator. $\mathbb{E}_{x \sim D_k}$: Expectation over real log data x from the local dataset D_k at client k. λ_{gp}:

Gradient penalty coefficient controlling the weight of the gradient penalty term. Larger values impose stronger constraints on the discriminator's gradient norm. $\hat{x}$: Interpolated samples. Generated by linear interpolation between real samples x and generated samples $G(z)$, used to compute the gradient penalty term. This enforces smooth gradient transitions in the discriminator between real and generated data at interpolation points, preventing overfitting to either. $\nabla_{\hat{x}} D(\hat{x})$: Gradient of the discriminator at interpolated samples. Computes the partial derivative of the discriminator's output with respect to input $\hat{x}$, reflecting the discriminator's sensitivity at interpolation points.

$(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2$ gradient penalty term. Enforces the discriminator's gradient norm at interpolated samples to be close to 1. Thus satisfying the Lipschitz continuity condition of WGAN and improving training stability.

After training is completed, clients send model updates (such as gradients or changes in model weights) to the central server instead of raw data. Upon receiving updates from multiple clients, the central server integrates these updates using an aggregation algorithm to update the global model. Clients are evaluated for data quality, with higher scores assigned to those with higher accuracy, larger data volumes, and distributions closer to the global one:

$$q_k = \alpha \cdot \text{Accuracy}_k + \beta \cdot \frac{n_k}{N} + \gamma \cdot \text{KL}(\mathcal{D}_k \| \mathcal{D}_{\text{global}}) \tag{3}$$

The data quality scores are mapped to weights via the Softmax function. The global model is aggregated with weighted contributions, prioritizing parameters from clients with higher data quality (larger q_k) to enhance the global model's adaptability to complex log environments:

$$w_k = \frac{\exp(q_k/T)}{\sum_{j=1}^{K} \exp(q_j/T)} \tag{4}$$

Parameters from high-quality clients dominate the global model update, while the influence of low-quality clients is suppressed. The global model parameter update formula is:

$$\theta_{\text{global}} = \sum_{k=1}^{K} \frac{w_k}{\sum_{j=1}^{K} w_j} \theta_k \tag{5}$$

Here, q_k is the data quality score of the k-th client, computed by weighted accuracy, data volume proportion, and data distribution similarity. θ_k is the aggregation weight of the k-th client, obtained through Softmax function normalization.

$\text{KL}(\mathcal{D}_k \| \mathcal{D}_{\text{global}})$ represents the KL divergence between the data distribution $\mathcal{D}_k$ of the k-th client and the global data distribution $\mathcal{D}_{\text{global}}$, measuring distribution discrepancy.

The server sends the updated global model back to each client. Clients continue local training using this model for the next round. This process iterates until the model achieves the desired performance or accuracy. Ultimately, the system performs log data analysis line-by-line in log environments to determine data anomalies.

4. Experiment

4.1. Datasets

Datasets. We evaluate the proposed GANFL on two log datasets, Hadoop Distributed File System (HDFS) and BlueGene/L(BGL). Table 1 shows the statistics of the datasets. For the HDFS dataset,

Table 1: The amount of data in HDFS and BGL

Dataset	Log Messages	Anomalies	of Log Sequences in Test Dataset	
			Normal	Anomalous
HDFS	11,172,157	284,818	558,223	16,746
BGL	4,747,963	348,460	10,045	2,630

we use 167,466 log data for training, and use 111,644 data to test the performance of GANFL with less data. Table 1 shows the amount of data in HDFS and BGL.

The HDFS dataset is generated by running Hadoop-based mapreduce jobs on Amazon EC2 nodes and manually labeled to identify anomalies using handcrafted rules.[9] The HDFS dataset contains 11,172,157 log messages, of which 284,818 are anomalous messages. For HDFS, we group log keys into log sequences based on the session ID in each log message. The average length of the log sequence is 19.

The BGL dataset is collected from the BlueGene/L supercomputer system at Lawrence Livermore National Laboratory (LLNL). (Ren et al.) The logs contain alert and non-alert messages identified by alert category labels. These alert messages are considered anomalies. The BGL dataset contains 4,747,963 log messages, of which 348,460 are anomalies.

Baselines. We compare the GANFL model with the following baselines.

- One-Class Support Vector Machine (OCSVM) (Zheng et al., 2023). One-class SVM is a well-known one-class classification model that can be deployed for log anomaly detection by constructing a feature matrix based on normal data.

- LogCluster (Vaarandi et al., 2016). LogCluster is a clustering-based method in which abnormal log sequences are detected by having a long distance from normal clusters.

- DeepLog (Du et al., 2017). DeepLog is a state-of-the-art log anomaly detection method. DeepLog adopts a recurrent neural network to capture the patterns of normal log sequences and further identifies abnormal log sequences based on the performance of log keyword prediction.

- LogAnomaly (Meng et al., 2019). Log Anomaly is a deep learning-based anomaly detection method that can detect serialized and quantitative log anomalies.

4.2. Experimental results

To verify the performance of the model, the experiment compares the anomaly detection effects of GANFL and multi-class baseline methods using only 20% of the data volume of two public datasets,experimental results on HDFS and BGL datasets as shown in Table 2.

This study evaluates model performance across diverse scenarios using three key metrics: Precision (quantifying prediction reliability by minimizing false positives), Recall (assessing anomaly detection completeness through reduced missed positives), and F1 Score (harmonic mean balancing both metrics). Experimental results demonstrate that conventional machine learning approaches (e.g., OCSVM) exhibit suboptimal F1 scores (below 30%), attributable to inherent architectural

Table 2: Experimental results on HDFS and BGL datasets

Method	HDFS			BGL		
	Precision	Recall	F-1 score	Precision	Recall	F-1 score
OCSVM	2.98	100.00	5.788	9.18	13.45	10.912
LogCluster	99.34	32.47	48.943	97.42	53.64	69.186
DeepLog	85.46	69.45	76.628	89.73	79.54	84.328
LogAnomaly	92.41	71.21	80.437	70.46	72.74	71.582
LogBert	87.02	74.12	80.054	74.50	69.30	71.806
GANFL	86.32	78.80	82.389	88.93	84.39	86.601

constraints: reliance on statistical feature engineering and dimensionality reduction techniques fundamentally limits their capacity to reconcile detection accuracy with recall rates, while inducing substantial performance instability across datasets. Conversely, the domain-specific LogCluster framework achieves a 40% F1 improvement over conventional methods through clustering-optimized feature representation, confirming the efficacy of log-aware architectural specialization. Deep learning architectures (DeepLog, LogAnomaly, LogBERT) demonstrate state-of-the-art performance with F1 scores surpassing 70%, benefiting from sophisticated temporal dependency modeling capabilities. Notably, LogBERT addresses historical information degradation limitations through two innovative self-supervised pre-training tasks specifically designed for normal log sequence reconstruction, establishing new benchmarks in log-based anomaly detection.

However, limitations persist in these models, as evidenced by LogAnomaly's 15% lower recall than the proposed method on the BGL dataset, revealing deficiencies in complex pattern recognition. The proposed model attains optimal cross-dataset performance, achieving an average F1 score improvement of over 3% compared to the second-best baseline.

5. Conclusion

This paper proposes GANFL, a federated log anomaly detection framework integrating FL, WGAN-GP, and BERT through collaborative optimization. The FL architecture enables local model training with encrypted parameter sharing, eliminating raw data exposure risks while ensuring privacy. To address data heterogeneity, dynamic federated aggregation prioritizes high-quality clients through data-driven weighting. For annotation scarcity, our WGAN-GP generates distribution-aligned synthetic logs via noise inputs, resolving mode collapse via gradient-penalized discriminators to ensure data authenticity. The BERT integration enhances contextual pattern mining, collectively optimizing privacy-data-detection synergy.

References

Jue Ding, Jun Yin, Jingyu Dun, Wanwan Zhang, and Yayun Wang. Attacking frequency information with enhanced adversarial networks to generate adversarial samples. In *Advances in Visual Computing: 17th International Symposium, ISVC 2022, San Diego, CA, USA, October 3–5, 2022, Proceedings, Part I*, page 61–73. Springer-Verlag, 2022. doi: 10.1007/978-3-031-20713-6_5.

Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference* *on Computer and Communications Security*, page 1285–1298. Association for Computing Machinery, 2017. doi: 10.1145/3133956.3134015.

Amir Farzad and T. Aaron Gulliver. Unsupervised log message anomaly detection. *ICT Express*, 6 (3):229–237, 2020. doi: https://doi.org/10.1016/j.icte.2020.06.003.

Haixuan Guo, Shuhan Yuan, and Xintao Wu. Logbert: Log anomaly detection via bert. 2021.

Zhiwei Liu, Xiaoyu Li, and Dejun Mu. Log anomaly detection and diagnosis method based on deep learning. *Int. J. Data Min. Bioinformatics*, 29(1–2):119–132, January 2025. doi: 10.1504/ijdmb.2025.142978.

Weibin Meng, Ying Liu, Yichen Zhu, and et al. Loganomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 4739–4745. AAAI Press, 2019.

Weiguang Ning, Yingjuan Zhu, Caixia Song, and et al. Blockchain-based federated learning: A survey and new perspectives. *Applied Sciences*, 14(20), 2024. doi: 10.3390/app14209459.

Rui Ren, JieChao Cheng, Hao Shi, and et al. Failure characterization based on lstm networks for bluegene/l system logs. Intelligent Computing and Block Chain, pages 123–133. Springer Singapore.

Risto Vaarandi, Markus Kont, and Mauno Pihelgas. Event log analysis with the logcluster tool. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 982–987, 2016. doi: 10.1109/MILCOM.2016.7795458.

Chan Wang, Zixian Dong, Wei Hu, and et al. Classification of IoT intrusion detection data based on WGAN-gp and E-GraphSAGE. In Hongzhi Wang and Shiling Zhang, editors, *Third International Conference on Green Communication, Network, and Internet of Things (CNIoT 2023)*, volume 12814, page 1281416. SPIE, 2023. doi: 10.1117/12.3010362.

Chenghong Wang, Jieren Deng, Xianrui Meng, and et al. A secure and efficient federated learning framework for NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7676–7682, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.606.

Liang Wang, Yilin Li, and Lina Zuo. Trust management for iot devices based on federated learning and blockchain. *The Journal of Supercomputing*, 81(1):232, 2024. doi: 10.1007/s11227-024-06715-4.

Mingguan Zhao, Xinsheng Dong, Yang Yang, and et al. A dynamic prediction approach for wire icing thickness under extreme weather conditions based on wgan-gp-rtabnet. *CMES - Computer Modeling in Engineering and Sciences*, 142(2):2091–2109, 2025. doi: https://doi.org/10.32604/cmes.2025.059169.

JiaMing Zheng, Jie Fu, KunSan Zhang, Yongji Zhang, and TaiNing Huang. Research and application of traffic anomaly detection technology based on single class support vector machine ocsvm. In *2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 11, pages 99–103, 2023. doi: 10.1109/ITAIC58329.2023.10408835.