IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 November 2023, accepted 23 November 2023, date of publication 27 November 2023, date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337029

RESEARCH ARTICLE

Root Cause Analysis of Communication Network Based on Deep Fuzzy Neural Network

BIXIAN ZHANG
College of Computing and Information Science, Fuzhou Institute of Technology, Fuzhou 350506, China
e-mail: bxzhang_fit@163.com

This work was supported in part by the Collaborative Education Project between the Department of Higher Education of the Ministry of Education and Industry under Grant 202102215005 and Grant 202102082014, in part by the Fujian Provincial Education and Research Project for Young and Middle-Aged Teachers under Grant JAT200906, and in part by the Fuzhou Institute of Technology Scientific Research Project under Grant FTKY20230002.

ABSTRACT In the realm of communication networks, root cause analysis plays a vital role in maintaining efficient and reliable operation. However, existing root cause analysis methods face limitations and drawbacks, including their inability to handle complex data and disturbances, as well as inaccuracies in identifying root causes. To this end, the paper presents the Deep Fuzzy Neural Network approach as an innovative solution. Integrating the strengths of deep learning and fuzzy logic inference, where the deep learning technique utilizes the parallel computing fusion of convolutional neural network and long short-term memory to extract the spatial-temporal features from sophisticated fault data of communication network. By leveraging this parallel computing fusion module, the proposed framework effectively addresses the flaws of traditional root cause analysis methods. Furthermore, the incorporation of fuzzy logic enables the proposed model to manage disturbances such as uncertainty and noise inherent in the data, ensuring robust performance. Experimental results also demonstrate our proposed deep fuzzy neural network approach is an effective method for network root cause analysis in overcoming limitations inherent in existing methods and providing superior accuracy and resilience.

INDEX TERMS Root cause analysis, communication networks, deep fuzzy neural network, fuzzy set theory.

I. INTRODUCTION

Modern communication networks are essential for facilitating seamless connectivity and efficient data transfer across various platforms and devices. However, network failures and disruptions can have severe implications, causing degraded performance, service interruptions, and financial losses. It is crucial for network operators to accurately and efficiently identify the root cause behind these failures to minimize downtime and ensure optimal network performance. Root cause analysis (RCA), a task to to identify the underlying factors or events that trigger a network failure and determine the appropriate remedial actions, has been considered as a critical process in troubleshooting communication network failures [1]

Conventional RCA methods typically rely on manual analysis or rule-based algorithms, which are expensive, error-prone, and often struggle to capture the complexity and dynamic nature of modern communication networks. Therefore, researchers have proposed various automated methods to assist in RCA, including statistical methods [2], [3], machine learning [4], [5], [6], [7], and deep learning [8], [9]. Among them, deep learning has been proven to be a powerful tool in a variety of applications, including image classification [10], natural language processing [11], and speech recognition [12]. However, deep learning techniques are not designed explicitly to handle the uncertainty and ambiguity that frequently arises in network analysis. Fuzzy set theory provides a framework for tackling such scenarios by incorporating fuzzy logic and set theory to represent vague and ambiguous concepts with linguistic variables [13]. By combining the strengths of deep learning and fuzzy set theory, deep fuzzy neural networks (DFNN) have

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

been demonstrated to be effective alternatives for modeling complex systems and making accurate predictions [14], [15].

Motivated by the limitations of current approaches, this paper proposed a DFNN approach for network RCA, which leverages the recent advancements in deep learning and fuzzy logic to overcome the shortcomings of conventional RCA methods. The deep learning architecture adopts parallel convolutional neural network (CNN) and long short-term memory (LSTM) modules, where the CNN is used for extracting spatial features from network data, and the LSTM is used for the extraction of temporal features. This design enables the model to learn complex relationships within the network data, thereby making the more accurate diagnosis of failures. Further, the fuzzy logic provides the ability to handle uncertainties and imprecisions in data by allowing the representation of vague or approximate knowledge. By incorporating fuzzy logic, the proposed DFNN model can capture the inherent uncertainties in network data and provide more robust and reliable fault diagnoses.

In this paper, we make the following main contributions:

1) We proposed a DFNN framework for RCA of communication networks, where the DFNN combine the deep learning technique and fuzzy logic inference. The deep learning can extract core features related to the root cause from complex fault data, while fuzzy logic enables the model to handle disturbances such as uncertainty and noise in fault data more effectively. This combination method allows us to achieve higher root cause identification accuracy and stronger robustness in RCA issues.

2) The deep learning architecture utilizes the parallel computing fusion of convolutional neural network (CNN) and long short-term memory (LSTM) [16], where the CNN is responsible for extracting spatial features from the fault data, while the LSTM is employed to extract temporal features. This design enhances the model's ability to decipher complex relationships within network data, consequently leading to more accurate failure diagnosis.

3) Through extensive experiments and comparisons with traditional methods, we demonstrate the superiority of our proposed approach in terms of accuracy and efficiency. Our model successfully identifies root causes of network failures promptly and accurately, surpassing the performance of manual analysis and rule-based algorithms. The results showcase the potential of our DFNN in addressing the challenges of RCA in communication networks.

To the best of our knowledge, this is the first work that combines deep learning and fuzzy logic for RCA problems. Our framework offers a more transparent and accurate analysis of network failures, enabling network operators to take targeted troubleshooting actions for improved network reliability.

The remainder of this paper is structured as follows. Section II provides a thorough literature review of related works in root cause analysis for communication networks. Section III presents the methodology and architecture of our proposed DFNN model, explaining the integration of deep learning and fuzzy logic. In Section IV, we describe the dataset and experimental setup for model training and evaluation. Section V presents the results and discussion, highlighting the performance of our proposed approach compared to existing methods. Finally, Section 6 concludes the paper and discusses potential future directions in the field of root cause analysis in communication networks.

II. RELATED WORKS

In general, RCA strategies are widely categorized into model-based, rule-based, case-based, and machine learning methodologies. Model-based strategies hinge on expert knowledge to codify the system's conduct into a mathematical model. It mandates a deep comprehension of the system's inherent structure and operational mechanism [17]. For instance, [18] introduced a simple network management protocol (SNMP) grounded on a management model. The model is adept in pinpointing the root cause of an event and providing problem-solving guidelines to operators. However, the challenges lie in obtaining and maintaining up-to-date models.

Cronk et al. [19] introduced a rule-based strategy for managing and operating communication networks. Such an approach typically encapsulates three components: a rule base, a rule discovery engine, and an inference engine. The initial two sections can be achieved via iterative and incrementing algorithms, where new rules are continuously incorporated into the rule base through iteration algorithms under diverse conditions. The inference engine subsequently determines the most applicable rule in a given scenario [20]. However, this approach poses significant complexity in updating and expanding the knowledge base and executing the inference process. This becomes particularly challenging in a network with constantly evolving topology, hence requiring frequent rule updates. Despite recent proposals for techniques that can automatically discern rules founded on observed symptoms [21], [22], this method has limited suitability for such networks.

Case-based strategies, meanwhile, leverage previous human experiences to solve fault cases [23], [24], [25]. Once a problem is solved, the experience is stored in the case base and future similar issues rely on it for solutions. Cai et al. [26] proposed a Bayesian networks-based method for fault diagnosis, Alaeddini and Dogan [2] used Bayesian networks for RCA in statistical process control. Bauer et al. [27] applied KNN to diagnose faults in optical-fiber communication networks. However, these methods are susceptible to constrain case-based strategies, like the time-consuming update of numerous cases, case matching, and case base enhancement.

Srinivasan et al. [28] applied machine learning techniques to pinpoint and localize faults in communication networks, taking into consideration the packet loss, end-to-end delay, and aggregate flow rate in normal and fault situations. While powerful in fault localization for complex communication networks [29], [30], [31], these techniques necessitate lengthy training periods and extensive data from fault situations, which may not always be feasible in high-reliability and high-security networks. Considering the powerful computational ability of deep learning, Jiang [32] utilized deep neural networks (DNNs) to predict future network traffic, and their results demonstrated a high accuracy and scalability compared to traditional methods of traffic forecasting. Lai et al. [33] used DNN to predict network traffic for digital twin networks. Another important area where deep learning has been applied is in anomaly detection for identifying network faults. Navya et al. [34] proposed a deep learning-based method for detecting network anomalies and achieved a high precision in their results. Furthermore, Naseer et al. [35] utilized a deep residual learning network for intrusion detection in wireless sensor networks, achieving higher accuracy over baseline methods.

Despite the success of deep learning in fault diagnosis and RCA for communication networks, DNN models are typically complex and can suffer from over-fitting due to the large number of parameters required for the network. Additionally, DNN models are not easily interpretable, making it challenging to understand the logic behind the root cause analysis results.

To address these challenges, fuzzy logic-based approaches have been integrated with deep learning techniques to develop deep fuzzy neural networks (DFNNs). These networks leverage the advantages of fuzzy logic-based inference and deep learning for more comprehensive and accurate RCA. For example, Zhang et al. [36] applied DFNNs for machinery fault diagnosis. Wang and Qiao [37] proposed a self-organizing fuzzy neural network for nonlinear system modeling.

Despite their potential, DFNNs remain relatively unexplored in the field of communication network fault diagnosis and RCA. Existing research in this area mainly focuses on case studies and laboratory experimentation. More research is needed to investigate the effectiveness and scalability of DFNNs in real-world communication network settings.

In summary, traditional methods of RCA, machine learning algorithms, and deep learning techniques have been proposed for fault diagnosis and RCA in communication networks. Recent research has shown that integrating fuzzy logic with deep learning can lead to more accurate and comprehensive RCA, as demonstrated by the effectiveness of DFNNs. However, more research is needed in this area to explore the applicability of these techniques in real-world communication networks.

III. PROPOSED METHOD

In this section, we present our proposed Deep Fuzzy Neural Network (DFNN) model for performing root cause analysis of communication networks. The DFNN model combines the capabilities of Convolutional Neural Networks (CNNs) and Long-Short Term Memory (LSTM) models with fuzzy logic inference to achieve accurate and transparent diagnosis of network failures.

A. MODEL ARCHITECTURE

As shown in Figure 1, the DFNN model is composed of three main components: CNN, LSTM and the fuzzy decision module.

CNN: CNNs are widely known for their ability to learn spatial dependencies in data. In the context of root cause analysis, CNNs are employed to extract meaningful features from the network data. These features capture the spatial relationships between different network components and provide valuable insights into the root causes of failures. Let X be the reshaped input data with the size of 16×16, and C be the set of convolutional layers in the CNN. Each layer c in C applies a set of filters W_c to the input X and generates feature maps H_c. The convolution operation can be represented as:

$$H_c = \delta(W_c * X) \tag{1}$$

where $*$ denotes the convolution operation and δ is the activation function. In the CNN module of Figure 1, "Conv 32@3 × 3 Stride2" means the convolutional layer is set by 32 convolutional kernels with size of 3 × 3 and stride of 2, and the "Deconv" means the deconvolutional layers. All the Conv, Deconv and FC layers use the ReLU activation function.

LSTM: LSTM models are a type of recurrent neural network (RNN) that excel at capturing temporal dependencies. By incorporating LSTM into our DFNN model, we enable the analysis of sequences of network events leading up to failures. This temporal information helps in understanding the dynamics and patterns that contribute to network failures.

Let Y be the input sequence of network events, and LSTM be the set of LSTM layers in the model. Each layer l in LSTM consists of a memory cell C_l, input gate I_l, and output gate O_l. The LSTM model updates the memory cell and generates the hidden state H_l according to the following equations:

$$I_l = \sigma(W_i * Y + U_i * H_{l-1} + b_i) \tag{2}$$
$$F_l = \sigma(W_f * Y + U_f * H_{l-1} + b_f) \tag{3}$$
$$O_l = \sigma(W_o * Y + U_o * H_{l-1} + b_o) \tag{4}$$
$$C_l = F_l \otimes C_{l-1} + I_l \otimes \tanh(W_c * Y + U_c * H_{l-1} + b_c) \tag{5}$$
$$H_l = O_l \otimes \tanh(C_l) \tag{6}$$

where $W_i, W_f, W_o, W_c, U_i, U_f, U_o, U_c, b_i, b_f, b_o, b_c$ are the weight matrices and bias terms of the LSTM model, and $\otimes$ denotes element-wise multiplication.

Fuzzy Decision Module: As shown in Figure 2, this module consists of five layers, i.e., an embedding input layer, a membership function layer, a fuzzy rule layer, two fully-connected layers and an output layer, which will be introduced below.

B. FUZZY LOGIC INFERENCE

To enhance the interpretability of the model outputs and facilitate the understanding of network operators, we integrate

TABLE 1. Related work in the field of RCA.

Ref.	Year	Method	Advantages	Limitation
Diaz et al. [18] Steinert et al. [17]	2000 2010	Model-Based Methods	Rich in expert knowledge; Can pinpoint root cause of an event	Requires deep understanding of system; Difficulty in maintaining up-to-date models
Diaz et al. [18] Steinert et al. [17]	2000 2010	Rule-Based Methods	Continuously incorporates new rules; Determines the most applicable rule in a scenario	Complexity in updating and expanding knowledge base; Limited suitability for constantly evolving networks
Bauer et al. [27]	2007	K-Neighbor Nearest	Computational simplicity; Disturbance tolerance.	Unable to handle complex nonlinear relationships in data features.
Alaeddini et al. [2] Cai et al. [26]	2011 2017	Bayesian Networks	Great at determining causal relationships; providing the understanding of structurally encoded independences.	Assuming variable independence might not hold true.
Cheng et al. [29] Zhang et al. [36]	2016 2019	Complex Networks	Localizes faults in communication networks considering various attributes	Requires lengthy training periods; Needs extensive fault data not available in high-security networks
Naseer et al. [35] Jiang et al. [32] Navya et al. [34]	2018 2021 2021	Deep Neural Networks	Highly accurate; Provides scalability over traditional methods	Can suffer from overfitting due to high parameters; Lacks interpretability

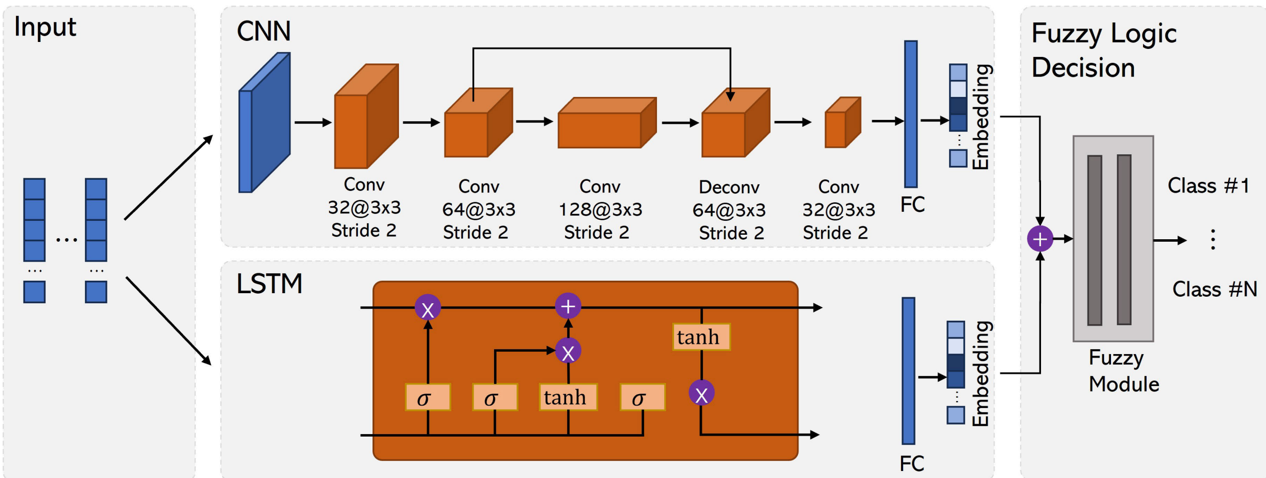

FIGURE 1. The framework of our proposed DFNN model.

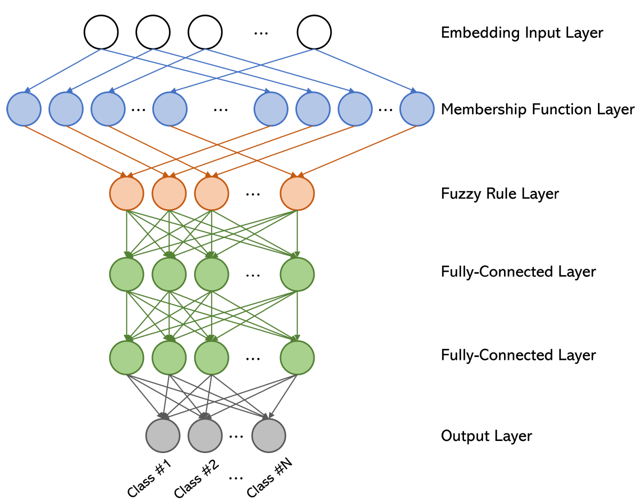

FIGURE 2. The structure of fuzzy module of the overall framework.

fuzzy logic inference into the DFNN model. Fuzzy logic allows for linguistic interpretations of the model's diagnosis, making it easier to comprehend and act upon by network operators.

Let Z be the output from the LSTM model, and L be the set of linguistic labels in the fuzzy logic inference system. Each label l in L is associated with a membership function $\mu_l(Z)$, which represents the degree of membership of Z to the linguistic label. The fuzzy output F is calculated by aggregating the membership functions as:

$$F = \sum (w_l * \mu_l(Z)) \tag{7}$$

where w_l represents the weights assigned to each linguistic label, ensuring the proper combination of the fuzzy outputs.

C. MODEL OPERATION

The DFNN model operates as follows:

Preprocessing: Before feeding the data into the model, preprocessing steps are applied to ensure its suitability. This may include data normalization, feature scaling, and handling missing or categorical data.

Feature Extraction: The network data is fed into the CNN component of the DFNN model, which learns and extracts relevant features from the spatial relationships among network components.

Temporal Analysis: The features extracted by the CNN are then passed as inputs to the LSTM model, which analyzes the temporal dependencies and sequential patterns within the network data.

Fuzzy Logic Decision: The outputs from the LSTM model are passed through a fuzzy logic inference system, which provides linguistic interpretations of the model's diagnosis. This fuzzy output enhances transparency and comprehensibility for network operators.

Assume the embedding input vector $u \in \mathbb{R}^I$, for the fuzzy function layer, each node calculates the membership degree function f_{ij}, $(i = 1, 2, \ldots, I; j = 1, 2, m_i)$ of each component of the input vector u_i belonging to the respective variable fuzzy set, where m_i represents the number of u_i divisions for the fuzzy rules. Let there be a total of I sets of membership degree functions, with each set containing m_i membership degree functions. Therefore, the membership degree function can be represented as:

$$f_{ij} = e^{-\frac{(u_i - c_{ij})^2}{\phi_{ij}^2}} \tag{8}$$

where c_{ij} and ϕ_{ij} denote the center and width of the membership function, respectively. The node number of the membership function layer is $N_2 = \sum_{i=1}^{n} m_i$.

The fuzzy rule layer is responsible for matching the conditions of fuzzy rules and calculating the utility of each rule. The membership function layer consists of I groups of membership functions, which are combined by selecting one membership function from each group without repetition to form the nodes of this layer, i.e.,

$$a_j = f_1^{i1} f_2^{i2} \cdots f_I^{iI} \tag{9}$$

or

$$a_j = \min\left\{f_1^{i_1}, f_2^{i_2}, \cdots, f_I^{i_I}\right\} \tag{10}$$

where $i_1 \in \{1, 2, \ldots, m_1\}, i_2 \in \{1, 2, \ldots, m_2\}, \ldots$. The total number of nodes in this layer is $N_3 = \prod_{i=1}^{I} m_i = m$. For a given input, only the variable values near the input point have higher membership degree values, while the membership degrees of variable values far from the input point are small or close to 0. When the membership degree function is very small, the approximate value is considered as 0.

D. TRAINING METHOD

Some parameters of the fuzzy neural network model need to be predefined, such as the number of fuzzy layer for each input component u_i, the selection of the membership function form (this paper chooses the normal distribution), and so on. The parameters that need to be trained include the weights of the fully connected layer and the centers and widths of the membership functions in the membership function layer.

Without loss of generality, Figure 3 represents the q-th layer, j-th node of the fuzzy neural network, where the node's

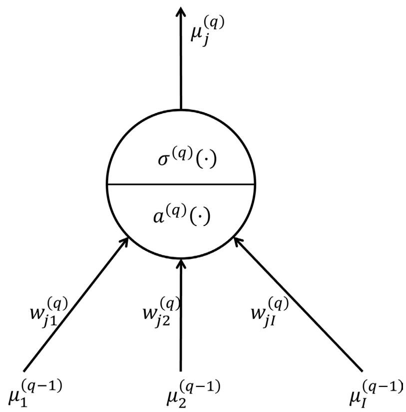

FIGURE 3. The model of fuzzy neuron.

input is

$$\sigma^q\left(u_1^{(q-1)}, u_2^{(q-1)}, \ldots, u_I^{(q-1)}; w_1^{(q)}, w_2^{(q)}, \ldots, w_I^{(q)}\right) \tag{11}$$

and the output of the node is

$$o_j^{(q)} = g^{(q)}\left(a^{(q)}\right) \tag{12}$$

For general neuron nodes, there is typically

$$a^q = \sum_{i=1}^{I} w_{ji}^{(q)} u_i^{(q-1)} \tag{13}$$

$$u_j^{(q)} = g^{(q)}\left(a^{(q)}\right) = \frac{1}{1 + e^{-f\left(a^{(q)}\right)}} \tag{14}$$

We use the error backpropagation algorithm to train the weight parameters in the network and the centers and widths of the membership functions in the membership function layer. The error cost function is designed as:

$$E = \frac{1}{2}\sum_{i=1}^{r}(y_i - \widehat{y}_i)^2 \tag{15}$$

where y_i and $\widehat{y}_i$ are the network output and desired output with respect to the i-th input.

We have

$$\delta_i^o = -\frac{\partial E}{\partial a_i^o} = -\frac{\partial E}{\partial y_i} = y_i - \widehat{y}_i \tag{16}$$

Thus,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_i^o}\frac{\partial a_i^o}{\partial w_{ij}} = -\delta_i^o u_j^{o-1} = -\left(y_i - \widehat{y}_i\right)\overline{a}_j \tag{17}$$

Continuing with the calculation backward, there is

$$\begin{aligned} \delta_j^{o-1} &= -\frac{\partial E}{\partial a_j^{o-1}} = -\sum_{i=1}^{r}\frac{\partial E}{\partial a_j^o}\frac{\partial a_i^o}{\partial \sigma_j^{o-1}}\frac{\partial \sigma_j^{o-1}}{\partial a_j^{o-1}} \\ &= \sum_{i=1}^{r}\delta_i^o w_{ij} \end{aligned} \tag{18}$$

$$\delta_j^{o-2} = -\frac{\partial E}{\partial a_j^{o-2}} = \frac{\partial E}{\partial a_j^{o-1}}\frac{\partial a_i^{o-1}}{\partial \sigma_j^{o-1}}\frac{\partial \sigma_j^{o-1}}{\partial a_j^{o-2}}$$

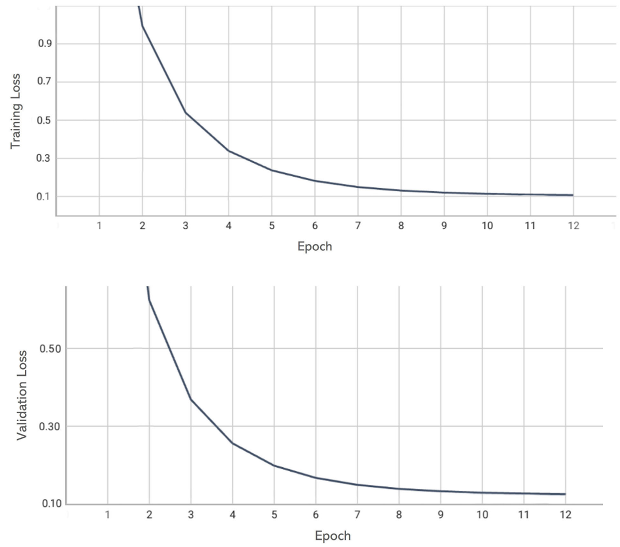

FIGURE 4. The training loss (top) and validation loss (bottom) of our DFNN.

$$= \delta_j^{o-1} \sum_{i=1, i\neq j}^{m} u_i^{o-2} / \left(\sum_{i=1}^{m} \left(\mu_i^{o-2} \right) \right)^2$$

$$= \delta_j^{o-1} \sum_{i=1, i\neq j}^{m} a_i / \left(\sum_{i=1}^{m} a_i \right)^2 \tag{19}$$

$$\delta_j^{o-1} = -\frac{\partial E}{\partial a_j^{o-3}} = -\sum_{i=1}^{r} \frac{\partial E}{\partial a_j^{o-2}} \frac{\partial a_i^{o-2}}{\partial \sigma_j^{o-3}} \frac{\partial \sigma_j^{o-3}}{\partial a_j^{o-3}}$$

$$= \sum_{i=1}^{r} \delta_i^{o-2} s_{ij} e^{\sigma_j^{o-3}}$$

$$= \sum_{i=1}^{r} \delta_i^{o-2} s_{ij} e^{-\frac{(\mu_i - c_{ij})^2}{\phi_{ij}^2}} \tag{20}$$

In this paper, a^{o-2} is selected as the min-max function, thus, if $\sigma_{ij}^{o-3} = \mu_i$ is the minimal value of the k-th rule, we have

$$s_{ij} = \frac{\partial a_k^{o-2}}{\partial \sigma_{ij}^{o-3}} = \frac{\partial a_k^{o-2}}{\partial \mu_j} = 1 \tag{21}$$

else

$$s_{ij} = \frac{\partial a_k^{o-2}}{\partial \sigma_{ij}^{o-3}} = \frac{\partial a_k^{o-2}}{\partial \mu_j} = 0 \tag{22}$$

Thus, we have the first-order gradient,

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial E}{\partial a_{ij}^{o-3}} \frac{\partial a_{ij}^{o-3}}{\partial c_{ij}} = -\delta_{ij}^{o-3} \frac{2\left(\mu_i - c_{ij}\right)}{\phi_{ij}^2} \tag{23}$$

$$\frac{\partial E}{\partial \phi_{ij}} = \frac{\partial E}{\partial a_{ij}^{o-3}} \frac{\partial a_{ij}^{o-3}}{\partial \phi_{ij}} = -\delta_{ij}^{o-3} \frac{\left(\mu_i - c_{ij}\right)^2}{\phi_{ij}^2} \tag{24}$$

The learning method for all the trainable parameters can be concluded as

$$w_{ij}(k+1) = w_{ij}(k) + \eta \frac{\partial E}{\partial w_{ij}},$$

$$c_{ij}(k+1) = c_{ij}(k) + \gamma \frac{\partial E}{\partial c_{ij}},$$

$$w_{ij}(k+1) = \phi_{ij}(k) + \xi \frac{\partial E}{\partial \phi_{ij}}, \quad (i=1,2,\ldots,r; j=1,2,\ldots,m)$$

where $\eta > 0, \gamma > 0, \xi > 0$ are the learning rate of weights, function center and width, respectively. k is the iterative step.

When the actual output of the network matches the desired output, it indicates the end of training. Otherwise, through error backpropagation, the parameters of each layer are adjusted until the error is reduced within the desired range.

IV. EXPERIMENTAL RESULTS

In this section, we present the detailed experimental results obtained from applying the proposed deep fuzzy neural network (DFNN) method for root cause analysis of communication networks. We evaluate the performance of our approach and compare it with other existing methods to demonstrate its effectiveness in identifying root causes accurately and efficiently.

A. DATASET

We use a real-world dataset collected from a large-scale communication network comprising routers, switches, and other network devices. The dataset includes performance metrics, logs, and events recorded over a period of several months. The dataset contains 6,000 communication network events that were collected from a real-world enterprise network. Each event is described by 10 attributes

1) Timestamp: the time when the event occurred, recorded with a granularity of one second. 2) Source IP: the IP address of the source host that generated the event. 3) Destination IP: the IP address of the destination host that received the event. 4) Source port: the port number of the source host. 5) Destination port: the port number of the destination host. 6) Protocol type: the protocol used for the event, represented as a numerical code (e.g., TCP=1, UDP=2). 7) Bytes transmitted: the number of bytes transmitted between the source and destination hosts during the event. 8) Packet rate: the rate at which packets were transmitted during the event, measured in packets per second. 9) Error rate: the rate of transmission errors during the event, measured as a percentage of total packets transmitted. 10) Event label: ground truth root causes for network failures or performance degradation, which are determined through manual analysis by network experts.

The dataset was preprocessed by normalizing the numerical attributes to have zero mean and unit variance, and converting the categorical attribute (protocol type) into a one-hot encoding representation. The 6,000 events were randomly split into a training set of 4,800 events and a testing set of 1,200 events, using a 80:20 ratio. The dataset was also balanced, meaning that the same number of benign and malicious events were included.

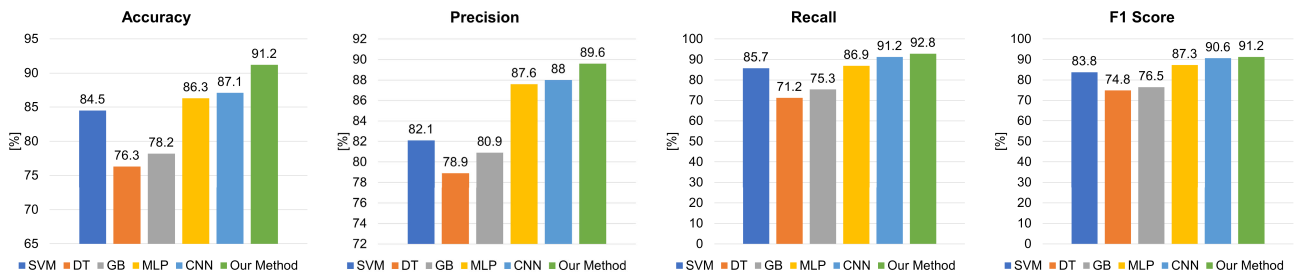

FIGURE 5. Performance comparison of root cause analysis methods.

TABLE 2. Detailed performance metrics of DFNN.

Class	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Network Link Failure	92.5	91.6	93.8	92.7
Hardware Malfunction	88.3	87.9	88.7	88.3
Congestion	94.1	93.8	94.4	94.1
Software Bug	87.2	87.4	87.0	87.2
Configuration Error	91.8	91.3	92.4	91.8
External Factors	89.6	90.1	89.1	89.6

TABLE 3. Case study of root cause analysis using DFNN.

Network Failure	Predicted Root Cause	Ground Truth	Membership Degree
Network link failure	Hardware malfunction on router X	Link failure	0.92
	Congestion due to high traffic on switch Y		0.79
	Software bug in switch Z		0.68

B. IMPLEMENTATION PLATFORM

The software framework employed for developing the DFNN model consisted of Python 3.7, taking advantage of its extensive libraries for data analysis. Pytorch were used for the architecture build-up and the orchestration of the deep learning components of the DFNN model, providing a flexible and user-friendly interface. Scikit-learn, on the other hand, was utilized for data preprocessing and performance evaluation of the model.

The fuzzy set manipulation necessary for handling uncertain network data was realized using the Scikit-Fuzzy library, a Python based toolkit for fuzzy logic and fuzzy systems development. Scikit-Fuzzy facilitated the handling of ambiguity innate in network data and helped ensure the robustness of the model under varying data uncertainties.

The implementation was performed on a PC with a CPU Intel Core i7, RAM 24GB and a graphics processing unit (GPU) Nvidia RTX 3090, which allowed for the efficient training of the deep learning component of the DFNN model and facilitated quick experimentation and prototyping. The GPU-accelerated implementation allowed the handling of large-scale network data and enabled a fast evaluation of the model's capabilities.

C. EVALUATION

To evaluate the performance of our proposed RCA method, we use real-world datasets collected from a communication network, and the loss curves of DFNN training and validation are given in Figure 4. We compare our method with several state-of-the-art RCA methods based on neural networks, fuzzy logic, and clustering techniques. We use measures such as Precision, Recall, F1-score, and Accuracy to assess the performance of our method. Let TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively, where.

True Positive (TP): The DFNN correctly identifies an event as the root cause for network failures or performance degradation. This would correspond to correctly classifying a 'malicious' event in the dataset, for example.

True Negative (TN): The DFNN correctly identifies the event as not a root cause for any issues. This would correspond to correctly classifying a 'benign' event in the dataset, for example.

False Positive (FP): The DFNN incorrectly identifies a benign event as the root cause for network failures or performance degradation.

False Negative (FN): The DFNN fails to identify a malicious event as the root cause despite it being one.

Then we have:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{25}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{26}$$

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{27}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{28}$$

Our experimental results show that our proposed method outperforms existing methods in terms of accuracy and efficiency.

D. PERFORMANCE COMPARISON

We trained several models, including a traditional decision tree (DT) model, multi-layer perceptron (ANN) model, support vector machine (SVM), gradient boosting (GB), convolutional neural network (CNN), and our proposed DFNN model. We used the Gini impurity index as the splitting criterion for the DT model, two hidden layers with 64 neurons each for the ANN and DFNN models. We also used the Adam optimizer with a learning rate of 0.001 and a batch size of 32 for both ANN and DFNN models.

Figure 5 shows the performance comparison between the DFNN with other five commonly used methods. The DFNN achieves the highest accuracy, precision, recall, and F1 score, indicating its superior ability to identify root causes accurately.

Table 2 provides detailed performance metrics of the DFNN method for each specific root cause class. The metrics include accuracy, precision, recall, and F1 score. The DFNN achieves high accuracy and balanced precision and recall across different root cause classes, indicating its ability to handle various types of network issues effectively. The network link failure class shows the highest accuracy, precision, recall, and F1 score, indicating the DFNN's strong performance in identifying this type of root cause.

E. CASE STUDY AND INTERPRETATION

To further illustrate the effectiveness of the DFNN method, we provide a case study where a network failure was analyzed using the DFNN. The predicted root causes are compared with the ground truth, and the fuzzy membership degrees provide insights into the confidence of the predictions.

Table 3 shows the root cause analysis results for a network failure case study. The DFNN successfully predicts the primary root cause as a hardware malfunction on router X with a high membership degree of 0.92, indicating a high level of confidence in the prediction. The DFNN also suggests secondary causes such as congestion and software bugs with lower membership degrees, providing additional insights into potential contributing factors.

Overall, the experimental results demonstrate that the proposed DFNN method outperforms traditional methods such as SVM and DT in root cause analysis of communication networks. The detailed performance metrics across different root cause classes highlight the DFNN's effectiveness and its ability to handle various types of network issues with high accuracy, precision, recall, and F1 score.

V. CONCLUSION

In this paper, we proposed a deep fuzzy neural network (DFNN) framework for root cause analysis (RCA) of communication networks. The DFNN synergistically merges deep learning techniques and fuzzy logic inferences, to extract core features related to the root cause from intricate fault data while effectively handling disturbances like uncertainty and noise. This novel fusion of methodologies has showcased higher accuracy and robustness in RCA. The integration of convolutional neural networks, which extracts spatial features, and long short-term memory (LSTM), which deals with temporal features, has amplified the model's capacity to decipher complex relationships within network data, consequently increasing the accuracy of our failure diagnosis.

Extensive experiments, alongside comparisons with traditional methods, validated the superiority of our proposed approach both in terms of precise failure causality identification and efficiency. Our model outshined manual analysis and rule-based algorithms, demonstrating not only its ability to accurately identify the root causes of network failures but also highlighting its significant potential in addressing intricate RCA challenges in communication networks.

It should be noted that our DFNN method applied a manual design for feature selection, it may be not a reliable and accurate approach, future work can focus on end-to-end framework with automatic feature extraction. On the other hand, more effective deep learning mechanisms, such as attention, transformer, can be applied for a computational stronger model.

REFERENCES

[1] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, "Survey on models and techniques for root-cause analysis," 2017, *arXiv:1701.08546*.

[2] A. Alaeddini and I. Dogan, "Using Bayesian networks for root cause analysis in statistical process control," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11230–11243, Sep. 2011.

[3] G. Li, S. J. Qin, and T. Yuan, "Data-driven root cause diagnosis of faults in process industries," *Chemometric Intell. Lab. Syst.*, vol. 159, pp. 1–11, Dec. 2016.

[4] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, and B. Vogel-Heuser, "Combining knowledge modeling and machine learning for alarm root cause analysis," *IFAC Proc. Volumes*, vol. 46, no. 9, pp. 1843–1848, 2013.

[5] H. Joe Steinhauer, A. Karlsson, G. Mathiason, and T. Helldin, "Root-cause localization using restricted Boltzmann machines," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 248–255.

[6] B. Steenwinckel, "Adaptive anomaly detection and root cause analysis by fusing semantics and machine learning," in *Proc. Eur. Semantic Web Conf.*, 2018, pp. 272–282.

[7] A. Lokrantz, E. Gustavsson, and M. Jirstrand, "Root cause analysis of failures and quality deviations in manufacturing using machine learning," *Proc. CIRP*, vol. 72, pp. 1057–1062, Jan. 2018.

[8] Y. Liang, X. Xu, X. Wu, Y. Chen, S. Dai, J. Yu, and L. Ruie, "Network fault root cause analysis algorithm based on deep learning," in *Proc. 12th Int. Conf. Comput. Eng. Netw.* Singapore: Springer, 2022, pp. 349–358.

[9] A. Saleem, M. Raeiszadeh, A. Ebrahimzadeh, R. H. Glitho, J. Eker, and R. A. F. Mini, "A deep learning approach for root cause analysis in real-time IIoT edge networks," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, May 2023, pp. 1–5.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[11] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–13.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[13] L. A. Zadeh, "Fuzzy set," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[14] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106275.
[15] N. Talpur, S. J. Abdulkadir, H. Alhussian, M. H. Hasan, N. Aziz, and A. Bamhdi, "A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods," *Neural Comput. Appl.*, vol. 34, no. 3, pp. 1837–1875, Feb. 2022.
[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[17] R. Steinert and D. Gillblad, "Long-term adaptation and distributed detection of local network changes," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2010, pp. 1–5.
[18] S. Diaz, J. Escudero, and J. Luque, "Expert system-based alarm management in communication networks," in *Proc. ICEIS*, 2000, pp. 116–120.
[19] R. N. Cronk, P. H. Callahan, and L. Bernstein, "Rule-based expert systems for network management and operations: An introduction," *IEEE Netw.*, vol. 2, no. 5, pp. 7–21, Sep. 1988.
[20] M. Klemettinen, H. Mannila, and H. Toivonen, "Rule discovery in telecommunication alarm data," *Netw. Syst. Manage. J.*, vol. 7, no. 4, pp. 395–423, 1999.
[21] Y. Chen and J. Lee, "Autonomous mining for alarm correlation patterns based on time-shift similarity clustering in manufacturing system," in *Proc. IEEE Conf. Prognostics Health Manage.*, Jun. 2011, pp. 1–8.
[22] J. Wang, C. He, Y. Liu, G. Tian, I. Peng, J. Xing, X. Ruan, H. Xie, and F. L. Wang, "Efficient alarm behavior analytics for telecom networks," *Inf. Sci.*, vol. 402, pp. 1–14, Sep. 2017.
[23] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, no. 1, pp. 39–59, 1994.
[24] M. A. Mohammed, M. K. A. Ghani, N. Arunkumar, O. I. Obaid, S. A. Mostafa, M. M. Jaber, M. A. Burhanuddin, B. M. Matar, S. K. Abdullatif, and D. A. Ibrahim, "Genetic case-based reasoning for improved mobile phone faults diagnosis," *Comput. Electr. Eng.*, vol. 71, pp. 212–222, Oct. 2018.
[25] G. Costa Silva, E. E. O. Carvalho, and W. M. Caminhas, "An artificial immune systems approach to case-based reasoning applied to fault detection and diagnosis," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112906.
[26] B. Cai, L. Huang, and M. Xie, "Bayesian networks in fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2227–2240, Oct. 2017.
[27] M. Bauer, J. W. Cox, M. H. Caveness, J. J. Downs, and N. F. Thornhill, "Nearest neighbors methods for root cause analysis of plantwide disturbances," *Ind. Eng. Chem. Res.*, vol. 46, no. 18, pp. 5977–5984, Aug. 2007.
[28] S. M. Srinivasan, T. Truong-Huu, and M. Gurusamy, "Machine learning-based link fault identification and localization in complex networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6556–6566, Aug. 2019.
[29] M. X. Cheng and W. B. Wu, "Data analytics for fault localization in complex networks," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 701–708, Oct. 2016.
[30] V. C. Ferreira, R. C. Carrano, J. O. Silva, C. V. N. Albuquerque, D. C. Muchaluat-Saade, and D. Passos, "Fault detection and diagnosis for solar-powered wireless mesh networks using machine learning," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 456–462.
[31] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors J.*, vol. 18, no. 1, pp. 340–347, Jan. 2018.
[32] W. Jiang, "Internet traffic prediction with deep neural networks," *Internet Technol. Lett.*, vol. 5, no. 2, p. e314, Mar. 2022.
[33] J. Lai, Z. Chen, J. Zhu, W. Ma, L. Gan, S. Xie, and G. Li, "Deep learning based traffic prediction method for digital twin network," *Cognit. Comput.*, vol. 15, no. 5, pp. 1748–1766, Sep. 2023.
[34] V. K. Navya, J. Adithi, D. Rudrawal, H. Tailor, and N. James, "Intrusion detection system using deep neural networks (DNN)," in *Proc. Int. Conf. Advancements Electr., Electron., Commun., Comput. Autom. (ICAECA)*, Oct. 2021, pp. 1–6.
[35] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
[36] S. Zhang, Z. Sun, M. Wang, J. Long, Y. Bai, and C. Li, "Deep fuzzy echo state networks for machinery fault diagnosis," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1205–1218, Jul. 2020.
[37] G. Wang and J. Qiao, "An efficient self-organizing deep fuzzy neural network for nonlinear system modeling," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2170–2182, Jul. 2022.

BIXIAN ZHANG received the M.S. degree from Fuzhou University, Fuzhou, in 2011. She is currently an Associate Professor with the School of Computing and Information Science, Fuzhou Institute of Technology. Her research interests include the communication network engineering and network function virtualization (NFV).