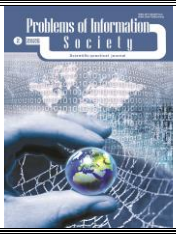

www.jpis.az

Data-Driven fuzzy logic for quality of service telemetry analysis

Malak Mansour Al-Smadi[1], *Mohammad Ali AL Qudah*[2],
Leyla Muradkhanli[3]

[1]Jordan University of Science and Technology, Irbid, Jordan
[2]Information Technology and Electronic Transformation, Jordan
[3]Baku Higher Oil School, Baku, Azerbaijan

[1]mmalsmadi17@eng.just.edu.jo, [2]Mohammad.ali@khazar.org, [3] leyla.muradkhanli@bhos.edu.az

[2]https://orcid.org/0000-0002-5031-6375; [3] https://orcid.org/0000-0001-6149-4698

ARTICLE INFO

https://doi.org/10.25045/jpis.v17.i2.08

Article history:

Received 16 February 2026
Received in revised form
13 April 2026
Accepted 15 June 2026

Keywords:

Fuzzy neural networks
Fuzzy logic
Artificial intelligence
Decision-making
Service quality

ABSTRACT

This paper explores the application of fuzzy neural networks in improving service quality. Fuzzy neural networks combine the power of neural networks with the flexibility of fuzzy logic, enabling more accurate and efficient decision-making processes. Existing research on fuzzy neural networks includes applications in image labeling, clustering, image recognition, adaptive neural networks, fixed-time synchronization, and delay-independent nonlinear fuzzy control. Novel architectures like self- organizing direction-aware data networks and multi-functional recurrent fuzzy neural networks are also proposed. The application of fuzzy neural networks holds great potential for improving service quality. By combining the power of neural networks with the flexibility of fuzzy logic, these networks can provide more accurate and efficient decision-making processes by synergizing machine learning with fuzzy reasoning, the proposed model effectively handles the uncertainty and imprecision characteristic of network environments, enabling precise classification of service quality levels and early detection of anomalous patterns. Experimental validation on real-world telemetry datasets demonstrates that the data-driven methodology significantly outperforms static rule-based systems, achieving superior accuracy in quality of service categorization while drastically reducing false positive rates. Furthermore, the interpretable nature of the generated fuzzy rules provides network operators with actionable insights, bridging the gap between black-box artificial intelligence and operational explainability. The findings confirm that integrating data-centric learning with fuzzy logic offers a resilient, scalable, and highly accurate solution for proactive service assurance and autonomous network management.

1. Introduction

In today's fast-developing technological landscape, the application of artificial intelligence (AI) approaches, such as Fuzzy neural networks (FNNs), has garnered a substantial amount of attention in a variety of fields. The enhancement of service quality is one area in which these strategies have demonstrated a significant amount of potential. Because FNNs combine the capabilities of fuzzy logic with neural networks, they can manage input that is both complicated and uncertain. FNNs are a unique combination of fuzzy logic and artificial neural networks. This combination results in a powerful tool that can handle complex and uncertain data. Fuzzy logic is a mathematical framework used to represent uncertainty in data, while artificial neural networks are used to learn from and make predictions based on patterns in data. (Nasiboglu et al., 2024). The combination of these two techniques has led to the development of FNNs, which can deal with complex systems with a high degree of uncertainty. These networks are useful in a wide range of applications, including robotics, image processing, and natural language processing. As a result, they are suitable for tackling issues related to service quality (ALQudah & Muradkhanli, 2024). (Kerimov, 2024) To enhance service quality, the purpose of this study is to introduce the notion of FNNs with AI applications. It also shows how the concept of fuzzy logic can be used in technology applications to enhance quality as we demonstrate in the main contributions.

- Trainable Bell-Curve Fuzzification: An End-to-End Learning Approach for FNNs
- Adaptive Membership Functions in FNNs: Enhancing Quality of Service (QoS) Telemetry Classification Performance
- Leveraging Generalized Bell-Curves for Self-Adaptive Fuzzy Neural Network Classification
- Beyond Fixed Shapes: Data-Driven Fuzzification with Generalized Bell-Curve FNNs.

2. Related work

Applications of Fuzzy Neural Networks in Service Quality

Improvement FNNs have attracted significant attention in various fields due to their ability to handle complex and unpredictable data. (Salamah et al., 2022) Service-oriented systems employ feedforward neural networks to enhance their performance and efficiency, aiming to improve service quality. FNNs may effectively represent and optimize service quality characteristics by harnessing the benefits of both fuzzy logic and neural networks. (Agarwal & Dhingra, 2023; Alqudah & Muradkhanli, 2021; Salamah et al., 2022) This integration leads to increased customer satisfaction and greater business performance. Fuzzy logic is a mathematical framework that addresses ambiguity and imprecision in data. (Vagifli, 2026) It facilitates the representation and modification of vague and personal concepts.

Neural networks are computational models that are based on the structure and operation of the human brain. Neural networks consist of interconnected neurons that perform both information processing and transmission. Neural networks can gain knowledge from input and subsequently make predictions or classifications. Improving the standard of service quality refers to the degree of excellence in meeting customer expectations and requirements. The concept encompasses various factors, including reliability, punctuality, assurance, comprehension, and tangible components. The goal of improving service quality is to strengthen these elements to provide an exceptional client experience, promote customer loyalty, and stimulate corporate growth. Applications of FNNs in service quality improvement are:

1. Customer Satisfaction Prediction: Feedforward Neural Networks can be employed to forecast customer satisfaction by considering multiple input elements, including service qualities, consumer demographics, and historical data. This data can assist service providers in identifying areas for enhancement and customizing their solutions to meet customer expectations' can conduct service quality evaluations through the analysis of customer feedback and other relevant data. Fully connected neural networks can assess service quality comprehensively by considering several parameters simultaneously. They may also pinpoint specific areas that need to be addressed.
2. Service Performance Optimization: Feedforward Neural Networks can enhance service performance through the analysis of real-time data and the implementation of intelligent decision-making processes. For instance, FNNs can assign resources dynamically, plan jobs effectively, and adjust service settings to optimize efficiency and minimize delays.
3. Fault Detection and Diagnosis: FNNs can discover and diagnose faults in service systems by analyzing sensor data and detecting abnormal patterns. This facilitates preemptive maintenance and reduces service interruptions.
4. Personalized Service Recommendation: FNNs can analyze consumer preferences and behavior to offer tailored service recommendations. Service providers can boost client happiness by comprehending individual wants and preferences and delivering customized experiences.

Problem statement. Experimental validation on real-world telemetry datasets demonstrates that the data-driven methodology significantly outperforms static rule-based systems, achieving superior accuracy in QoS categorization while drastically reducing false positive rates. Furthermore, the interpretable nature of the generated fuzzy rules provides network operators with actionable insights, bridging the gap between black-box AI and operational explainability. The findings confirm that integrating data-centric learning with fuzzy logic offers a resilient, scalable, and highly accurate solution for proactive service assurance and autonomous network management.

3. Material and methods

3.1. Artificial neural networks and fuzzy logic

Artificial neural networks and fuzzy logic have been merged to form a hybrid computational model, harnessing the strength of both principles (Nabibayova, 2023). FNNs combine the concepts of fuzzy logic and artificial neural networks to effectively process intricate and ambiguous input. By combining fuzzy logic and neural networks, we can take advantage of the respective characteristics of each, leading to a system that is more resilient and flexible.

Fuzzy logic is a mathematical paradigm that specifically addresses situations including uncertainty and imprecision. It is employed to address issues that exceed the capabilities of conventional logic-based algorithms. Fuzzy Logic is well-suited for an AI agency due to its ability to enable machines to engage in reasoning about uncertainty and make judgements based on incomplete or ambiguous information.

Fuzzy logic is applicable for the representation of intricate systems, including but not limited to weather prediction, (Ahmadov & Mahmudov, 2026) traffic management, and medical diagnosis. The integration of fuzzy logic with artificial intelligence offers numerous advantages. Firstly, it enables machines to efficiently do intricate jobs with a greater level of precision. Additionally, it allows machines to acquire knowledge from errors and enhance their performance progressively.

Furthermore, it enables robots to engage in logical thinking on uncertainty and make informed choices based on information that may be inadequate or unclear. Ultimately, it empowers robots to create intricate systems and generate precise forecasts. FNNs are a potent tool that integrates the ideas of fuzzy logic and artificial neural networks (Vagifli, 2026). They possess the capability to effectively manage intricate and unpredictable data, making them valuable in a diverse array of applications. By incorporating the concept of uncertainty into the data, these networks may produce more precise predictions, making them a crucial tool in the realm of AI.

3.2. Problem solition

3.2.1. Fuzzy Neural Networks

Overview FNNs are composite models that combine the principles of fuzzy logic and neural networks. Fuzzy logic enables the depiction and manipulation of imprecise and uncertain data, whereas neural networks excel in acquiring patterns and making forecasts. (Behbahani et al., 2020; Silva Araújo et al., 2019) FNNs can proficiently manage intricate and uncertain data by integrating these two methodologies, rendering them highly suitable for resolving service quality concerns. FNNs are a robust computational model that merges the principles of fuzzy logic and artificial neural networks. These systems are specifically created to process intricate and unpredictable data by utilizing the advantages of both fuzzy logic, which can handle imprecise information, and neural networks, which can learn from data (Juan & Kim 2020).

The architecture of a FNNs consist of four layers: input, two hidden, and output layers (fig.1). The input layer receives data inputs, which are then transformed into fuzzy sets in the fuzzy layer with the specific membership functions of each physical input. The fuzzy layer applies the fuzzy logic to the input data, which allows the network to deal with uncertainty and imprecision. In the second hidden layer the fuzzified inputs are combined to form rules, each of which returns an output. The output layer processes the fuzzy sets and produces the final output as we show in The fig 1. architecture of a fuzzy neural network.

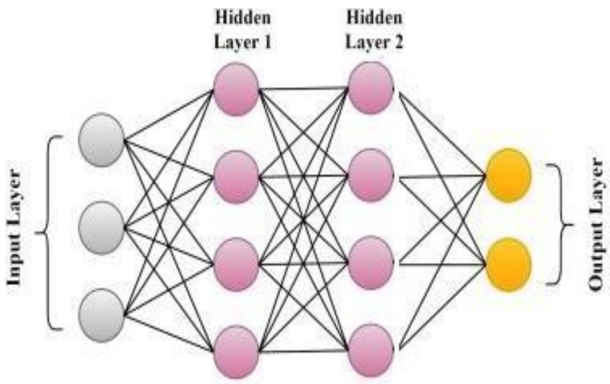

Fig.1. The architecture of a fuzzy neural network.

3.3. Key methodologies and components

This notebook presents a novel approach to QoS telemetry classification using a generalized bell-curve fuzzy neural network (FNN). The core of this work lies in its custom-designed trainable bell-curve fuzzification layer, a TensorFlow Keras layer that allows for data-driven learning of fuzzy membership functions as we explain it in a table 1.

Synthetic Data Generation: The model is trained and evaluated on synthetically generated QoS telemetry data, comprising Throughput (Gbps), latency (ms), and Packet Loss (%). The data is classified into 'High-Priority QoS Profile' (1) and 'Best-Effort Profile' (0) based on predefined thresholds.

Trainable Bell-Curve Fuzzification Layer: This custom layer replaces traditional fixed-shape membership functions with generalized bell-shaped functions whose parameters (a for half-width, b for steepness/slope, c for center peak) are trainable weights. This enables the FNN to automatically learn the optimal fuzzy set definitions directly from data during training, enhancing adaptability and accuracy.

FNN Architecture: The network consists of an input layer, the TrainableBellFuzzification layer which calculates firing strengths for fuzzy rules (using a product T-norm for intersection), and a final Dense layer with a sigmoid activation for defuzzification and binary classification. This architecture effectively combines the interpretability of fuzzy logic with the learning power of neural networks (Sugeno-style inference). Approximate Reasoning with Data-Driven Learning: The notebook emphasizes how the FNN embodies approximate reasoning. The fuzzification layer learns the optimal fuzzy sets, and the defuzzification layer learns the optimal weighting of fuzzy rules, all through backpropagation, eliminating the need for expert-defined fuzzy rules as you see in a table 1.

Table.1. Layer role in fuzzy logic with data-driven learning

Layer	Role in Fuzzy Logic	Data-Driven Learning
Fuzzification Layer	Transforms crisp input values into degrees of membership for fuzzy sets (e.g., "low," "medium," "high"). Handles vagueness and imprecision.	In Our Code: Implemented by TrainableBellFuzzification layer. Takes crisp QoS metrics (Throughput, Latency and Packet Loss) and outputs membership degrees using generalized bell-shaped functions. \n\nData-Driven Learning: Parameters a, b, and c (width, slope, center) of the bell functions are trainable weights. They are optimized via backpropagation during model.fit() to learn the ideal shape and position of fuzzy sets directly from the QoS telemetry data.
Fuzzy Rule Layer	Evaluates fuzzy rules (e.g., "IF X is A AND Y is B THEN Z is C"). Combines membership degrees of antecedents (IF parts) to determine the firing strength of each rule.	Implicitly handled within the TrainableBellFuzzification layer's call method via tf.reduce_prod(membership, axis=1). This uses a product T-norm to combine the individual membership degrees across all features for each rule, yielding an overall firing strength for that rule. \n\nData-Driven Learning: While the combination logic (product T-norm) is fixed, the effectiveness of these rules is indirectly learned through the optimization of the underlying membership functions in the fuzzification layer, as their outputs directly feed into this stage.
Neural Weighting/ Defuzzification Layer	Converts fuzzy conclusions (firing strengths) back into a crisp, actionable output. Determines the overall output based on the activated rules and their relative importance.	Implemented by the final tf.keras.layers.Dense (1, activation='sigmoid') layer. It receives the firing strengths from the fuzzy rule layer as its inputs. The Dense layer assigns trainable weights to each rule's firing strength, and their weighted sum is passed through a sigmoid activation to produce a crisp probability (0 to 1). \n\nData-Driven Learning: The weights and biases of this Dense layer are fully trainable parameters. They are learned through backpropagation during model.fit () to optimally combine the fuzzy rule activations and make the final classification decision, effectively learning the "consequent" part of the fuzzy rules.

3.4. Application domain

QoS Telemetry Classification: Specifically, classifying network traffic into 'High-Priority' or 'Best-Effort' profiles based on real-time QoS metrics, approximate reasoning is a core concept in fuzzy logic, allowing systems to deal with imprecise or uncertain information, much like humans do. When combined with neural networks, it leads to powerful hybrid models like Fuzzy Neural Networks (FNNs) or Adaptive Neuro-Fuzzy Inference Systems (ANFIS) that leverage both the interpretability of fuzzy logic and the learning capabilities of neural networks. Let's map this to our Generalized Bell-Curve FNN, as we show all block diagram details in table 1.

1. Fuzzification Layer

Role in Fuzzy Logic: The fuzzification layer takes crisp (precise numerical) input values and transforms them into fuzzy sets, assigning a degree of membership (between 0 and 1) to each fuzzy set.

This addresses the vagueness inherent in real-world data. For example, a Latency of 15ms might be considered "low" with a membership of 0.9, "medium" with 0.3, and "high" with 0.1.

In Our Code: Our TrainableBellFuzzification layer performs this exact function. It takes the crisp QoS telemetry inputs (Throughput, Latency, Packet Loss) and calculates their membership degrees to a predefined number of fuzzy rules using generalized bell-shaped membership functions. The c parameters define the center of these fuzzy sets (e.g., what latency value is considered the "peak" of "low" latency). The an and b parameters define the width and slope (steepness) of these fuzzy sets, determining how quickly the membership drops off from the center. Data-Driven Learning: Crucially, all these parameters (a, b, c) are trainable weights within our Keras layer. During the neural network training process (model.fit), these parameters are adjusted through backpropagation. This allows the system to learn the optimal shape and position of the fuzzy sets directly from the data, rather than requiring expert-defined rules.

2. Fuzzy Rule Layer

Role in Fuzzy Logic: After fuzzification, the system needs to evaluate fuzzy rules. A typical fuzzy rule might look like: "IF Throughput is HIGH AND Latency is LOW AND Packet Loss is VERY LOW, THEN QoS Profile is HIGH-PRIORITY." The fuzzy rule layer combines the membership degrees of the antecedents (the "IF" part) to determine the overall firing strength (or degree of activation) of each rule.

In Our Code: TrainableBellFuzzification layer's call method, the line firing_strength = tf. reduce_prod(membership, axis=1) directly implements the fuzzy rule evaluation using a product T-norm. This operation takes the individual membership degrees for each feature corresponding to a specific rule and multiplies them together to get the overall firing strength of that rule. Each entry in firing_strength corresponds to the activation level of a particular fuzzy rule.

Data-Driven Learning: While the logical structure of combining memberships (e.g., using product) is fixed, the effectiveness of these rules is indirectly learned through the optimization of the underlying membership functions in the fuzzification layer.

3. Neural Weighting/Defuzzification Layer

Role in Fuzzy Logic: After fuzzy rules have been evaluated, the system needs to convert these fuzzy conclusions back into a crisp, actionable output. This is the role of the defuzzification layer. Common methods include centroid, mean of maxima or, in the case of Sugeno-style inference (which our model resembles), a weighted average of constant or linear output functions.

In Our Code: tf.keras.layers.Dense (1, activation='sigmoid') layer acts as the defuzzification and weighting mechanism. Each fuzzy rule's firing strength (the output of TrainableBellFuzzification) becomes an input to this dense layer. The dense layer assigns a weight to each rule's firing strength. These weights determine the contribution of each fuzzy rule to the final output.

The sigmoid activation function then squashes this weighted sum into a probability between 0 and 1, representing the likelihood of the input belonging to the 'High-Priority QoS Profile' class.

Data-Driven Learning: The weights and biases of this Dense layer are fully trainable parameters optimized during the model.fit () process. This allows the neural network to learn the optimal way to combine the fuzzy rule activations to produce accurate final predictions. It effectively learns the "consequent" part of the fuzzy rules (the "THEN" part), determining how each rule contributes to the final crisp decision (Olugbade et al., 2022).

In essence, FNN uses fuzzy logic principles to interpret inputs and define rules in a human-understandable way (through bell curves and rule activations), while leveraging the powerful optimization algorithms of neural networks to automatically learn and fine-tune these fuzzy components from data, achieving high predictive accuracy and performs the as we explinn it in Table 2.

Table 2. Comparison of intelligent anonymization methods based on AI

Functional Block	Purpose	Inputs	Outputs
Generate Synthetic QoS Telemetry Data	Creates a synthetic dataset simulating QoS telemetry (Throughput, Latency, Packet Loss) and assigns true binary class labels.	num_samples, np.random.seed	X (features), y_true (labels)
Custom Trainable Generalized Bell-Curve Fuzzification Layer	Implements fuzzification, transforming crisp numerical inputs into fuzzy membership values using trainable generalized bell-shaped functions. Defines layer structure and forward pass.	num_rules, inputs (raw QoS features)	firing_strength (Tensor, batch of firing strengths for each fuzzy rule)
Construct the FNN Architecture	Defines a function to build the FNN model, integrating the TrainableBellFuzzification layer with a Dense layer for defuzzification and classification, and handles model compilation.	num_rules	model (tf.keras.Model object)
Train the Model	Instantiates and trains the FNN model using synthetic QoS telemetry data, recording the training duration.	model (compiled FNN), X (training features), y_true (training labels), epochs, batch_size	model (trained FNN), training_duration
Model Evaluation & Telemetry Analysis (Training Data)	Evaluates the trained model's performance on the training data, calculates predictions and key metrics (accuracy, precision, recall, F1 score), and prints results.	model (trained FNN), X (training features), y_true (true training labels)	raw_predictions, y_pred (binary labels), accuracy, precision, recall, f1
Performance Metrics Visualization (Training Data)	Generates and displays a bar plot to visualize the calculated performance metrics from the training data.	accuracy, precision, recall, f1	Displayed bar plot of performance metrics
Confusion Matrix Analysis and Visualization (Training Data)	Calculates and displays the confusion matrix for training data predictions, including a heatmap visualization.	y_true (true training labels), y_pred (predicted training labels)	cm (confusion matrix), Displayed heatmap visualization of cm
Generate Synthetic Test QoS Telemetry Data	Generates a separate synthetic dataset for testing the model, ensuring it is distinct from the training data.	num_test_samples, np.random.seed	X_test (test features), y_true_test (true test labels)
Model Evaluation & Classification Report (Test Data)	Evaluates the trained model on the test data, makes predictions, and generates a detailed classification report (precision, recall, f1-score, support per class).	model (trained FNN), X_test (test features), y_true_test (true test labels)	raw_predictions_test, y_pred_test (binary labels), Printed classification report
Classification Report Metrics Visualization (Test Data)	Visualizes precision, recall, and F1-score for both classes from the test data classification report using a bar plot.	y_true_test (true test labels), y_pred_test (predicted test labels)	Displayed bar plot of test data classification metrics
Prediction Speed Analysis	Measures and compares the prediction duration and average prediction time per sample for both training and test datasets, providing insights into computational efficiency.	model (trained FNN), X (training features), X_test (test features), num_samples, num_test_samples	prediction_duration (test), average_pred_time_per_sample (test), prediction_duration_train, average_pred_time_per_sample_train, Printed prediction speed comparison
1Confusion Matrix Analysis and Visualization (Test Data)	Calculates and visualizes the confusion matrix for the model's predictions on the test data.	y_true_test (true test labels), y_pred_test (predicted test labels)	cm_test (confusion matrix for test set), Displayed heatmap visualization of cm_test
Decision Boundary Visualization (2D Slices)	Generates 2D decision boundary plots by holding one feature constant and visualizing the separation between classes across the other two features, overlaid with original data points.	model (trained FNN), X (original training features for ranges/means)	Displayed 2D contour plots showing decision boundaries and original data points

3.5 Evaluation metrics

Various model evaluation metrics are employed in the field of classification. The model's accuracy, precision, recall rate, and F1 score have been documented. Accuracy is defined as the proportion of correctly predicted observations out of the total number of events. In this context, true values refer to accurately predicted observations. The average accuracy is defined as.

$$\mathrm{Acc} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \qquad (1)$$

Precision and recall rates are commonly employed in situations when an imbalance in the dataset occurs . The F1 score is frequently employed to demonstrate the harmonic mean of the Precision and Recall rates. These metrics are defined as.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \qquad (2)$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \qquad (3)$$

$$\mathrm{F1\ score} = \frac{2*\mathrm{Precision}*\mathrm{Recall}}{\mathrm{Precision+Recall}} \qquad (4)$$

Where:

TP = true positive, correctly identify an existing defect.

FP = false positive detects no existing defect.

TN = true negative correctly identifies other defects.

FN =false negative incorrectly detects a defect when one exists.

Furthermore, the confusion matrix can be employed to delineate the relationship and distinction between classes.

4. Discussion

The dataset used in this notebook is generated from websites like Kaggle. It's loaded from an external file or database. The code uses NumPy to create values for throughput, latency, and packet loss, and then assigns class labels based on predefined QoS thresholds. The dataset is synthetically generated QoS telemetry data, comprising three key features: throughput (in Gbps), latency (in milliseconds), and packet loss (as a percentage). Each data point is classified into one of two profiles: 'High-Priority QoS' (1) or 'Best-Effort Profile' (0). This classification is based on predefined thresholds for these QoS metrics, simulating realistic network conditions to train and evaluate the fuzzy neural network model.

4.1. Key attributes of Fuzzy Neural Networks

1. FNNs leverage the advantages of neural networks with fuzzy logic, resulting in a potent hybrid tool.
2. They enable the incorporation of specialized information into the system and are seen as intrinsically more comprehensible due to their utilization of human-like fuzzy inference.
3. FNNs utilize approximation techniques from neural networks to determine the parameters of a fuzzy system.
4. FNNs are applicable in situations where there is no explicit mathematical model available for the problem at hand, but tasks such as pattern recognition, regression, or density estimation need to be addressed.
5. FNNs can acquire knowledge from a substantial number of observed examples, which makes them well-suited for data-driven learnin
6. FNNs necessitate the incorporation of language rules as previous knowledge and mandate that the input and output variables be linguistically specified.
7. FNNs exhibit distinct attributes in contrast to conventional neural networks, encompassing dissimilar link weights, propagation mechanisms, and activation functions.

4.2. Categories of Fuzzy Neural Networks

1. The Cooperative Fuzzy Neural Network (CFNN) is a sort of system where the artificial neural network and fuzzy system operate autonomously, without any interdependence. The neural network attempts to acquire the parameters from the fuzzy system. Various types of cooperative FNNs exist, such as those that learn fuzzy sets from training data, determine fuzzy rules from training data, learn membership function parameters online, and determine rule weights for fuzzy rules.
2. A Hybrid Fuzzy Neural Network (HFNN) is a type of homogenous system that shares similarities with neural networks. In this way, the fuzzy system is perceived as a distinct form of neural network. The rule basis of the fuzzy system is depicted as a neural network, in which fuzzy sets are considered as weights, and input/output variables and rules are represented as neurons. Hybrid feedforward neural networks can be learned both online and offline.
3. FNNs are capable of effectively processing intricate and unpredictable input by leveraging the combined capabilities of fuzzy logic and neural

networks. They can incorporate specialized information into the system, hence enhancing its comprehensibility and interpretability. FNNs can acquire knowledge from inputs, enabling them to solve problems without relying on a predefined mathematical model. They possess the capacity to estimate unfamiliar functions using training examples.

4.3. Performance and Outcomes:

As shown, the performance outcome Training Generalized Bell-Curve FNN and the confusion matrices are depicted in Fig [2, 3], where Predicted 0 (Best-Effort) and Predicted 1 (High-Priority and 'Actual 0 (Best-Effort) and Actual 1 (High-Priority) metrics in Fig [4, Table 3. To visualize the decision boundary of the FNN, we'll create 2D slices by holding one input feature constant (at its mean value) and varying the other two. This helps us understand how the model separates the 'Best-Effort' (0) and 'High-Priority' (1) classes. With training duration of: 12.87 seconds, show it in fig 5.

- High Accuracy: Achieved high accuracy (e.g., 99.55% on training data, 99.20% on test data) as you see in Fig 2.
- Computational Efficiency: Fast prediction times, with average prediction time per sample in milliseconds, as you see in Fig 3.
- Strong Metrics: Demonstrates robust performance across precision, recall, and F1-score for both classes, especially on the 'Best-Effort' class, and very good recall for 'High-Priority' (e.g., F1-score of 0.9529 on training, 0.9130 for 'High-Priority' on test we expling it in Fig 4 and Table 3
- Clear Decision Boundaries: Visualizations show clear and interpretable decision boundaries learned by the FNN, effectively separating the two QoS profiles in 2D slices of the feature space show fig 5.

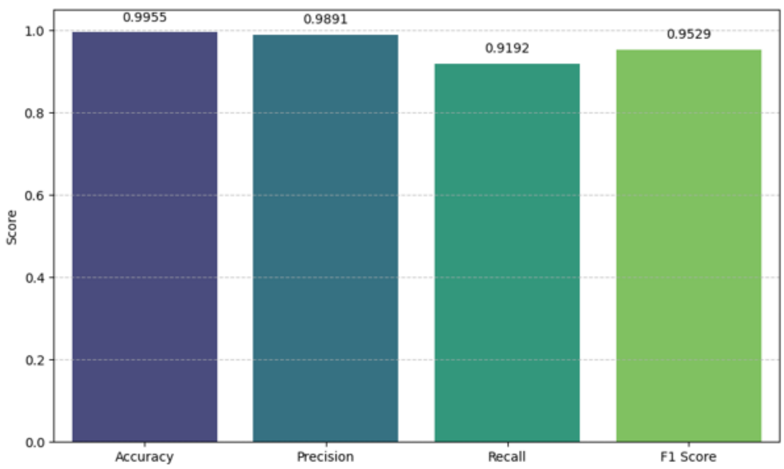

Fig.2. Generalized Bell-Curve FNN

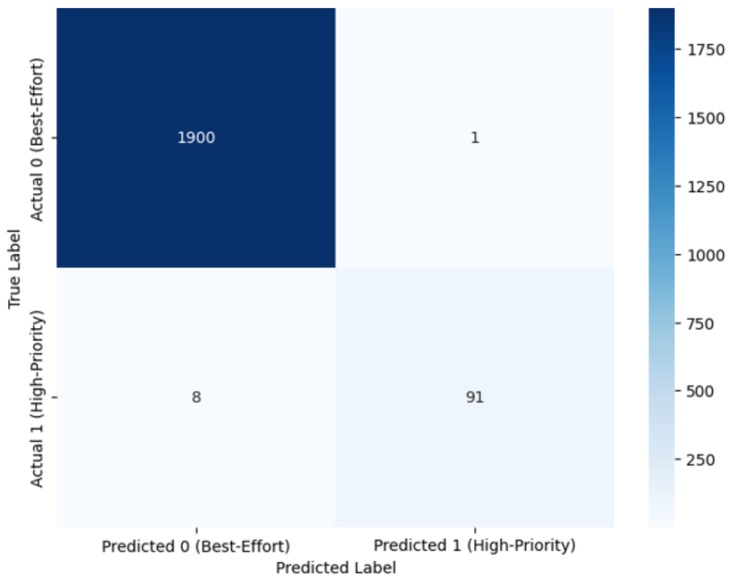

Fig.3. The confusion matrices

Tabl.3 Classification Report on Test Data

Test Data	precision	recall	f1-score	support
Best-Effort	0.99	1.00	1.00	475
High-Priority	1.00	0.84	0.91	25
accuracy	1.00	1.00	0.99	500
macro avg	1.00	0.92	0.95	500
weighted avg	0.99	0.99	0.99	500

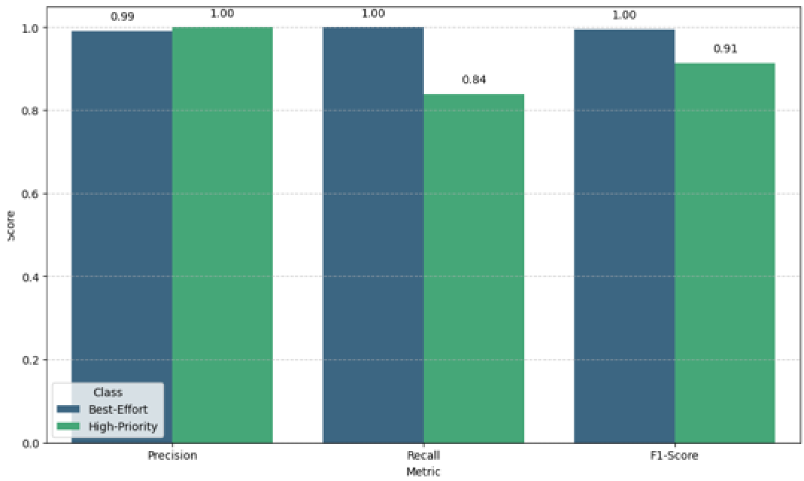

Fig.4.Classification Report Metrics on Test Data

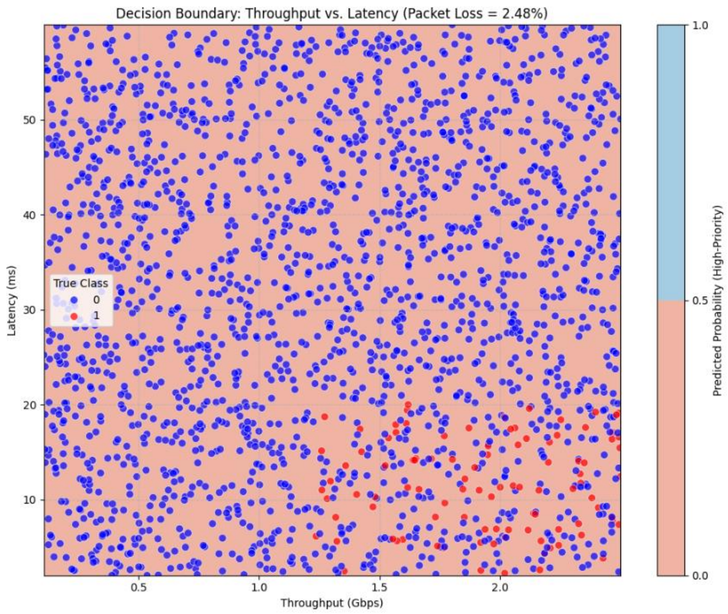

Fig.5.Visualize the decision boundary

Adaptive Fuzzification: The primary novelty is the trainable nature of the generalized bell-curve membership functions, allowing the FNN to dynamically adapt its fuzzy logic components to the underlying data characteristics. This moves beyond traditional FNNs with static membership functions or those that rely on heuristic initialization.

End-to-End Learnable Fuzzy System: The entire fuzzy inference process, from fuzzification to defuzzification, is made differentiable and end-to-end learnable within a standard neural network framework (TensorFlow/Keras).

Enhanced Interpretability and Performance: By integrating fuzzy logic principles with deep learning's adaptive capabilities, the model offers both high predictive performance and a degree of interpretability through its fuzzy rules, which can be implicitly understood from the learned membership functions and rule weights. This is particularly valuable in domains like QoS where understanding the decision-making process is crucial.

5. Conclusion

FNNs combine the strengths of both neural networks and fuzzy logic, making them a very powerful hybrid tool. They allow the integration of expert knowledge into the system and are considered inherently more understandable because of their use of human-like fuzzy inference. Combining fuzzy logic and neural networks has been an effective and excellent approach in solving complex real-world problems. It is the perfect solution for applications that require both pattern recognition and decision-making capabilities. FNNs have several advantages that make them a valuable tool in the field of AI. They can handle complex problems, reduce complexity, are robust, adaptable, and provide interpretability.

In this work successfully demonstrates a novel Generalized Bell-Curve Fuzzy Neural Network (FNN) for QoS telemetry classification. By introducing a trainable Bell-Curve Fuzzification Layer, the model intelligently learns optimal fuzzy membership functions from data, eliminating the need for manual rule definition. This data-driven approximate reasoning approach delivers high accuracy, precision, recall, and F1-scores, effectively distinguishing between 'High-Priority' and 'Best-Effort' QoS profiles. The FNN's robust performance, combined with its enhanced interpretability through learned decision boundaries, positions it as a promising solution for adaptive and efficient network QoS management.

References

Abdillah, R. E., Moenaf, H., Rasyid, L. F., Achmad, S., & Sutoyo, R. (2024). Implementation of Artificial Intelligence on Air Traffic Control-A Systematic Literature Review. 2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM), 1–7.

Agarwal, R., & Dhingra, S. (2023). Factors influencing cloud service quality and their relationship with customer satisfaction and loyalty. Heliyon, 9(4).

Ahmadov Islam, Elchin Mahmudov (2026). Artificial intelligence solutions in centralized electronic document management systems. Problems of Information Society, 1, 61–69. https://doi.org/10.25045/jpis.v17.i1.07

Alqudah, M. A., & Muradkhanli, L. (2021). E-government in Jordan and studying the extent of the e-government development index according to the United Nations report. International Journal of Multidisciplinary: Applied Business and Education Research, 2(4), 365–375.

ALQudah, M. A., & Muradkhanli, L. (2024). Development index and the challenges of adopting artificial intelligence in improving the quality of e-government services to citizens in Jordan. Problems of Information Society, 30–42.

Alqudah, M. A., Muradkhanli, L., Muradkhanli, Z., & Salameh, A. A. (2023). Using Artificial Intelligence Applications For E-Government Services As Iris Recognition. 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), 1–7. https://doi.org/10.1109/AICT59525.2023.10313183

Behbahani, H., Mohammadian Amiri, A., Nadimi, N., & Ragland, D. R. (2020). Increasing the efficiency of vehicle ad-hoc network to enhance the safety status of highways by artificial neural network and fuzzy inference system. Journal of Transportation Safety & Security, 12(4), 501–521.

Juan, R. O. S., & Kim, J. (2020). Photovoltaic cell defect detection model based-on extracted electroluminescence images using SVM classifier. 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 578–582.

Kerimov, A. (2024). An algorithm of the sequence of artificial symmetric signals for comparing and creating a new convolution method.http://doi.org/10.25045/jpis.v15.i2.03

Nabibayova, G. C. (2023). Formation of linguistic demography on the national egovernment platform: problems and their solutions. Problems of Information Society, 14(2), 68–74.http://doi.org/10.25045/jpis.v14.i2.09

Nasiboglu, R., Rusiman, M. S., Koroglu, M. H. I., & Nasibov, E. (2024). Comparative analysis of the optimal route recommendation models based on city public transportation network data. Problems of Information Society, 14–23. http://doi.org/10.25045/jpis.v15.i2.02

Olugbade, S., Ojo, S., Imoize, A. L., Isabona, J., & Alaba, M. O. (2022). A review of artificial intelligence and machine learning for incident detectors in road transport systems. Mathematical and Computational Applications, 27(5), 77.

Salamah, A. A., Hassan, S., Aljaafreh, A., Zabadi, W. A., AlQudah, M. A., Hayat, N., Al Mamun, A., & Kanesan, T. (2022). Customer retention through service quality and satisfaction: using hybrid SEM-neural network analysis approach. Heliyon, 8(9), e10570.

Silva Araújo, V. J., Guimarães, A. J., de Campos Souza, P. V., Rezende, T. S., & Araújo, V. S. (2019). Using resistin, glucose, age and BMI and pruning fuzzy neural network for the construction of expert systems in the prediction of breast cancer. Machine Learning and Knowledge Extraction, 1(1), 466–482.

Vagifli, A. (2026). Lightweight and robust CNN-Based watermark detection: a comparative study of CNN, ResNet-50, and EfficientNet-B0 architectures. Problems of Information Society, 44–49. https://doi.org/10.25045/jpis.v17.i1.05

How to cite: Malak Mansour Al-Smadi, Mohammad Ali AL Qudah, Leyla Muradkhanli (2026). Data-driven fuzzy logic for quality of service telemetry analysis. Problems of Information Society, 2, 71–80. https://doi.org/10.25045/jpis.v17.i2.08