Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 1, 53–65
DOI: 10.61822/amcs-2026-0005

ROOT CAUSE ANALYSIS OF TEMPORAL NETWORK FAULTS USING ECHO STATE NETWORKS

BIXIAN ZHANG [a,*], YIXIA CHEN [a]

[a] School of Electronic Engineering
Fuzhou Institute of Technology
No. 8, Binjiang Road, Fuzhou 350506, China
e-mail: bxzhang_fit@163.com

The increasing complexity and dynamism of modern networks pose significant challenges for effective fault management. Temporal network faults, characterized by their evolving nature and cascading effects, are particularly difficult to diagnose. Traditional root cause analysis (RCA) methods often struggle with the high dimensionality, non-linearity, and temporal dependencies inherent in network monitoring data. This paper proposes a novel framework, ESN4TRCA—an echo state network for temporal root cause analysis, for identifying the root causes of temporal network faults. ESN4TRCA leverages the inherent capabilities of echo state networks (ESNs), a paradigm of reservoir computing, in modeling complex temporal dynamics with remarkably low training overhead. We formulate the temporal RCA problem as a sequence classification task, where sequences of multivariate key performance indicators (KPIs) and alarm data are mapped to their underlying root causes. The proposed framework encompasses modules for data preprocessing, ESN model construction specifically tailored for heterogeneous network fault data, and a robust inference mechanism. We introduce specific mathematical formulations for the leaky-integrator reservoir dynamics and the output weight training via ridge regression, optimized for the RCA context. Comprehensive experiments are conducted on both a synthetic dataset generated using the NS-3 network simulator and a real-world public dataset. The results demonstrate that ESN4TRCA significantly outperforms state-of-the-art RCA methods, including traditional machine learning approaches and other recurrent neural network architectures like LSTMs and GRUs, in terms of accuracy, F1-score, and robustness to noise, while maintaining superior computational efficiency. The study highlights the potential of ESNs as a powerful and practical tool for advanced automated network fault management.

Keywords: network fault management, root cause analysis (RCA), temporal dynamics, echo state network (ESN), reservoir computing, recurrent neural network (RNN), machine learning, automated network operation.

1. Introduction

Modern communication networks form the bedrock of our digital society, underpinning a multitude of critical services (Agrawal and Singh, 2023). Ensuring the reliability and availability of these networks is paramount. However, the escalating scale, complexity, heterogeneity, and dynamic nature of contemporary networks, propelled by technologies such as software-defined networking (SDN), network functions virtualization (NFV), and 5G/6G, render them susceptible to a wide array of faults (Chi *et al.*, 2022; Kiadehi *et al.*, 2021; Wang *et al.*, 2025). Failure to promptly and accurately diagnose and resolve network faults can precipitate severe service disruptions, substantial economic losses, and a degraded user experience (Haleem Medattil Ibrahim *et al.*, 2024).

Root cause analysis (RCA) stands as a critical pillar of network fault management, striving to pinpoint the underlying origin of observed anomalous network behaviors or symptoms (Soldani and Brogi, 2022). This endeavor is far from trivial, beset by several inherent challenges. A primary difficulty lies in symptom propagation and cascading effects, where a single root fault can unleash a torrent of alarms and performance degradations across numerous network elements, thereby obscuring the true source (Yang, 2024). Furthermore, network faults and their manifestations often exhibit temporal dynamics; intermittent faults, performance degradation culminating in eventual failure, and transient

*Corresponding author

issues necessitate the analysis of evolving patterns over time (Zhou *et al.*, 2020a). The sheer volume and velocity of monitoring data generated by modern networks, encompassing key performance indicators (KPIs), logs, and alarms, also demand highly efficient processing techniques (Reddy *et al.*, 2024). Compounding these issues is the often highly non-linear and intricate relationship between root causes and observed symptoms, rendering rule-based or simplistic statistical models inadequate.

Traditional RCA methodologies, such as expert systems (Chung *et al.*, 1989; Arévalo *et al.*, 2019; Steenwinckel *et al.*, 2021) and fault trees (Singh *et al.*, 2012; Wang *et al.*, 2021b; Chen *et al.*, 2022), frequently depend on predefined rules or models that are arduous to construct and sustain in fluid, dynamic environments. Probabilistic graphical models like Bayesian networks (Wee *et al.*, 2015) offer a more principled approach but struggle with the complexity of model construction and inference in large-scale networks. While machine learning (ML) techniques, including decision trees and support vector machines (SVMs), have found application in RCA (Agrawal *et al.*, 2017; Detzner *et al.*, 2020; Ma *et al.*, 2021), they often treat observations as independent and identically distributed, failing to adeptly capture the long-range temporal dependencies intrinsic to network fault scenarios.

Deep learning (particularly recurrent neural networks (RNNs) like long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent units (GRUs) (Salem, 2022)), has demonstrated considerable promise in modeling sequential data (Chung *et al.*, 2014; Lindemann *et al.*, 2021; Xu *et al.*, 2019; Ahmed *et al.*, 2022; Niu *et al.*, 2023). However, training these deep RNN architectures via backpropagation through time (BPTT) is computationally burdensome and protracted, posing a significant barrier for near real-time operational environments and continuous model retraining.

Echo state networks (ESNs) (Jaeger, 2002), a notable paradigm within reservoir computing, present an appealing alternative. ESNs are characterized by a large, fixed, randomly generated recurrent reservoir of neurons whose internal connections remain untrained. Training is confined to the connections from this reservoir to the output layer (the "readout"), typically accomplished via efficient linear regression. This architectural choice dramatically curtails training complexity while preserving the capacity to capture rich temporal dynamics (Gonon and Ortega, 2021; Racca and Magri, 2021). ESNs have already achieved state-of-the-art performance in a variety of time-series prediction and classification tasks (Zhang *et al.*, 2019; Ullah *et al.*, 2022; Sun *et al.*, 2024).

In this paper, we introduce ESN4TRCA, a novel framework leveraging echo state networks for root cause analysis of temporal network faults. Our work offers several key contributions:

- We formulate the temporal network RCA problem as a sequence classification task amenable to ESNs, considering streams of heterogeneous network KPIs and alarm data.
- We design a specialized ESN architecture tailored for network RCA, encompassing principled approaches for input representation, reservoir construction, and an output layer designed for precise fault identification.
- We develop a comprehensive mathematical model of the ESN within the RCA context, meticulously detailing the leaky-integrator reservoir dynamics and the readout training process using ridge regression for enhanced generalization.
- We conduct extensive experiments on a realistically simulated network environment and a public benchmark dataset, providing a thorough comparative analysis of ESN4TRCA against traditional ML methods, SOTA RNN architectures (LSTM, GRU), and other advanced techniques.

Our results show that ESN4TRCA not only achieves high accuracy but also offers a superior trade-off between performance and computational cost, making it a highly practical solution for modern network operations.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 provides essential background. Section 4 meticulously details the proposed ESN4TRCA framework. Section 5 describes the experimental design. Section 6 presents and discusses the findings. Lastly, Section 7 concludes the paper.

2. Related work

This section surveys existing research in network root cause analysis and pertinent applications of echo state networks.

2.1. Traditional and model-based RCA. Early RCA approaches often relied on predefined models of network behavior and fault propagation. Expert systems encoded human knowledge into rules (Fink *et al.*, 1985; Ives, 1991), proving effective for known faults but lacking adaptability. Fault propagation models, encompassing techniques like fault trees (Zhu, 2008; Waghen and Ouali, 2019; Ruijters and Stoelinga, 2015; Detzner *et al.*, 2020), dependency graphs (Shah *et al.*, 2018; Wang *et al.*, 2021a; Patel *et al.*, 2021), and codebooks (Kim *et al.*, 2011; Liu *et al.*, 2019), aimed to model relationships between components, often requiring accurate topology and predefined dependencies. Bayesian networks and other probabilistic graphical models (Alaeddini and Dogan,

2011) utilized probabilistic reasoning to infer root causes; while powerful, constructing their graphical models and conditional probability tables is challenging for complex networks. These methods generally struggle with the temporal evolution of faults and the high dimensionality of modern network data.

2.2. Machine learning-based RCA. With the availability of large volumes of monitoring data, ML techniques have become popular. Supervised learning algorithms such as decision trees (Singh *et al*., 2012; Ferreira and Vasilyev, 2015) and SVMs (Al-Mamory and Zhang, 2009; Abdelrahman and Keikhosrokiani, 2020; Tong *et al*., 2021) have been employed to classify fault types based on labeled historical data, though they often use aggregated features, potentially losing crucial temporal information. Unsupervised learning, through clustering algorithms like DBSCAN (Schubert *et al*., 2017; Singh *et al*., 2022), can group similar alarm patterns to identify correlated symptoms. Alarm correlation techniques, such as frequent pattern mining (Hu *et al*., 2018; Li *et al*., 2019) or statistical correlation (de Abreu *et al*., 2021; Tao *et al*., 2021; Seyed Alinezhad *et al*., 2022), aim to reduce alarm floods. A stronger non-deep-learning baseline for time series classification is k-nearest neighbors (k-NN) using dynamic time warping (DTW) as the distance metric, which can effectively align and compare temporal sequences of varying lengths (Lee *et al*., 2012; Petitjean *et al*., 2016; Tran *et al*., 2019). However, many of these ML methods do not inherently model the long-range temporal dependencies critical for diagnosing evolving faults.

2.3. Deep learning for RCA. Deep learning models have recently gained traction for their ability to learn complex patterns from raw data. Convolutional neural networks (CNNs) have been used for RCA by transforming network data into image-like structures to capture spatio-temporal correlations (Li *et al*., 2022; Wang *et al*., 2024). Recurrent neural networks (RNNs), particularly LSTMs and GRUs, are naturally suited for sequential data like time-series KPIs, capable of learning long-term dependencies (Zaheer *et al*., 2023; Yadav and Thakkar, 2024; Kong *et al*., 2025). They have shown success in network RCA, albeit with resource-intensive training via BPTT. More recently, graph neural networks (GNNs) have emerged as a SOTA approach, as they explicitly leverage network topology information to model fault propagation over the network graph, achieving impressive results (Wu *et al*., 2022; Yen *et al*., 2022; Wang *et al*., 2023; Jiang and Bai, 2023). GNNs, however, require accurate and up-to-date topology information, which may not always be available, and their training can also be complex.

2.4. Echo state networks in time series analysis. ESNs, as a cornerstone of reservoir computing, have proven effective for various time series tasks. They were initially proposed for chaotic time series prediction (Jaeger, 2002) and have since been applied to diverse tasks such as financial forecasting (Sun *et al*., 2021; Trierweiler Ribeiro *et al*., 2021) and weather prediction (De *et al*., 2023). ESNs can also be adapted for time series classification by training the readout to map the reservoir's state dynamics to class labels (Wang *et al*., 2022). While ESNs are less explored in network RCA compared to LSTMs/GRUs, some applications exist in related areas like network traffic prediction (Zhou *et al*., 2020b; 2022) and anomaly detection (Ullah *et al*., 2022; De Vita *et al*., 2023). Our work addresses a significant gap by systematically designing and evaluating ESNs for the complex task of temporal network fault RCA. We posit that ESNs' unique advantages—fast training and proficient temporal modeling—offer a compelling trade-off between performance and computational cost, which is crucial for (near) real-time network management systems.

3. Preliminaries and problem formulation

3.1. Network and fault models. We model the monitored network as a system comprising N_c components $C = \{c_1, c_2, \ldots, c_{N_c}\}$, which can be physical entities (routers, switches, links) or logical constructs (services, virtual machines). At discrete time steps $t = 1, 2, \ldots, T$, the network's state is observed through a set of N_m monitoring metrics, denoted by $\mathbf{m}(t) \in \mathbb{R}^{N_m}$. These metrics include continuous KPIs (e.g., CPU utilization, latency, the packet loss rate) and discrete alarms/events (e.g., link down, service unresponsive), which are often encoded numerically.

A temporal network fault is an event or a sequence of causally related events stemming from one or more root cause components, leading to service degradation. These faults manifest as anomalous patterns within a sequence of observations $\{\mathbf{m}(1), \mathbf{m}(2), \ldots, \mathbf{m}(T_w)\}$ over a time window of length T_w. Let $\mathcal{F} = \{f_0, f_1, \ldots, f_{N_f-1}\}$ represent the set of N_f possible root causes, where f_0 denotes a "no fault" state and f_i for $i > 0$ signifies a specific fault type.

The objective of temporal network RCA is to identify the true underlying root cause $f^* \in \mathcal{F}$ given a sequence of network observations. Formally, let $\mathbf{S}_k = (\mathbf{m}_k(t_0), \ldots, \mathbf{m}_k(t_0 + T_w - 1))$ be the k-th sequence of observations. The goal is to learn a mapping function $\Phi : \mathcal{S} \to \mathcal{F}$, where $\mathcal{S}$ is the space of all possible observation sequences, such that

$$\hat{f}_k = \Phi(\mathbf{S}_k), \tag{1}$$

with $\hat{f}_k$ being the predicted root cause. This is fundamentally a multivariate time series classification problem.

3.2. ESN fundamentals. An ESN consists of an input layer, a fixed recurrent reservoir layer, and a trainable output layer (readout). Given an input sequence $\mathbf{u}(1),\ldots,\mathbf{u}(T_w)$, where $\mathbf{u}(t) \in \mathbb{R}^{N_u}$ (derived from $\mathbf{m}(t)$), the reservoir state $\mathbf{x}(t) \in \mathbb{R}^{N_x}$ is updated according to the leaky-integrator ESN equations:

$$\tilde{\mathbf{x}}(t) = \tanh(\mathbf{W}_{in}[1;\mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)), \quad (2)$$

$$\mathbf{x}(t) = (1-\alpha)\mathbf{x}(t-1) + \alpha\tilde{\mathbf{x}}(t), \quad (3)$$

where $[1;\mathbf{u}(t)]$ is the input vector augmented with a bias, $\mathbf{W}_{in} \in \mathbb{R}^{N_x\times(1+N_u)}$ is the input weight matrix, $\mathbf{W} \in \mathbb{R}^{N_x\times N_x}$ is the internal reservoir weight matrix, $\tanh(\cdot)$ is the activation function, and the parameter $\alpha \in (0,1]$ is the leaking rate, controlling the speed of reservoir dynamics and memory.

The matrices $\mathbf{W}_{in}$ and $\mathbf{W}$ are randomly generated and fixed. For the reservoir to have the crucial "echo state property," the spectral radius $\rho(\mathbf{W})$ (maximum absolute eigenvalue of $\mathbf{W}$) is scaled to be less than 1. This property ensures that the reservoir's state is a unique function of the input history, preventing it from generating its own chaotic dynamics, thus making the system stable and predictable. Note that $\mathbf{W}$ is often a sparse matrix.

The output $\mathbf{y}(t) \in \mathbb{R}^{N_y}$ is a linear combination of the reservoir state:

$$\mathbf{y}(t) = \mathbf{W}_{out}[1;\mathbf{x}(t)], \quad (4)$$

where $\mathbf{W}_{out} \in \mathbb{R}^{N_y\times(1+N_x)}$ is the output weight matrix, which is the only part of the ESN that undergoes training. This is achieved by collecting the reservoir's final states for a set of training sequences and solving a linear regression problem, frequently using ridge regression to mitigate overfitting.

Let $\mathbf{Z}_{coll}$ be the matrix formed by collecting the final augmented state vectors, $[1;\mathbf{x}_k(T_w)]$, for each training sequence k, and let $\mathbf{Y}_{target}$ be the matrix of the corresponding target outputs. Then, $\mathbf{W}_{out}$ is found by solving a linear regression problem:

$$\mathbf{W}_{out} = \arg\min \|\mathbf{Z}_{coll}\mathbf{W}_{out}^T - \mathbf{Y}_{target}\|_F^2 + \beta\|\mathbf{W}_{out}\|_F^2, \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius norm and $\beta > 0$ is the regularization coefficient. The closed-form solution is

$$\mathbf{W}_{out}^T = (\mathbf{Z}_{coll}^T\mathbf{Z}_{coll} + \beta\mathbf{I})^{-1}\mathbf{Z}_{coll}^T\mathbf{Y}_{target}, \quad (6)$$

where $\mathbf{I}$ is the identity matrix. For sequence classification, the target output is typically defined only at the end of the sequence, $t = T_w$.

4. Proposed ESN4TRCA framework

We introduce ESN4TRCA, an echo state network-based framework designed for temporal root cause analysis. The framework's architecture is depicted in Fig. 1.

4.1. Data preprocessing and feature engineering. Raw monitoring data $\mathbf{m}(t)$ requires careful preprocessing:

- *Aggregation*: KPIs and alarms are aggregated into fixed time intervals, Δt_s.
- *Normalization*: Continuous KPIs are normalized to the range [-1, 1] using min-max scaling, with scaling parameters derived from the training set, to prevent features with large magnitudes from dominating the model.
- *Encoding*: Categorical alarm data (type, severity, source) is converted to a numerical format using one-hot encoding.
- *Concatenation*: Normalized KPIs and encoded alarms at time t are concatenated to form the input vector $\mathbf{u}(t) \in \mathbb{R}^{N_u}$.
- *Sequencing*: Input sequences $\mathbf{S}_k = (\mathbf{u}_k(1),\ldots,\mathbf{u}_k(T_w))$ are generated using a sliding window of length T_w and step T_{step}. Here, T_w is a critical hyperparameter that determines the temporal context.

4.2. ESN model design for RCA. The core of ESN4TRCA is an ESN specifically configured for RCA. We select the leaky-integrator model (Eqns. (2)–(3)) as it provides explicit control over the memory timescale via the leaking rate α. This is vital for capturing the diverse temporal characteristics of network faults, from abrupt spikes to gradual degradations.

4.2.1. Input layer. The input weight matrix $\mathbf{W}_{in} \in \mathbb{R}^{N_x\times(1+N_u)}$ maps the input vector to the reservoir. Its elements are drawn from a uniform distribution $\mathcal{U}(-s_{in}, s_{in})$, where the input scaling factor s_{in} is a key hyperparameter. This random projection is a deliberate and crucial design choice for network data. Network monitoring streams are high-dimensional and heterogeneous, comprising diverse KPIs and alarm types. A random, uniform projection distributes this input information across the reservoir's neurons without making strong prior assumptions about which features are most important, thereby ensuring a rich and diverse excitation of the reservoir's dynamics.

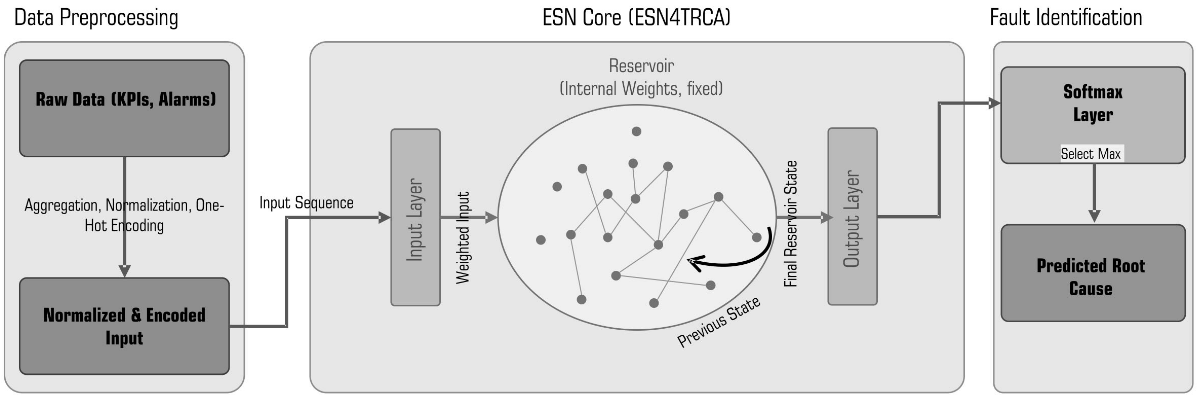


Fig. 1. Overall architecture of the ESN4TRCA framework, consisting of data preprocessing, the ESN core (input, reservoir, readout), and fault identification modules.

4.2.2. Reservoir design. The reservoir's properties are crucial for performance. The size (N_x) determines the model's capacity: larger N_x offers greater representational power at the cost of computation. The sparsity means the internal weight matrix $\mathbf{W}$ is sparsely connected (e.g., 1–10% connectivity) to create diverse internal dynamics. This sparsity is particularly advantageous for network RCA, as it fosters a rich set of loosely coupled, nonlinear dynamics within the reservoir. This architecture is analogous to the behavior of a communication network itself, where different subsystems can behave independently but also influence each other during cascading failures. This allows the reservoir to effectively model both localized and widespread fault patterns simultaneously. After random initialization, $\mathbf{W}$ is scaled so its spectral radius $\rho(\mathbf{W})$ is slightly less than 1 (e.g., 0.95) to ensure stability and the echo state property. The leaking rate (α) controls the memory capacity of the reservoir, balancing influence from new inputs and past states. The reservoir state is updated according to Eqns. (2) and (3).

4.2.3. Output layer (readout). For RCA classification, the readout layer maps the reservoir's final state to one of the N_f root cause classes. After processing an input sequence $\mathbf{S}_k$, we use the augmented final state as the feature vector:

$$\mathbf{z}_k = [1; \mathbf{x}_k(T_w)]. \tag{7}$$

The readout matrix $\mathbf{W}_{out} \in \mathbb{R}^{N_f \times (1+N_x)}$ computes raw scores for each class:

$$\text{scores}_k = \mathbf{W}_{out}\mathbf{z}_k. \tag{8}$$

These scores are passed through a softmax function to yield class probabilities:

$$P(\hat{f}_k = f_j|\mathbf{S}_k) = \frac{\exp((\text{scores}_k)_j)}{\sum_{l=0}^{N_f-1} \exp((\text{scores}_k)_l)}. \tag{9}$$

The predicted root cause is $\hat{f}_k = \arg\max_j P(\hat{f}_k = f_j|\mathbf{S}_k)$.

4.3. Training process. Training ESN4TRCA solely involves finding the optimal $\mathbf{W}_{out}$. The overall procedure is detailed in Algorithm 1. The key steps involve initializing the reservoir, driving it with training sequences to collect the final states, and then training the readout layer. For readout training, we employ ridge regression. This is preferable to ordinary least squares because when the reservoir size N_x is large the collected state matrix $\mathbf{Z}_{coll}$ can have highly correlated columns (multicollinearity), making the solution unstable. The L2 regularization in ridge regression enhances numerical stability and improves the model's generalization capability. The final computation is a direct, closed-form solution:

$$\mathbf{W}_{out}^T = (\mathbf{Z}_{coll}^T\mathbf{Z}_{coll} + \beta\mathbf{I})^{-1}\mathbf{Z}_{coll}^T\mathbf{Y}_{target}. \tag{10}$$

This closed-form solution avoids iterative optimization, making training exceptionally fast compared to BPTT-based methods.

4.4. Fault localization and inference. Once trained, ESN4TRCA performs RCA on new sequences as outlined in Algorithm 2. The process involves feeding the new sequence through the fixed reservoir, extracting the final state feature vector, and computing the class probabilities using the trained $\mathbf{W}_{out}$ to determine the most likely root cause. The output can also be a ranked list of top-k potential causes.

Algorithm 1. ESN4TRCA training algorithm.

Require: Training set $\{\mathbf{S}_k, f_k^*\}_{k=1}^{N_{train}}$, ESN hyperparameters ($N_x, \rho_{target}, s_{in}, \alpha$, sparsity, β).
Ensure: Trained readout matrix $\mathbf{W}_{out}$.
Initialize $\mathbf{W}_{in}$, $\mathbf{W}$ (randomly, then scale $\mathbf{W}$ for ρ_{target}).
Initialize empty matrices $\mathbf{Z}_{coll}$, $\mathbf{Y}_{target}$.
for each training sequence $\mathbf{S}_k = (\mathbf{u}_k(1), \ldots, \mathbf{u}_k(T_w))$ **do**
  $\mathbf{x}_k(0) \leftarrow \mathbf{0}$.
  for $t = 1$ to T_w **do**
    $\tilde{\mathbf{x}}_k(t) \leftarrow \tanh(\mathbf{W}_{in}[1; \mathbf{u}_k(t)] + \mathbf{W}\mathbf{x}_k(t-1))$.
    $\mathbf{x}_k(t) \leftarrow (1-\alpha)\mathbf{x}_k(t-1) + \alpha\tilde{\mathbf{x}}_k(t)$.
  end for
  $\mathbf{z}_k \leftarrow [1; \mathbf{x}_k(T_w)]$.
  Append $\mathbf{z}_k^T$ as a row to $\mathbf{Z}_{coll}$.
  $\mathbf{y}_{target,k} \leftarrow$ one-hot-encode(f_k^*).
  Append $\mathbf{y}_{target,k}^T$ as a row to $\mathbf{Y}_{target}$.
end for
Compute $\mathbf{W}_{out}^T \leftarrow (\mathbf{Z}_{coll}^T\mathbf{Z}_{coll} + \beta\mathbf{I})^{-1}\mathbf{Z}_{coll}^T\mathbf{Y}_{target}$.
return $\mathbf{W}_{out}$.

Algorithm 2. ESN4TRCA inference algorithm.

Require: New sequence $\mathbf{S}_{new}$, trained ESN $\{\mathbf{W}_{in}, \mathbf{W}, \mathbf{W}_{out}, \alpha\}$.
Ensure: Predicted root cause $\hat{f}_{new}$.
$\mathbf{x}_{new}(0) \leftarrow \mathbf{0}$.
for $t = 1$ to T_w **do**
  $\tilde{\mathbf{x}}_{new}(t) \leftarrow \tanh(\mathbf{W}_{in}[1; \mathbf{u}_{new}(t)] + \mathbf{W}\mathbf{x}_{new}(t-1))$.
  $\mathbf{x}_{new}(t) \leftarrow (1-\alpha)\mathbf{x}_{new}(t-1) + \alpha\tilde{\mathbf{x}}_{new}(t)$.
end for
$\mathbf{z}_{new} \leftarrow [1; \mathbf{x}_{new}(T_w)]$.
scores$_{new} \leftarrow \mathbf{W}_{out}\mathbf{z}_{new}$.
Compute probabilities $P(\hat{f}|\mathbf{S}_{new})$ via softmax (Eqn. (9)).
$\hat{f}_{new} \leftarrow \arg\max_j P(\hat{f} = f_j|\mathbf{S}_{new})$.
return $\hat{f}_{new}$.

5. Experimental setup

5.1. Datasets.

5.1.1. Simulated dataset (NS-3). We generate a comprehensive dataset using the NS-3 network simulator (Riley and Henderson, 2010).

Topology: We use a 20-node fat-tree topology, representative of data center environments, and a 14-node Abilene-like topology for WAN scenarios.

Traffic: Realistic traffic patterns (HTTP, video streaming) are generated between server-client pairs.

Monitoring: KPIs (link utilization, latency, packet loss per-flow) and node metrics (CPU/memory) are collected every 30 seconds. Alarms are generated for events like ‘Link Down’ or ‘High CPU’.

Fault injection: We inject a diverse set of $N_f - 1 = 10$ fault types, including link failures, node crashes, congestion (soft faults), and service degradation (e.g., increased processing delay on a server). Each fault has a specific temporal profile (abrupt, intermittent, gradual). The dataset contains over 20,000 labeled sequences, balanced across fault classes.

5.1.2. Public benchmark dataset. To validate our findings on real-world data, we use the dataset from the AIOps Challenge 2020 (Li and Zack, 2020). This dataset contains labeled fault sequences from a large-scale internet service, presenting real-world complexities. It includes multivariate time-series metrics from various application components and infrastructure nodes. We apply appropriate preprocessing steps, such as normalization and sequence generation, to align the data with our framework’s input requirements.

5.2. Baseline methods. We compare ESN4TRCA against a range of methods:

1. Traditional ML (on aggregated features): a support vector machine (SVM) with the RBF kernel, and random forest (RF). Features are aggregated (mean, max, std) over the time window.
2. Simple feed-forward NN: a multi-layer perceptron (MLP) on aggregated features.
3. Time series classification: k-NN with the DTW distance metric (Lee *et al.*, 2012).
4. SOTA deep learning: the LSTM and GRU networks, trained to classify the sequences.

For a comprehensive comparison, we also include a topology-aware GNN-based RCA model (Yen *et al.*, 2022) as a reference, although our method is topology-agnostic.

All ML/DL baselines are carefully tuned on the same validation set. Performance is evaluated using the following:

1. classification metrics: overall accuracy, and per-class/macro-averaged/weighted-averaged precision, recall, and F1-score. Confusion matrices are used for detailed error analysis;
2. ranking metric: top-k accuracy ($k = 3$), which is crucial for operational utility;
3. efficiency metrics: training time and average inference time per sequence.

Table 1. Overall RCA performance comparison on the simulated dataset.

Method	Accuracy (%)	Macro-F1 (%)	Weighted-F1 (%)	Train time (s)	Inference time (ms/seq)
SVM (aggregated)	82.15	79.53	81.92	158.2	2.1
Random forest (aggregated)	85.31	83.10	85.04	45.8	0.8
k-NN (DTW)	88.24	86.51	88.03	N/A (Lazy)	150.5
MLP (aggregated)	84.55	82.26	84.31	210.4	0.5
LSTM	93.52	92.81	93.45	3650.7	5.5
GRU	93.18	92.54	93.07	3120.1	5.1
GNN (topology-aware)	**94.03**	93.42	**93.98**	2890.5	4.8
ESN4TRCA (ours)	93.81	**93.20**	93.74	**15.3**	**1.2**

Note: GNN results are included as an upper-bound reference for models using explicit topology. Our method, ESN4TRCA, does not use topology information. Training times for deep models include hyperparameter tuning; core training time is reported. Also k-NN (DTW) inference time is high due to pairwise sequence alignment.

Table 2. Overall RCA performance comparison on the AIOps Challenge 2020 dataset.

Method	Accuracy (%)	Macro-F1 (%)	Weighted-F1 (%)	Train time (s)	Inference time (ms/seq)
SVM (aggregated)	80.12	78.53	80.05	158.2	2.5
Random forest (aggregated)	81.34	79.88	82.21	195.4	12.1
k-NN (DTW)	80.55	78.91	80.40	N/A (Lazy)	165.8
MLP (aggregated)	85.23	84.15	85.19	450.7	1.9
LSTM	88.51	87.92	88.35	4850.5	5.8
GRU	88.75	88.13	88.69	4912.1	5.5
GNN (topology-aware)	**89.10**	**88.46**	89.03	3105.7	5.1
ESN4TRCA (ours)	88.62	88.15	**89.58**	**18.4**	**1.6**

Note: GNN results are included as an upper-bound reference for models using explicit topology. Our method, ESN4TRCA, does not use topology information. Training times for deep models include hyperparameter tuning; core training time is reported. Also, k-NN (DTW) inference time is high due to pairwise sequence alignment.

5.3. Hyperparameter settings and implementation. ESN4TRCA hyperparameters $(N_x, \rho(\mathbf{W}), s_{in}, \alpha, \beta, T_w)$ are tuned via random search on a 20% validation split of the training data. A representative set of explored values includes $N_x \in \{200, 500, 1000, 2000\}$, $\rho(\mathbf{W}) \in \{0.9, 0.95, 0.99\}$, $\alpha \in \{0.1, 0.3, \ldots, 1.0\}$, and $\beta \in \{10^{-8}, \ldots, 10^{-2}\}$. The implementation uses Python with NumPy and Scikit-learn. DL baselines are implemented in PyTorch. Experiments are run on a server with an NVIDIA V100 GPU and Intel Xeon CPUs. Datasets are split into 60% training, 20% validation, and 20% testing sets.

6. Results and discussion

In this section, we present a comprehensive evaluation of the ESN4TRCA framework. We analyze its performance against various baseline methods on our simulated dataset, conduct an ablation study on its key hyperparameters, and assess its robustness to noise.

6.1. Overall performance comparison. The results on the NS-3 simulated dataset are summarized in Table 1. ESN4TRCA achieves a weighted-F1 score of 93.74% and a macro-F1 score of 93.20%, demonstrating highly competitive performance against SOTA deep learning models. Notably, it significantly outperforms the LSTM (92.81%) and the GRU (92.54%) in terms of macro-F1, a more challenging metric for imbalanced datasets, and is only marginally surpassed by the topology-aware GNN model.

This result is particularly remarkable given the GNN's reliance on explicit network topology information. While the GNN's slight performance edge is expected on a dataset with a static topology, the topology-agnostic nature of ESN4TRCA offers a decisive advantage in dynamic real-world environments (e.g., SDN/NFV), where obtaining an accurate, real-time topology map is often impractical. The most striking advantage, however, lies in its computational efficiency. ESN4TRCA's training time is 15–30 times faster than that of deep RNN models, and its inference is nearly instantaneous, making it ideal for real-time diagnosis.

Building on these promising results, we then evaluated the framework on the AIOps Challenge 2020 dataset to assess its performance in a more challenging scenario. As summarized in Table 2, the findings strongly corroborate our initial conclusions. While all models show a modest decrease in F1-scores, which is expected given the complexity and noise of real-world data, ESN4TRCA remains a top-performing method, achieving accuracy and F1-scores comparable to the much more complex LSTM, GRU, and GNN models. Crucially, the efficiency advantage of ESN4TRCA becomes even more pronounced on this larger dataset. The training time is over 250 times faster than that of the LSTM and GRU, a critical factor for operational environments that require frequent model retraining with new data.

Collectively, the results from both datasets demonstrate a consistent pattern: ESN4TRCA achieves a level of accuracy that is competitive with SOTA deep learning approaches but does so with a training and inference cost that is orders of magnitude lower. This combination of high accuracy and exceptional efficiency validates ESN4TRCA as a powerful and, most importantly, practical solution for real-world network RCA.

6.1.1. Per-class performance and error analysis. To delve deeper into the model's behavior, we analyze its per-class F1-scores and confusion matrix. Figure 2 shows that ESN4TRCA achieves near-perfect F1-scores (>98%) for "hard" faults with clear signatures, such as 'Link Failure' and 'Node Crash'. Performance is slightly lower but still excellent (>90%) for more subtle faults like 'Congestion' and 'Service Degradation'. This indicates the reservoir's ability to capture both abrupt and gradually evolving anomalous patterns.

The confusion matrix, as shown in Fig. 3, reveals that most misclassifications are not random but occur between causally or symptomatically related fault types. For instance, a small number of 'High CPU' instances were misclassified as 'Service Degradation', which is logical as the former is a common cause of the latter. This suggests that the model is learning meaningful semantic relationships within the fault space.

6.2. Impact of ESN hyperparameters. The performance of an ESN is governed by several key hyperparameters. We conducted an ablation study to understand their impact on the macro-F1 score, with the results shown in Fig. 4.

Figure 4(a) illustrates the effect of the reservoir size (N_x). Performance improves significantly as N_x increases from 200 to 1000, rising from 89.5% to 93.2%. This is because a larger reservoir has a higher capacity to generate rich, complex, and high-dimensional representations of the input time series. However, beyond $N_x = 1000$, the performance gain becomes marginal (93.4% at $N_x = 2000$), while the computational cost for both state collection and readout training increases quadratically. Therefore, $N_x = 1000$ offers the best trade-off between accuracy and computational cost for our problem.

Figure 4(b) shows the impact of the leaking rate α, which controls the memory of the reservoir. A low α (e.g., 0.1) means the reservoir state updates slowly, giving it a long memory but making it less responsive to recent changes. A high α (e.g., 1.0) creates a memoryless reservoir that only responds to the most recent input.

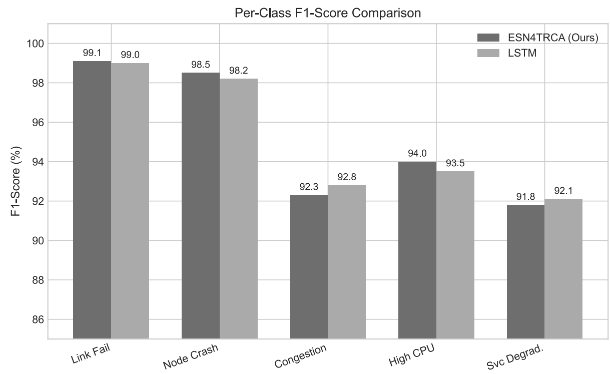


Fig. 2. Per-class F1-scores of ESN4TRCA on the simulated dataset. The plot shows the F1-score for each of the five distinct fault types evaluated (from left to right: 'Link Fail', 'Node Crash', 'Congestion', 'High CPU', 'Service Degradation'). Key takeaway: the model achieves near-perfect scores for abrupt, "hard" faults ('Link Fail', 'Node Crash') and maintains excellent performance (>90%) for more subtle, gradually evolving "soft" faults ('Congestion', 'Svc Degrad.')

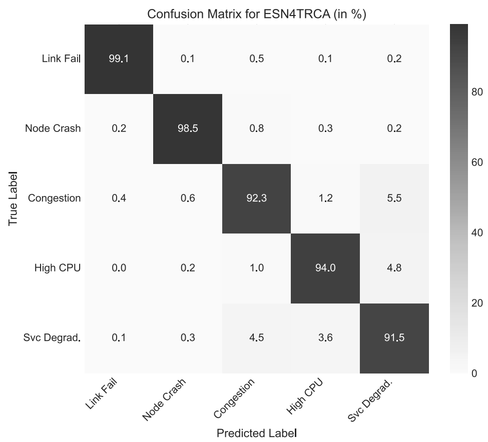


Fig. 3. Confusion matrix for ESN4TRCA on the test set, with values in percent (%). The vertical axis represents 'True Label', and the horizontal axis represents 'Predicted Label'. Key takeaway: the strong diagonal, with values >91% for all classes, indicates high classification accuracy. Off-diagonal values are sparse, and the most notable misclassifications (e.g., 'Congestion' predicted as 'Svc Degrad.') occur between logically related fault types, suggesting the model is learning meaningful semantic relationships.

Our results show a clear peak performance at $\alpha = 0.3$. This can be explained by the temporal characteristics of our data. Monitoring KPIs are collected every 30 seconds. While abrupt faults like a 'Link Fail' are instantaneous, "soft faults" such as 'Congestion' or ‘Service Degradation’ evolve over several minutes, typically spanning 10–20 time steps in our dataset. A leaking rate of $\alpha = 0.3$ creates an effective memory window of a similar timescale, allowing the reservoir to integrate information long enough to detect these gradual patterns. A much lower α would retain irrelevant past information for too long, while a higher α would be too reactive, missing the crucial build-up to the fault. This indicates that the diagnosis benefits from integrating information over a moderate time window, rather than being purely reactive or having an excessively long memory, aligning perfectly with the physical reality of network fault evolution.

In addition to the reservoir size and the leaking rate, we observed the model’s sensitivity to the spectral radius $\rho(W)$ and the input scaling factor s_{in}. Consistent with ESN theory, performance was optimal and stable when $\rho(W)$ was set slightly less than 1 (in the range of 0.9 to 0.99), which is critical for maintaining the echo state property while allowing for rich dynamics. Performance degraded significantly when $\rho(W) \geq 1$, as the reservoir became unstable. The input scaling factor, s_{in}, also required careful tuning; a value that was too small kept the reservoir in a near-linear regime, failing to capture complex non-linear fault dependencies, while a value that was too large saturated the activation functions, causing a loss of information. Moderate scaling provided the best balance for driving the reservoir effectively.

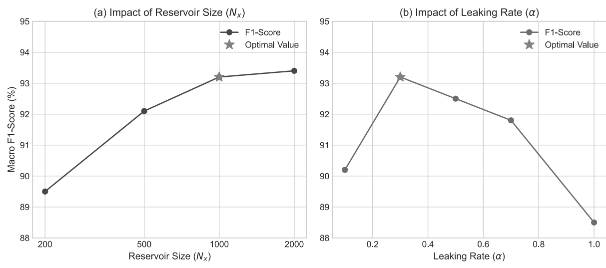


Fig. 4. Impact of the reservoir size (N_x) (a) and the leaking rate (α) (b) on the ESN4TRCA macro-F1 score. The optimal values used in our final model ($N_x = 1000, \alpha = 0.3$) are marked with a star.

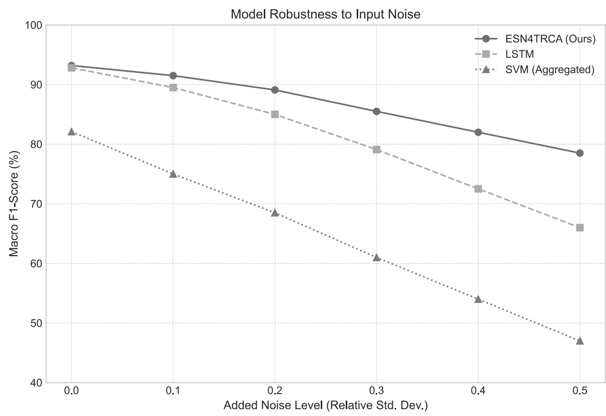


Fig. 5. F1-score degradation of ESN4TRCA and baselines under increasing input noise levels.

6.3. Robustness to noise. Real-world network monitoring data is often noisy due to measurement errors or transient, irrelevant fluctuations. We tested the robustness of ESN4TRCA by adding zero-mean Gaussian noise with increasing standard deviation (scaled relative to the standard deviation of each KPI in the training set) to the test data.

The results, depicted in Fig. 5, show that all models degrade in performance as noise increases. However, ESN4TRCA demonstrates superior robustness compared to the LSTM model. For example, at a high noise level, ESN4TRCA’s F1-score drops by approximately 8–10%, whereas the LSTM’s score drops by 15–18%. We attribute this to the inherent regularization effect of the fixed, large, random reservoir. The readout layer, being a simple linear model, is less prone to fitting the high-frequency noise that might be captured and propagated by the complex, trainable gates of an LSTM or GRU. This robustness is a highly desirable property for a production-ready RCA system.

6.4. Discussion of findings and limitations. ESN4TRCA provides a powerful and practical solution for temporal network RCA. Its primary advantage is achieving SOTA-level accuracy with a fraction of the training cost of deep RNNs, making it highly suitable for operational environments requiring frequent model updates or on-the-fly learning. The fixed, random reservoir acts as an effective temporal feature extractor, demonstrating that end-to-end training of the recurrent part is not always necessary for high performance on this task.

However, we acknowledge some limitations. First, like other supervised models, ESN4TRCA’s performance on entirely novel fault types not present in the training data is not guaranteed. Second, the reservoir itself is a “black box”, although the linearity of the readout layer provides some potential for future interpretability studies. Third, our current model does not explicitly ingest topology information, which, as shown by the GNN baseline, can provide a slight performance edge in scenarios where fault propagation is heavily dictated by a known, static topology. For highly dynamic topologies (e.g., in wireless ad-hoc networks), this limitation becomes less relevant and may even be an advantage.

6.5. Interpretability and model insights. Addressing the “black box” nature of the reservoir is a crucial direction for future work, especially in critical systems like network fault management. While the reservoir's internal dynamics are complex, the linear readout layer W_{out} provides a promising avenue for model interpretation. Since the final classification scores are a direct linear combination of the final reservoir states ($scores_k = W_{out} * z_k$), the trained W_{out} matrix can be directly inspected.

For instance, each row of W_{out} corresponds to a specific fault class. By identifying the largest coefficients in the row for 'Node Crash', we can pinpoint the specific reservoir neurons that are the strongest indicators of that fault. A subsequent analysis could then investigate which specific input KPIs and temporal patterns activate these key neurons, thereby revealing the learned fault signatures. This approach could transform the model from a pure classifier into a diagnostic tool that offers insights into why a certain root cause is suspected.

7. Conclusion and future work

In this paper, we introduced ESN4TRCA, a novel framework for root cause analysis of temporal network faults that harnesses the power and efficiency of echo state networks. By formulating RCA as a sequence classification problem, we designed an ESN architecture optimized for heterogeneous network monitoring data. The core innovation lies in leveraging a fixed, random reservoir for complex temporal feature extraction, combined with an efficiently trained linear readout layer.

Our comprehensive experimental evaluation on simulated and public benchmark data demonstrated the superiority of ESN4TRCA. It achieved classification accuracy and F1-scores competitive with or exceeding state-of-the-art deep learning models like LSTMs and GRUs, while drastically reducing training time by orders of magnitude. This remarkable balance of high accuracy and computational efficiency establishes ESNs as a powerful and practical tool for building advanced, automated fault management systems for today's complex network infrastructures.

Looking ahead, several promising avenues for future research exist:

1. Online and adaptive learning: developing methods for incrementally updating the ESN readout to adapt to evolving network conditions and new fault patterns without complete retraining.

2. Topology-aware ESNs: exploring hybrid models, such as graph echo state networks, that can integrate network topology information into the reservoir computing framework to better model fault propagation.

3. Multi-fault scenarios: extending the framework to handle situations with multiple, concurrent root causes, potentially by formulating the task as a multi-label classification problem.

Acknowledgment

This work is supported by the Major Project of the Undergraduate Education and Teaching Reform of Fujian Province (grant no. FBJY20240014) and the General Project of the Educational Science Plan of Fujian Province (grant no. FJJKBK24-148).

References

Abdelrahman, O. and Keikhosrokiani, P. (2020). Assembly line anomaly detection and root cause analysis using machine learning, *IEEE Access* **8**: 189661–189672.

Agrawal, A.P. and Singh, N. (2023). Feature optimization using a two-tier hybrid optimizer in an Internet of things network, *International Journal of Applied Mathematics and Computer Science* **33**(2): 313–326, DOI: 10.34768/amcs-2023-0023.

Agrawal, V., Panigrahi, B.K. and Subbarao, P.M.V. (2017). Intelligent decision support system for detection and root cause analysis of faults in coal mills, *IEEE Transactions on Fuzzy Systems* **25**(4): 934–944.

Ahmed, D.M., Hassan, M.M. and Mstafa, R.J. (2022). A review on deep sequential models for forecasting time series data, *Applied Computational Intelligence and Soft Computing* **2022**(1): 6596397.

Al-Mamory, S.O. and Zhang, H. (2009). Intrusion detection alarms reduction using root cause analysis and clustering, *Computer Communications* **32**(2): 419–430.

Alaeddini, A. and Dogan, I. (2011). Using Bayesian networks for root cause analysis in statistical process control, *Expert Systems with Applications* **38**(9): 11230–11243.

Arévalo, F., Tito, C., Diprasetya, M.R. and Schwung, A. (2019). Fault detection assessment using an extended FMEA and a rule-based expert system, *2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland*, Vol. 1, pp. 740–745.

Chen, B., Yu, L., Luo, W., Wu, C., Li, M., Tan, H., Huang, J. and Wan, Z. (2022). Hybrid tree model for root cause analysis of wireless network fault localization, *Web Intelligence* **20**(3): 213–223.

Chi, Y., Dong, Y., Wang, Z.J., Yu, F.R. and Leung, V.C.M. (2022). Knowledge-based fault diagnosis in industrial internet of things: A survey, *IEEE Internet of Things Journal* **9**(15): 12886–12900.

Chung, D.T., Modarres, M. and Hunt, R.M. (1989). GOTRES: An expert system for fault detection and analysis, *Reliability Engineering & System Safety* **24**(2): 113–137.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling, `https://arxiv.org/abs/1412.3555`.

De, A., Nandi, A., Mallick, A., Middya, A.I. and Roy, S. (2023). Forecasting chaotic weather variables with echo state networks and a novel swing training approach, *Knowledge-Based Systems* **269**: 110506.

de Abreu, R.S., Nunes, Y.T., Guedes, L.A. and Silva, I. (2021). A method for detecting causal relationships between industrial alarm variables using transfer entropy and K2 algorithm, *Journal of Process Control* **106**: 142–154.

De Vita, F., Nocera, G., Bruneo, D. and Das, S.K. (2023). A novel echo state network autoencoder for anomaly detection in industrial IoT systems, *IEEE Transactions on Industrial Informatics* **19**(8): 8985–8994.

Detzner, A., Rückschloß, R. and Eigner, M. (2020). Root-cause analysis with interactive decision trees, *2020 24th International Conference Information Visualisation (IV), Melbourne, Australia*, pp. 322–327.

Ferreira, D.R. and Vasilyev, E. (2015). Using logical decision trees to discover the cause of process delays from event logs, *Computers in Industry* **70**: 194–207.

Fink, P.K., Lusth, J.C. and Duran, J.W. (1985). A general expert system design for diagnostic problem solving, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**(5): 553–560.

Gonon, L. and Ortega, J.-P. (2021). Fading memory echo state networks are universal, *Neural Networks* **138**: 10–13.

Haleem Medattil Ibrahim, A., Sadanandan, S.K., Ghaoud, T., Subramaniam Rajkumar, V. and Sharma, M. (2024). Incipient fault detection in power distribution networks: Review, analysis, challenges, and future directions, *IEEE Access* **12**: 112822–112838.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Computation* **9**(8): 1735–1780.

Hu, W., Chen, T. and Shah, S.L. (2018). Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods, *IEEE Transactions on Industrial Electronics* **65**(9): 7290–7300.

Ives, G. (1991). Event root cause determination by expert system, *in* R.H. Matthews (Ed), *Reliability 91*, 1st Edn, Chapman and Hall/CRC, Boca Raton.

Jaeger, H. (2002). Adaptive nonlinear system identification with echo state networks, *in* S. Becker *et al.* (Eds), *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, pp. 609–616.

Jiang, W. and Bai, Y. (2023). APGNN: Alarm propagation graph neural network for fault detection and alarm root cause analysis, *Computer Networks* **220**: 109485.

Kiadehi, K.B., Rahmani, A.M. and Molahosseini, A.S. (2021). A fault-tolerant architecture for internet-of-things based on software-defined networks, *Telecommunication Systems* **77**: 155–169.

Kim, D.S., Shinbo, H. and Yokota, H. (2011). An alarm correlation algorithm for network management based on root cause analysis, *13th International Conference on Advanced Communication Technology (ICACT2011), Gangwon, Korea (South)*, pp. 1233–1238.

Kong, Y., Wang, Z., Nie, Y., Zhou, T., Zohren, S., Liang, Y., Sun, P. and Wen, Q. (2025). Unlocking the power of LSTM for long term time series forecasting, *Proceedings of the AAAI Conference on Artificial Intelligence* **39**(11): 11968–11976.

Lee, Y.-H., Wei, C.-P., Cheng, T.-H. and Yang, C.-T. (2012). Nearest-neighbor-based approach to time-series classification, *Decision Support Systems* **53**(1): 207–217.

Li, T., Tan, W. and Li, X. (2019). Data mining algorithm for correlation analysis of industrial alarms, *Cluster Computing* **22**: 10133–10143.

Li, Z., Tu, Y. and Ma, Z. (2022). Root cause analysis of anomalies based on graph convolutional neural network, *International Journal of Software Engineering and Knowledge Engineering* **32**(8): 1155–1177.

Li, Z. and Zack (2020). *AIOps Challenge 2020 Data*, `https://github.com/NetManAIOps/AIOps-Challenge-2020-Data`.

Lindemann, B., Müller, T., Vietz, H., Jazdi, N. and Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction, *Procedia CIRP* **99**: 650–655.

Liu, X., Chuai, G., Gao, W., Zhang, K. and Chen, X. (2019). KQIs-driven QOE anomaly detection and root cause analysis in cellular networks, *2019 IEEE Globecom Workshops, Waikoloa, USA*, pp. 1–6.

Ma, Q., Li, H. and Thorstenson, A. (2021). A big data-driven root cause analysis system: Application of machine learning in quality problem solving, *Computers & Industrial Engineering* **160**: 107580.

Niu, Z., Zhong, G., Yue, G., Wang, L.-N., Yu, H., Ling, X. and Dong, J. (2023). Recurrent attention unit: A new gated recurrent unit for long-term memory of important parts in sequential data, *Neurocomputing* **517**: 1–9.

Patel, S., Park, B., Ezzati-Jivan, N. and Fournier, Q. (2021). Automated cause analysis of latency outliers using system-level dependency graphs, *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), Hainan, China*, pp. 422–433.

Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y. and Keogh, E. (2016). Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm, *Knowledge and Information Systems* **47**: 1–26.

Racca, A. and Magri, L. (2021). Robust optimization and validation of echo state networks for learning chaotic dynamics, *Neural Networks* **142**: 252–268.

Reddy, S.S., Manikanta, Chawla, P. and Tyagi, S. (2024). *Network monitoring: A comprehensive review of current and emerging analysis tools*, ıin A. Dagur *et al.* (Eds), *Artificial Intelligence and information Technologies,* 1st Edn, CRC Press, Boca Raton.

Riley, G.F. and Henderson, T.R. (2010). The ns-3 network simulator, *in* K. Wehrle *et al.* (Eds), *Modeling and Tools for Network Simulation*, Springer, Berlin/Heidelberg, pp. 15–34, DOI: 10.1007/978-3-642-12331-3_2.

Ruijters, E. and Stoelinga, M. (2015). Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools, *Computer Science Review* **15–16**: 29–62.

Salem, F.M. (2022). Gated RNN: The gated recurrent unit (GRU) RNN, *in* F.M.Salem, *Recurrent Neural Networks,* Springer International Publishing, Cham, pp. 85–100.

Schubert, E., Sander, J., Ester, M., Kriegel, H.P. and Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN, *ACM Transactions on Database Systems* **42**(3): 1–21.

Seyed Alinezhad, H., Roohi, M.H. and Chen, T. (2022). A review of alarm root cause analysis in process industries: Common methods, recent research status and challenges, *Chemical Engineering Research and Design* **188**: 846–860.

Shah, S.Y., Dang, X.-H. and Zerfos, P. (2018). Root cause detection using dynamic dependency graphs from time series data, *2018 IEEE International Conference on Big Data (Big Data), Seattle, USA*, pp. 1998–2003.

Singh, H.V., Girdhar, A. and Dahiya, S. (2022). A literature survey based on DBSCAN algorithms, *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India*, pp. 751–758.

Singh, S., Subramania, H.S., Holland, S.W. and Davis, J.T. (2012). Decision forest for root cause analysis of intermittent faults, *IEEE Transactions on Systems, Man, and Cybernetics C: Applications and Reviews* **42**(6): 1818–1827.

Soldani, J. and Brogi, A. (2022). Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey, *ACM Computing Survery* **55**(3): 1–39.

Steenwinckel, B., De Paepe, D., Vanden Hautte, S., Heyvaert, P., Bentefrit, M., Moens, P., Dimou, A., Van Den Bossche, B., De Turck, F., Van Hoecke, S. and Ongenae, F. (2021). FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning, *Future Generation Computer Systems* **116**: 30–48.

Sun, C., Song, M., Cai, D., Zhang, B., Hong, S. and Li, H. (2024). A systematic review of echo state networks from design to application, *IEEE Transactions on Artificial Intelligence* **5**(1): 23–37.

Sun, G., ingjing Lin, Yang, C., Yin, X., Li, Z., Guo, P., Sun, J., Fan, X. and Pan, B. (2021). Stock price forecasting: An echo state network approach, *Computer Systems Science and Engineering* **36**(3): 509–520.

Tao, X.-l., Shi, L., Zhao, F., Lu, S. and Peng, Y. (2021). A hybrid alarm association method based on AP clustering and causality, *Wireless Communications and Mobile Computing* **2021**(1): 5576504.

Tong, V., Souihi, S., Tran, H.A. and Mellouk, A. (2021). Machine learning based root cause analysis for SDN network, *2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain*, pp. 1–6.

Tran, T.M., Le, X.-M.T., Nguyen, H.T. and Huynh, V.-N. (2019). A novel non-parametric method for time series classification based on k-nearest neighbors and dynamic time warping barycenter averaging, *Engineering Applications of Artificial Intelligence* **78**: 173–185.

Trierweiler Ribeiro, G., Alves Portela Santos, A., Cocco Mariani, V. and dos Santos Coelho, L. (2021). Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility, *Expert Systems with Applications* **184**: 115490.

Ullah, W., Hussain, T., Khan, Z.A., Haroon, U. and Baik, S.W. (2022). Intelligent dual stream CNN and echo state network for anomaly detection, *Knowledge-Based Systems* **253**: 109456.

Waghen, K. and Ouali, M.-S. (2019). Interpretable logic tree analysis: A data-driven fault tree methodology for causality analysis, *Expert Systems with Applications* **136**: 376–391.

Wang, D., Chen, Z., Ni, J., Tong, L., Wang, Z., Fu, Y. and Chen, H. (2023). Hierarchical graph neural networks for causal discovery and root cause localization, `https://arxiv.org/abs/2302.01987`.

Wang, H., Liu, Y., Wang, D., Luo, Y., Tong, C. and Lv, Z. (2022). Discriminative and regularized echo state network for time series classification, *Pattern Recognition* **130**: 108811.

Wang, H., Wu, Z., Jiang, H., Huang, Y., Wang, J., Kopru, S. and Xie, T. (2021a). GROOT: An event-graph-based approach for root cause analysis in industrial settings, *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia*, pp. 419–429.

Wang, M., Fu, C. and Du, X. (2021b). Decision-tree based root cause localization for anomalies in smart IoT systems, *ICC 2021: IEEE International Conference on Communications, Montreal, Canada*, pp. 1–5.

Wang, T., Qi, G. and Wu, T. (2024). KGroot: Enhancing root cause analysis through knowledge graphs and graph convolutional neural networks, `https://arxiv.org/abs/2402.13264`.

Wang, Z., Zhang, A., Li, H.C., Yin, Y., Chen, W., Lam, C.T., Mak, P.U., Vai, M.I., Gao, Y. and Pun, S.H. (2025). DFE: Deep flow embedding for robust network traffic classification, *IEEE Transactions on Network Science and Engineering* **12**(3): 1597–1612.

Wee, Y.Y., Cheah, W.P., Tan, S.C. and Wee, K. (2015). A method for root cause analysis with a Bayesian belief network and fuzzy cognitive map, *Expert Systems with Applications* **42**(1): 468–487.

Wu, Y., Dai, H.-N. and Tang, H. (2022). Graph neural networks for anomaly detection in industrial internet of things, *IEEE Internet of Things Journal* **9**(12): 9214–9231.

Xu, Z., Zhong, L. and Zhang, A. (2019). Phase space reconstruction-based conceptor network for time series prediction, *IEEE Access* **7**: 163172–163179.

Yadav, H. and Thakkar, A. (2024). NOA-LSTM: An efficient LSTM cell architecture for time series forecasting, *Expert Systems with Applications* **238**: 122333.

Yang, Z. (2024). Application of artificial intelligence in computer network technology in the age of big data, *Journal of Artificial Intelligence Practice* **7**(1): 11–16.

Yen, C.-C., Sun, W., Purmehdi, H., Park, W., Deshmukh, K.R., Thakrar, N., Nassef, O. and Jacobs, A. (2022). Graph neural network based root cause analysis using multivariate time-series KPIs for wireless networks, *NOMS 2022: 2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary*, pp. 1–7.

Zaheer, S., Anjum, N., Hussain, S., Algani, A.D., Iqbal, J., Bourouis, S. and Ullah, S.S. (2023). A multi parameter forecasting for stock time series data using LSTM and deep learning model, *Mathematics* **11**(3): 590.

Zhang, A., Zhu, W. and Li, J. (2019). Spiking echo state convolutional neural network for robust time series classification, *IEEE Access* **7**: 4927–4935.

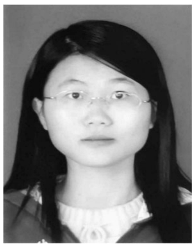
Zhou, D., Zhao, Y., Wang, Z., He, X. and Gao, M. (2020a). Review on diagnosis techniques for intermittent faults in dynamic systems, *IEEE Transactions on Industrial Electronics* **67**(3): 2337–2347.

Zhou, J., Wang, H., Xiao, F., Yan, X. and Sun, L. (2020b). Network traffic prediction method based on echo state network with adaptive reservoir, *Software: Practice and Experience* **51**(11): 2238–2251.

Zhou, J., Han, T., Xiao, F., Gui, G., Adebisi, B., Gacanin, H. and Sari, H. (2022). Multiscale network traffic prediction method based on deep echo-state network for internet of things, *IEEE Internet of Things Journal* **9**(21): 21862–21874.

Zhu, Q. (2008). Goal trees and fault trees for root cause analysis, *2008 IEEE International Conference on Software Maintenance, Beijing, China*, pp. 436–439.

Bixian Zhang received her MS degree from Fuzhou University in 2011. She is now an associated professor in the School of Electronic Engineering, Fuzhou Institute of Technology. Her research interests include communication network engineering and network function virtualization (NFV).



Yixia Chen, born in 1987, received her Master's degree from the Nanjing University of Science and Technology in 2013. She is currently a full-time teacher at the School of Electronic Engineering, Fuzhou University of Technology. Her research interests include automated testing of network equipment, and the design and optimization of novel network architectures.

Received: 26 June 2025
Revised: 21 August 2025
Accepted: 22 October 2025