Granular Computing (2024) 9:68
https://doi.org/10.1007/s41066-024-00490-6

ORIGINAL PAPER

Granular neural networks learning for time series prediction under a federated scenario

Mingli Song[1] · Xinyu Zhao[1]

Received: 29 May 2024 / Accepted: 15 June 2024 / Published online: 26 June 2024

Abstract

Granular neural networks (GNNs) are a type of advanced prediction models that produce information granules, offer more abstract and adaptable results. In this study, we address three significant issues in time series prediction within a federated learning (FL) scenario: the management of distributed data, the aggregation of GNNs, and the optimization of granularity levels. Traditional centralized models are insufficient for managing distributed data while ensuring privacy and reducing communication costs, and existing studies on GNNs have not explored their aggregation under a federated framework, which is essential for enhancing model robustness and stability. Additionally, determining the optimal level of granularity for GNNs remains a challenge, impacting the model's predictive accuracy and computational efficiency. To address these issues, we propose a novel federated learning framework that enhances the performance of GNNs for time series prediction. Our approach involves a comprehensive FL framework that enables the collaborative training of local GNNs, refining their granular weights through global aggregation, ensuring better privacy management, and reducing communication overhead. By focusing on the aggregation of parameters within the federated scenario, we enhance the robustness and stability of GNNs which are crucial for effective time series prediction. Furthermore, we determine the optimal levels of information granularity by employing multi-objective optimization techniques, specifically using Pareto fronts to balance the trade-offs between different objectives. Experiments on predicting air quality index for 35 stations in Beijing (China) show the effectiveness of our method.

Keywords Granular neural networks (GNN) · Time series prediction · Federated learning (FL) · PSO

1 Introduction

Granular neural networks (GNNs) have been proposed and designed in several studies according to the location of "information granules": (1) input data are transformed into information granules; (2) the structure of a network is granulated; (3) outputs are information granules; (4) any combinations of the above three cases. No matter which type of GNNs, they all aim to provide more abstract and general models. The existing studies of GNN have defined their structures, shown the learning theories and testified their effectiveness (Pedrycz 2018; Qin et al. 2023). However, there is no study exploring the aggregation of GNNs to form a global GNN. This is the main motivation of our study. In fact, we are also motivated by the wide application of federated learning (FL) (Yang et al. 2019; Zhang et al. 2022; Pedrycz 2023), Which FL technologies are powerful when aggregating GNNs become one of our objectives.

Recently, more and more intelligent devices are equipped on the client sides to collect data. In general, those data are then transmitted to a server side for modeling or analysis. However, this increases the risk of privacy disclosure and the communication cost. In addition, no one can guarantee that there is no information lost during the transmission process. Hence, training a model locally and sending parameters to the server side to obtain some guidance become a popular selection. This is called federated learning. In general, it is hoped to obtain high

✉ Mingli Song
songmingli@cuc.edu.cn

Xinyu Zhao
zhaoxinyu0818@cuc.edu.cn

[1] School of Computer and Cyber Sciences, Communication University of China, Beijing, China

accuracy of local models or pursue global insight into complex problems.

Among the large amount of prediction models of time series, granular models own a special property that they can capture more abstract outputs and reflect more abstract structures which are more like the way human beings' thinking. A single granular neural network reveals the performance of a local data set; whereas the aggregation of several local granular neural networks motivates us to explore whether the global result shows more useful or better knowledge than the local ones. Therefore, the main objective of this work is to design an effective framework which aggregates local granular networks to create a global time series prediction model.

In brief, the main contributions are:

1. We design a comprehensive federated learning framework to interactively train local granular neural networks. In this case, the granular weights obtain a second round of refining from a global perspective.
2. The robustness and stability of GNNs when executing FL strategies are testified and verified using air quality index prediction data.
3. The best level of information granularity of a GNN is determined through drawing pareto fronts and listing values of a hybrid objective function.

The organization of the paper is as follows:

Section 2 introduces several fundamental concepts including GNN, particle swarm optimization (PSO) and federated learning. Section 3 elaborates on our methodology. Section 4 show the experimental studies and comparison. Final is the conclusion in Sect. 5.

2 Preliminaries

In this section, we will introduce some preliminaries: granular neural networks (GNN), optimization of multiple variables with strict constraints, and federated learning.

2.1 GNN

Granular neural networks are a type of abstract models that can deal with both numerical and text data and output corresponding more abstract data units. Researchers have proposed several kinds of granular neural networks with different architectures for different tasks in literature. Melin and Sánchez (2018) delved into multi-objective optimization strategies for modular GNNs in pattern recognition tasks, underlining their versatility. Sánchez et al. (2020) compared variants of particle swarm optimization with fuzzy dynamic parameter adaptation for modular GNNs in human recognition tasks. Al-Hmouz et al. (2015) laid foundational groundwork in granular computing for time series description and prediction. Chen and Chen (2015) showcased the potential of a hybrid fuzzy time series model based on granular computing for stock price forecasting, emphasizing its applicability in financial settings. Ghiasi et al. (2022) addressed uncertainty quantification in pollutant longitudinal dispersion coefficient prediction using a granular computing-based neural network model, highlighting the importance of considering uncertainty in predictive modeling tasks. Song et al. (2023) proposed feature ranking techniques within an improved GNN framework, enhancing interpretability and performance.

In recent years, there are more and more studies on granular computing-based methods for time series prediction. Chen et al. (2019) proposed a novel fuzzy time series forecasting method based on interval ratio and particle swarm optimization (PSO) technique, which exhibited excellent performance. Pant and Kumar (2022) introduced a weighted fuzzy time series forecasting method based on particle swarm optimization and computational algorithms which add to the toolbox of time series prediction techniques. Vovan (2023) explored forecasting models for interval time series using fuzzy clustering techniques and augmented the methodologies available for tackling uncertainty in time series data. Song et al. (2023) demonstrated the efficacy of GNNs in time series prediction and captured complex temporal patterns. Liu and Wang (2024) proposed a method for long-term time series prediction based on fuzzy time series and information granulation, shedding light on handling long-range dependencies in forecasting.

In more recent developments, Karahasan et al. (2024) unveiled a deep recurrent hybrid artificial neural network for forecasting seasonal time series, showcasing advancements in modeling intricate temporal patterns. Song et al. (2024) introduced a hybrid time series interval prediction method by combining a granular neural network and ARIMA to increase prediction accuracy. Furthermore, Song and Wang (2024) presented a complexity-aided time series model which contributes to the evolution of methodologies in the context of granular neural networks. These advancements highlight the growing sophistication and effectiveness in time series forecasting methodologies, especially within the realm of granular neural networks.

This study will concentrate on developing more abstract granular neural networks from the perspective of parameters and thus, we choose granular neural networks introduced by Song and Pedrycz (2013). This kind of granular neural networks is developed on a basis of numeric neural networks and optimized by using swarm optimization algorithms, such as particle swarm optimization (PSO).

Let’s firstly look at the learning process of a GNN. The process is illustrated in Fig. 1.

A numeric neural network is well trained using an original data set and a weight matrix $\boldsymbol{W}$ is returned. A system level of information granularity ε is provided by experts. $\boldsymbol{W} = [w_{ij}]$ together with ε is used to granulate $\boldsymbol{W}$. Some protocols are defined by Song and Pedrycz (2013) and Song et al. (2024) to help the granulation process. Now we obtain a granular weight matrix $\tilde{\boldsymbol{W}}$ and a set of levels of information granularities which are stored in a matrix $\boldsymbol{E} = [\varepsilon_{ij}.]$. The original numeric neural network becomes a granular one due to the granular connections and the granular neural network defines some advanced arithmetical operations for its training. After calculating the objective function value using $\boldsymbol{E}$, $\tilde{\boldsymbol{W}}$ and the data set, a comparison is done to determine whether stopping or starting an optimization process. This kind of optimization problem is often assisted by swarm intelligent algorithms, such as PSO. Finally, the best granular neural network is returned as well as the best set of information granularities.

There are several classical frameworks of information granules, such as intervals, fuzzy sets, and rough sets. In this study, intervals are chosen are the formal information granules. This is because intervals can capture the abstract information of original data with fewer numbers of parameters which determines the subsequent computing resources. Assume that a numeric weight is represented by w_{ij} and its extended interval form is $\widetilde{w_{ij}}$.

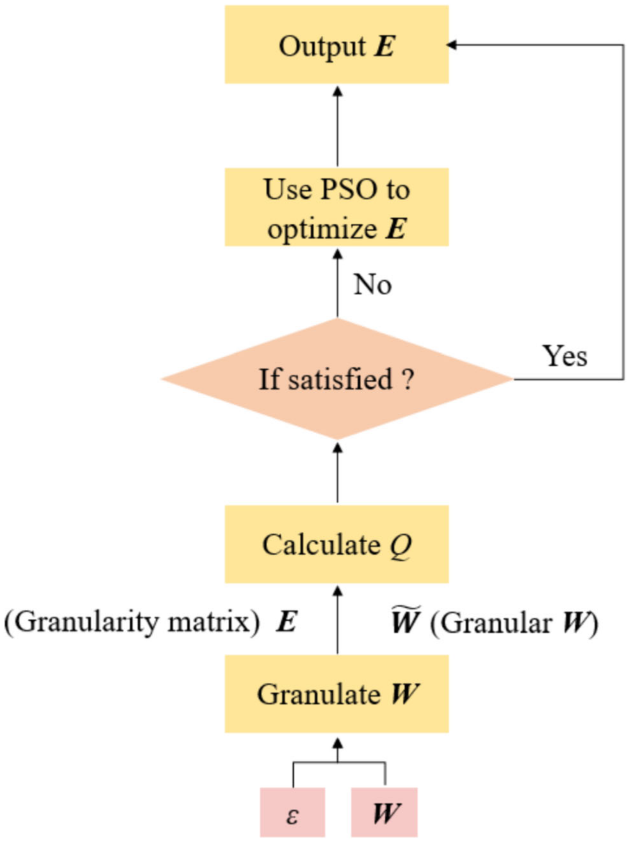
Output $\boldsymbol{E}$
Use PSO to optimize $\boldsymbol{E}$
No
If satisfied ?
Yes
Calculate Q
(Granularity matrix) $\boldsymbol{E}$
$\widetilde{\boldsymbol{W}}$ (Granular $\boldsymbol{W}$)
Granulate $\boldsymbol{W}$
ε
$\boldsymbol{W}$

Fig. 1 The training process of a GNN

$$\widetilde{w_{ij}} = [w_{ij}^-, w_{ij}^+] \tag{1}$$

Thus, a critical issue is how to define w_{ij}^- and w_{ij}^+. To correlate the information granularity and the original weight w_{ij}, we set the information granularity (ε_{ij}) as a parameter like learning rate to control the moving amplitude.

$$w_{ij}^- = w_{ij} - \varepsilon_{ij}^- \times w_{ij} \tag{2}$$

$$w_{ij}^+ = w_{ij} + \varepsilon_{ij}^+ \times w_{ij} \tag{3}$$

All possible allocation protocols are defined in (Song and Wang 2024). In this study, we use the following assumption to simplify the optimization process:

$$\varepsilon_{ij} = \varepsilon_{ij}^- = \varepsilon_{ij}^+ \tag{4}$$

where $\boldsymbol{E} = [\varepsilon_{ij}.]$. This simple setting will decrease the size of optimizing parameters in a half way. If a further better result is required, a refinement operation may be executed locally. Now the number of parameters that need to be optimized is equal to the number of weights (and biases) of the numeric neural network. Note that there should be some constraints between the system level of information granularity ε and each ε_{ij}. Simultaneously, each ε_{ij} should change in a rational range. Therefore, we require:

$$\varepsilon \times n \times m = \sum_{i=1}^{n} \sum_{j=1}^{m} \varepsilon_{ij}, \quad 0 < \varepsilon_{ij} < 1 \tag{5}$$

The equation sets a constraint that there is a system level of information granularity and all parameters under the system are controlled by this information granularity. It formally defines the relationship between the system level and the structure level from the mathematical perspective. The inequation is rational and mainly functions when executing the optimization process.

2.2 Optimization of a set of parameters with strict constraints

A challenging issue in the optimization research field is when optimizing a set of variables that has no guidance for the update process. In this case, heuristic optimization methods provide a competitive alternative, and swarm optimization methods, in particular. Swarm optimization algorithms are computing algorithms that simulate swarm activities in nature or human society. The algorithms learn the interaction and rules among individuals in the swarm.

Particle swarm optimization (PSO) is a type of swarm intelligence optimization technique, proposed by Kennedy and Eberhart (1995). It is inspired by the foraging behavior of bird flocks and simulates the behavior of individual birds within a flock when searching for food. PSO shows strong optimization capabilities in a variety of scientific and engineering fields, especially for optimizing continuous nonlinear problems. It is an iterative process and stops until some conditions are satisfied.

The process begins by randomly generating a group of particles and each particle is described using a random position and a velocity. A fitness function is predefined according to the problem. At each generation, each particle records its best position until now (p_{best}) and the best position among all particles is recorded as well (g_{best}). Each particle's velocity and position are then updated based on p_{best} and g_{best}. The evaluation and updating are repeated until a maximum number of iterations is reached or other termination criteria are met. Thus, the core of PSO is the updating of speed and position. Please refer to the following formula:

$$v_{id}^{(t+1)} = \omega \cdot v_{id}^{(t)} + c_1 \cdot rand_1 (p_{\text{best},id} - x_{id}^{(t)}) + c_2 \cdot rand_2 \cdot (g_{\text{best},d} - x_{id}^{(t)}) \tag{6}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)} \tag{7}$$

where $v_{id}^{(t)}$ and $x_{id}^{(t)}$ represent the velocity and position of the i-th particle in the d-th dimension during the t-th iteration, respectively. $rand_1$ and $rand_2$ are random numbers within the [0,1] interval. $p_{\text{best},id}$ is the best position of the i-th particle in the d-th dimension, and $g_{\text{best},d}$ is the best global solution in the d-th dimension. ω is the inertia weight that controls the preservation of particle velocity. c_1 and c_2 are learning factors which are often regarded as cognitive and social parameters, respectively. When the algorithm terminates, it outputs the best vector and its fitness value, serving as the best solution to the problem. Depending on the above discussion, it seems that any multiple-parameter optimization problems can be solved by using PSO as long as the objective function is predefined or provided.

$$Q = f(data, W, \varepsilon) \tag{8}$$

2.3 Federated learning

With the fast development of Internet of Things (IoT), a plethora of IoT devices is utilized to collect data for analysis or prediction tasks (Zhang et al. 2022). In this case, traditional machine learning schemes become less efficient and more complex because transferring distributed data from many IoT devices to a central location is a complicated and time-consuming process (Tonellotto et al. 2021; Xing et al. 2022). Therefore, federated learning or local–global modeling strategies become popular. When considering privacy or communication issues between different sides, the term "federated learning" is more proper; when modeling issues in client and server sides are the main requirements, the terms "local and global" can be used to describe the related methods as well. In this paper, we concentrate on comparing the performances of federated learning methods with centralized learning methods and not consider the privacy or communication issues.

Recent studies demonstrate the effectiveness of federated learning in diverse domains. Paragliola (2022a, b) applied federated learning in eHealth time-series classification, while Repetto et al. (2022) optimized goal programming for time series forecasting. Truong et al. (2022) developed a light-weight federated learning-based anomaly detection system for time-series data in industrial control systems. Dogra et al. (2023) proposed a federated learning approach for consumer profiling in energy load forecasting. Furthermore, Paragliola (2023) introduced a federated learning-based approach to recognize subjects at high risk of hypertension in a non-stationary scenario. Perifanis et al. (2023) explored federated learning for 5G base station traffic forecasting, improving network management strategies.

Applications of federated learning approaches for time-series predictions are still challenging because there are a lot of complex conditions or requirements (Liu et al. 2020). One important application field is the climate study, especially the air quality issues. The study of air quality index issues is important due to its critical effects on human health, crop growth and daily life. For instance, in Beijing (China), the air pollution issues have attracted more attention than before and many observation stations are allowed to collect different types of gases to monitor air quality. Thus, air quality prediction research may help people to make better decisions and avoid potential risks. This study will run some experiments on the air quality index data to verify the effectiveness of the proposed method.

3 Methodology

We propose a federated learning framework to refine individual information granularity matrix for time series prediction problems. All individual models are built upon numeric neural networks and then extended into granular neural networks. The federated learning strategy is utilized to refine each individual local granular model from both the local and global perspectives. Figure 2 shows the architecture of the framework.

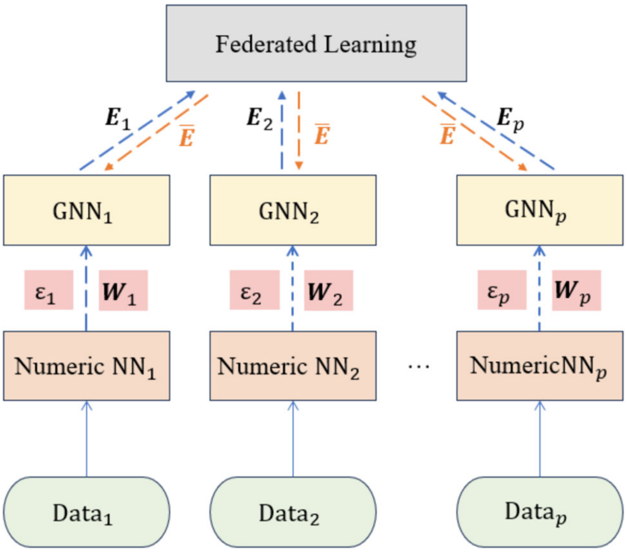

Fig. 2 The architecture of the proposed framework

Assume that there are p data (time series) observation stations or sensors and they constantly collect time series independently. At each station, we build a prediction model using an artificial neural network (shallow or deep). The type of neural networks depends on the complexity of the time series. To make the subsequent aggregation process easier, we require that each local station uses the same architecture of the network, i.e., the same numbers of hidden layers and the same number of input and output neurons. Next, for each station, three items are adopted to develop a GNN: an original time series, a numeric weight matrix, and a system level of information granularity. The system level of information granularity is assigned by experts through their experiences or randomly. For standardization purpose, the information granularity is set in the range of zero and one ([0, 1]). Note that each local system level of information granularity may be the same or not and this issue is another interesting topic that will be further discusses in our future work.

Now, each local GNN is well trained and returns an optimized set of information granularities. Next, we design a federated learning framework to explore whether each local granular model can be further refined. Furthermore, we eager to summarize the future trend of the entire region through the global–local modeling process. Among all federated learning approaches, the federated averaging algorithm shows outstanding performances and is preferred in many papers (Li et al. 2020). Therefore, we choose federated averaging algorithm as the federated learning framework and utilize it to update the parameters of local models. The term "averaging" reflects the way to compute the global parameters: $\tilde{\boldsymbol{E}}$. Please refer to formula (9).

$$\tilde{\boldsymbol{E}}(t+1) = \frac{\sum_{i=1}^{p} \boldsymbol{E}_i(t)}{p} \tag{9}$$

where $\boldsymbol{E}_i(t)$ represents the parameters (weights) of the i-th local model at time t, and $\tilde{\boldsymbol{E}}(t+1)$ refers to the aggregated matrix using all local models at time t. The federated averaging algorithm aggregates all local parameters through computing their mean values. Take neural networks for example. In the training process of each local model, $\boldsymbol{E}_i(t)$ is obtained through the backpropagation method and then $\boldsymbol{E}_i(t+1)$ is obtained at the global model side.

It is an iterative process which includes local iterative processes. The interactive process is illustrated in Fig. 3, and the specific procedure is shown in Algorithm 1.

Note that all local models have identical network structures in Fig. 3. The set of GNN parameters is optimized through employing evolutionary methods like PSO. Therefore, we need to define a hybrid objective function comprising two objectives:

$$Q(Q') = coverage/(width+1) \tag{10}$$

In formula (10), "*coverage*" calculates the ratio that how many real outputs are covered by the intervals of the final model. "*width*" describes the average length of all output intervals. These two criteria are the most important features of information granules and thus are set as two objectives when constructing models.

The algorithm is summarized as follows.

Algorithm 1: GNN with federated learning (GNN-FL)

```
Input: Number of clients $P$, number of global iterations $T$, number
  of local epochs $K$ for PSO, same local model parameter matrix
  $W_0$
Output: granular matrix of each local model $E_p(T)$
Initialization: Initialize all clients with the same local model
  granular matrix $E_0$
for $t = 0, \ldots, T$ do
    for $i \in (1, \ldots, P)$ in parallel do
        Perform $K$ rounds of PSO
        $E_i(t) \leftarrow \mathrm{PSO}(\tilde{\boldsymbol{E}}(t), W_0)$
    end for
$\tilde{\boldsymbol{E}}(t+1) \leftarrow \frac{\sum_{i=1}^{p} \boldsymbol{E}_i(t)}{p}$
end for
```

4 Experimental studies

All the experiments are run on a computer with an Intel Core i7 processor, 16 GB of RAM, and Windows 11 operating system. The programming environment is Visual Studio Code, and the experiments were coded in Python 3.7. For deep learning tasks, the PyTorch framework was employed.

In this section, we apply our method to solve a real time series prediction task: an air quality prediction issue in Beijing, China. The data are collected from 35 monitoring stations and we choose several of them to form groups for FL analysis. The link of the data set is: http://www.bjmemc.com.cn/. We choose the univariate time series and one-step ahead prediction topic. 35 time series of year 2022 are downloaded as the training and testing data. The ratio of the test set to training set is 2:8.

We conduct experiments on three groups of stations, and show their results with and without FL at different granularities. The experiments for each group are repeated five times and the averaged value is taken as the result. The three groups are: (1) station 1 and station 6; (2) station 16 and station 17; (3) station 29 and station 30. Each group has two stations and this number can be modified according to the experts or other conditions. Here, we choose two stations because they show better results in numeric federated learning methods.

We compare the performances of federated learning and centralized learning using GNN through visualizing the hybrid value of the objective function and the individual criterion in the objective function.

4.1 The comparison of federated learning and centralized learning of GNN in terms of the hybrid objective function value

Three different levels of information granularities are adopted (0.005, 0.02 and 0.1) and it can be inferred that higher level of granularity will return larger *coverage* and *width*. To compare the performances of federated learning GNN with centralized learning GNN, we use Q and Q' to represent the two methods, respectively. The definition of the objective function is the same. Please refer to formula (10). Larger values reveal better performance of the model.

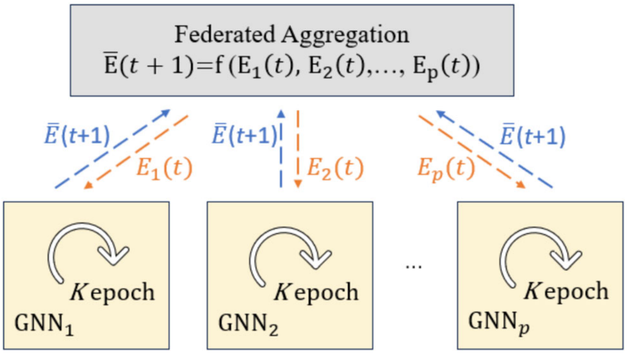

Fig. 3 The interactive process of local models and global side

Table 1 shows the objective values using FL and without using FL for station 1 and station 6. Both stations have similar level of the objective values and with the increasing of the granularity, the objective values decrease. This implies that when the level of granularity increases, the change amplitude of the length of intervals (specificity) is larger than the change amplitude of the *coverage*. In other words, the *coverage* doesn't change much when increasing the granularity. However, the premise is that the granularity is proper to capture enough evidence (*coverage*).

Table 2 reveals similar rules as Table 1. The difference is that the time series from the two stations are not alike. The time series from station 16 seems more suitable for the model because the objective values are larger than the ones of station 17. Another point is there is no big differences between the values of Q and Q'. To further explore the effectiveness of our method, we resort to the two criteria comprised in the objective function in the following section.

Table 3 displays the results of station 29 and station 30. The overall performance of the comparison decreases, that is the objective function values are smaller. This may because for these two stations, smaller levels of granularity can capture knowledge well and we may decrease the granularity. In all three tables, the values of Q are larger than the values of Q'. This shows the robustness of the FL in time series prediction tasks when using GNN as models.

It is obvious that the value of Q is always larger than the value of Q' at all three granularities in Tables 1, 2 and 3. This means that the prediction with FL performs better than without using FL. In other words, our method effectively refines the model parameters through adopting FL strategies.

The above experiments are realized on the groups of two stations and now let's look at the results of more stations. Stations 1, 6, and 2 are collected to form a group due to their close distances on the earth. Centralized modeling and

Table 1 The average value of the objective function (five times) with and without FL under three granularities (stations 1 and 6)

Granularity	Station 1		Station 6	
	Q	Q'	Q	Q'
0.005	0.0565	0.0447	0.0503	0.0452
0.02	0.0290	0.0265	0.0278	0.0254
0.1	0.0188	0.0179	0.0192	0.0178

Table 2 The average value of the objective function (five times) with and without FL under three granularities (stations 16 and 17)

Granularity	Station 16		Station 17	
	Q	Q'	Q	Q'
0.005	0.0571	0.0562	0.0436	0.0429
0.02	0.0359	0.0323	0.0276	0.0251
0.1	0.0256	0.0237	0.0189	0.0184

Table 3 The average value of the objective function (five times) with and without FL under three granularities (stations 29 and 30)

Granularity	Station 29		Station 30	
	Q	Q'	Q	Q'
0.005	0.0413	0.0366	0.0366	0.0355
0.02	0.0272	0.0269	0.0250	0.0242
0.1	0.0203	0.0192	0.0148	0.0142

Table 4 The average of the objective function with and without FL under granularity of 0.005 (station1, 6 and 2)

	Station 1	Station 6	Station 2
Q	0.0494	0.0497	0.0531
Q'	0.0447	0.0452	0.0385

FL modeling are executed on the three stations. Table 4 displays the objective values using FL and without using FL. It is obvious that FL strategies further optimizes the model parameters through increasing the value of Q. There are obvious improvements when using FL, especially in the case of station 2.

4.2 The comparison of federated learning and centralized learning of GNN in terms of subobjectives

Section 4.1 shows the comparison the FL and centralized learning in terms of the single hybrid objective function. However, there are two conflicting subobjectives in the single objective function: *coverage* and *width*. "*coverage*" describes the ratio of the testing samples falling into the output intervals, whereas "*width*" describes the average length of all output intervals. In fact, the two objectives are conflicting. Under optimized results, we draw the values of *coverage* and *width* for different groups.

Figures 4 and 5 show the results of station 1 and station 6. The left axis represents *coverage*, the right axis represents *width,* and the horizontal axis represents three granularities. It can be found that both width and coverage values without FL are slightly higher than the values with FL when granularity is 0.02.

Figures 6 and 7 show the results of station 16 and station 17, respectively. For both *width* and *coverage*, the values are larger when trained without FL.

Figures 8 and 9 show the results of station 28 and station 29. In this case, the two metrics without FL are not consistently higher than those for FL. When the granularity is equal to 0.005, the value of training with FL is higher. And

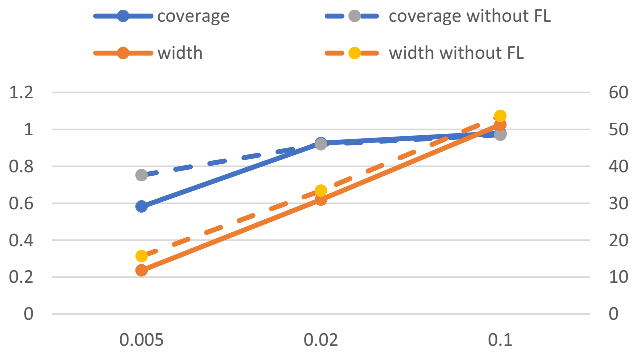

Fig. 4 The *coverage* and *width* of station 1 with and without FL under three granularities

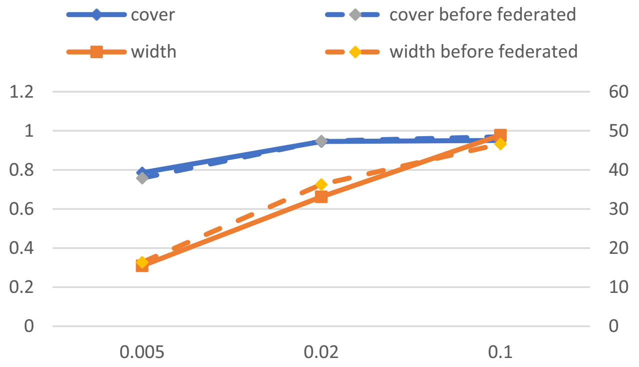

Fig. 5 The *coverage* and *width* of station 6 with and without FL under three granularities

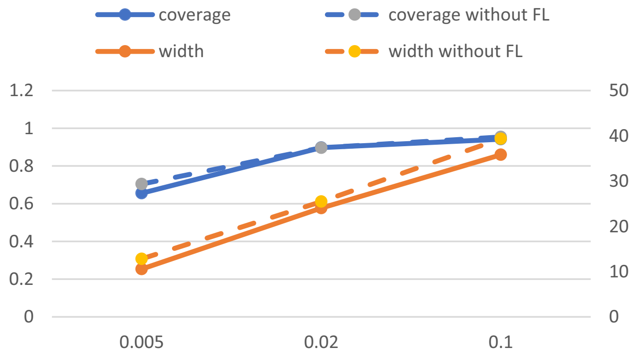

Fig. 6 The *coverage* and *width* of station 16 with and without FL under three granularities

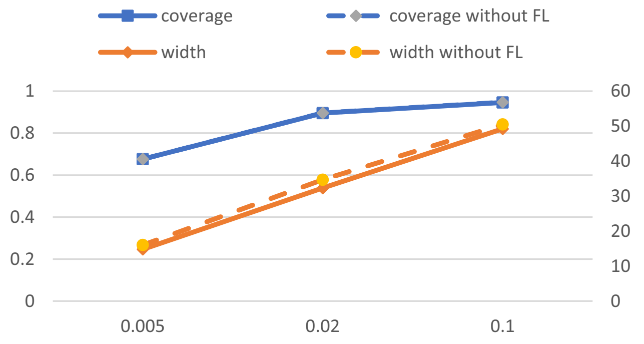

Fig. 7 The *coverage* and *width* of station 17 with and without FL under three granularities

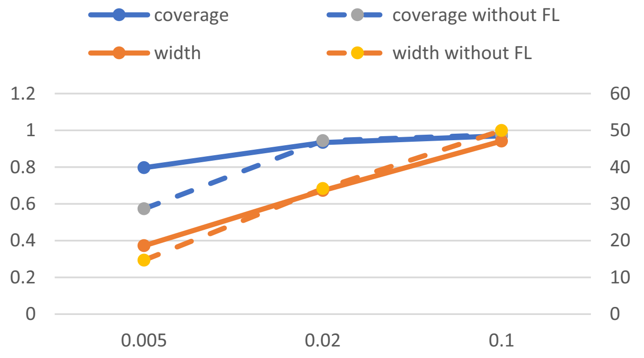

Fig. 8 The *coverage* and *width* of station 29 with and without FL under three granularities

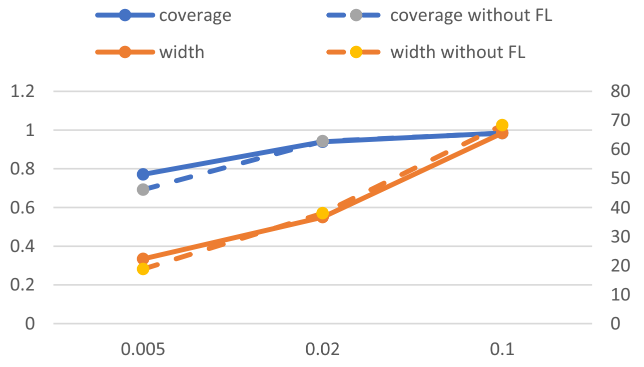

Fig. 9 The *coverage* and *width* of station 30 with and without FL under three granularities

when the granularity is 0.1, the value of training without FL is higher.

The most proper granularity can be determined by several ways. One is through selecting the granularity with the largest objective function value. Another alternative is through observing the pareto front when each subobjective needs to be considered with the same importance. Tables 1, 2 and 3 have list the values of objective functions. Now let's look at the distribution of the two subobjectives: *width* and *coverage*. Please refer to Figs. 10 and 11.

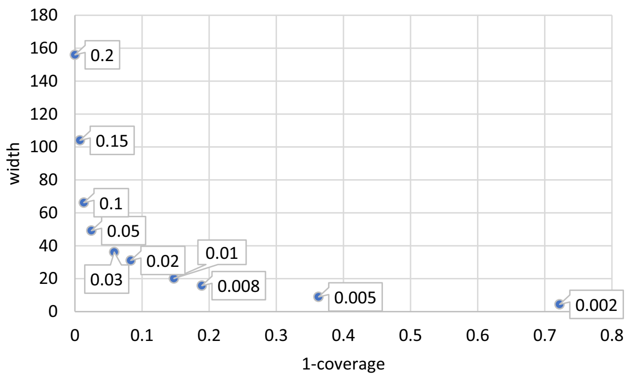

Fig. 10 Performance of station 1 at different granularities with FL (Pareto front)

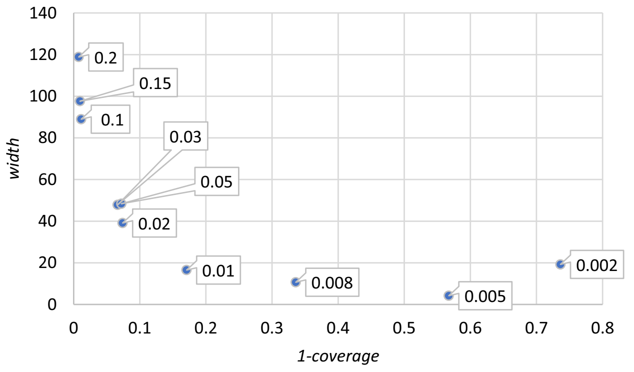

Fig. 11 Performance of station 6 at different granularities with FL (Pareto front)

Figure 10 draws the distribution of ten points which represent different levels of information granularity. Smaller values of both *width* and (1-*coverage*) are preferred. Hence, we may choose the granularity of 0.02 or 0.03 as the best result for station 1.

Figure 11 reveals the performances of ten different levels of granularity on station 6. In this case, the granularity may be determined as 0.02 or 0.01. Although it is obvious when using Pareto front to choose a proper information granularity for an expert, there is no mechanisms or evaluation metrics to automatically determine this critical parameter.

4.3 The optimized GNN structure using FL

No matter FL is adopted or not, the final optimized GNN will return a network with granular structure. To capture the nature of this GNN and the change of GNN when FL is applied, we draw the final GNNs on different stations. Please refer to Fig. 12.

Figure 12 visualizes the granularity matrices of station 1 and station 6 without FL and with FL when the granularity is equal to 0.005. The thickness of the connection lines

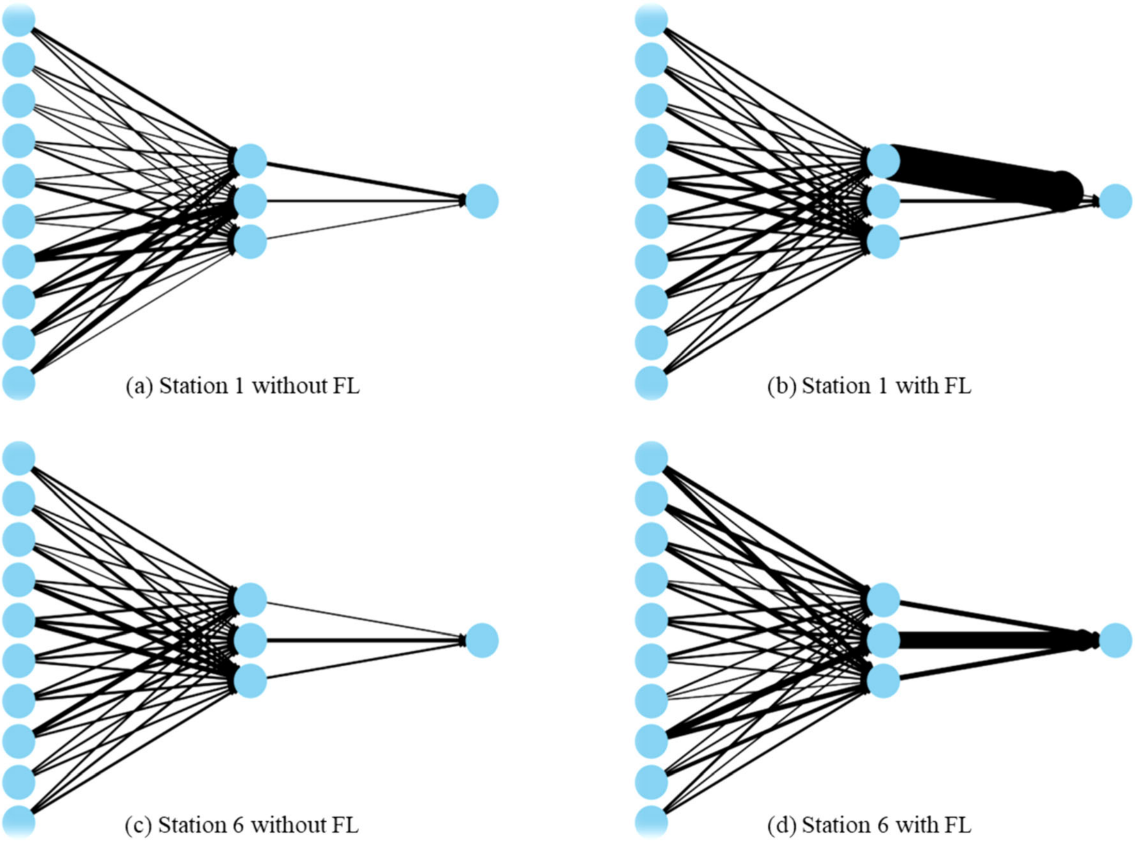

Fig. 12 The final structure of the optimized GNN (granularity is 0.005)

represents the size of the granularity. It can be observed that after FL, the allocation of granularities has dramatic change. In other words, most GNNs trained locally only reach local optima and FL helps move towards global optima.

4.4 The comparison of GNN and numerical neural networks with and without FL

In this section, we compare our interval prediction results with the point prediction results of numerical neural networks. To standardize the metrics, we convert the information granules obtained from intervals into specific values, aligning them with the format of point predictions. The upper and lower bounds of the interval prediction are denoted as g_{up} and g_{up}. The average of the upper and lower bounds of information granules y_{pred} is selected as a representative of the model prediction results (formula (11)), and the model performance is assessed using the following evaluation metrics: MAE and MAPE.

$$y_{\mathrm{pred}} = (g_{low} + g_{up})/2 \tag{11}$$

Table 5 lists the experimental results of four methods under the metrics of MSE and MAPE. The granularity was set to 0.008, and the training epoch was 100. To mitigate the extended learning time associated with large-scale datasets, we compared the predictions of the initial 20 data points in the test set. Four classical methods were used: MLP, GNN, FL_GNN (Granular Neural Network with Federated Learning), and FL_MLP (Multilayer perceptron with Federated Learning).

Table 5 The performances comparison on four methods using MAE and MAPE (stations 16 and 17)

Method	Station 16		Station 17	
	MAE	MAPE	MAE	MAPE
MLP	4.716	8.837	9.416	20.608
GNN	3.763	6.979	8.624	17.865
FL_MLP	3.933	7.102	8.474	17.678
FL_GNN	3.676	6.871	8.449	17.318

From Table 5, it is evident that FL_GNN has the smallest MAE and MAPE values, supporting the efficacy of our proposed framework. Additionally, FL_NN and GNN exhibit comparable performances, while the NN was the most common method among four methods.

5 Conclusion

This study focuses on proposing an interactive refinement of GNN under a federated scenario. The proposed method is then used to solve a multiple-station air quality index prediction problem. Experimental studies reveal that after FL, each local GNN is refined and returns better prediction results in terms of the objective function defined in the paper. However, it is only used for predicting one time series data set and our future work will try to apply it to more time series data sets. Another direction should be noted is the interpretability of GNN. The explainability of artificial neural networks has been an obsession for a long time and GNN also faces this problem. How to explain the granular weights will become our next topic.

Author contributions Mingli Song: methodology and writing. Xinyu Zhao: experiments.

Funding This work was supported by the National Natural Science Foundation of China (NSFC) 61773352 and the Fundamental Research Funds for the Central Universities (CUC24GT12).

Data availability The data used to support the finding are given and cited within the article.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval The work is original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part.

Informed consent Manuscript is approved by all authors for publication.

References

Al-Hmouz R, Pedrycz W, Balamash A (2015) Description and prediction of time series: a general framework of granular computing. Expert Syst Appl 42(10):4830–4839

Chen MY, Chen BT (2015) A hybrid fuzzy time series model based on granular computing for stock price forecasting. Inf Sci 294:227–241

Chen SM, Zou XY, Gunawan GC (2019) Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques. Inf Sci 500:127–139

Dogra A, Anand A, Bedi J (2023) Consumers profiling based federated learning approach for energy load forecasting. Sustain Cities Soc 98:104815

Ghiasi B, Noori R, Sheikhian H, Zeynolabedin A, Sun Y, Jun C, Hamouda M, Bateni SM, Abolfathi S (2022) Uncertainty quantification of granular computing-neural network model for prediction of pollutant longitudinal dispersion coefficient in aquatic streams. Sci Rep 12:4610

Karahasan O, Bas E, Egrioglu E (2024) New deep recurrent hybrid artificial neural network for forecasting seasonal time series. Granul Comput 9(1):19

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of international conference on neural networks (ICNN'95), pp 1942–1948

Li T, Sahu AK, Talwalkar A, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Mag 37(3):50–60

Liu Y, Garg S, Nie J, Zhang Y, Xiong Z, Kang J, Hossain MS (2020) Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach. IEEE Internet Things J 8(8):6348–6358

Liu Y, Wang L (2024) Long-term prediction of time series based on fuzzy time series and information granulation. Granul Comput 9(2):46

Melin P, Sánchez D (2018) Multi-objective optimization for modular granular neural networks applied to pattern recognition. Inf Sci 460:594–610

Pant S, Kumar S (2022) Particle swarm optimization and computational algorithm based weighted fuzzy time series forecasting method. In: Proceedings of international joint conference on advances in computational intelligence (IJCACI), pp 9–23

Paragliola G (2022a) Application of federated learning approaches for time-series classification in eHealth domain. Procedia Comput Sci 207:3545–3552

Paragliola G (2022b) Evaluation of the trade-off between performance and communication costs in federated learning scenario. Future Gener Comput Syst 136:282–293

Paragliola G (2023) A federated learning-based approach to recognize subjects at a high risk of hypertension in a non-stationary scenario. Inf Sci 622:16–33

Pedrycz W (2018) Granular computing for data analytics: a manifesto of human-centric computing. IEEE-CAA J Autom Sin 5(6):1025–1034

Pedrycz W (2023) Advancing federated learning with granular computing. Fuzzy Inf Eng 15(1):1–13

Perifanis V, Pavlidis N, Koutsiamanis RA, Efraimidis PS (2023) Federated learning for 5G base station traffic forecasting. Comput Netw 235:109950

Qin J, Martínez L, Pedrycz W, Ma X, Liang Y (2023) An overview of granular computing in decision-making: extensions, applications, and challenges. Inf Fusion 101833

Repetto M, La Torre D, Tariq M (2022) Goal programming in federated learning: an application to time series forecasting. In: International conference on decision aid sciences and applications (DASA), pp 1672–1677

Sánchez D, Melin P, Castillo O (2020) Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition. J Intell Fuzzy Syst 38(3):3229–3252

Song M, Pedrycz W (2013) Granular neural networks: concepts and development schemes. IEEE Trans Neural Netw Learn Syst 24(4):542–553

Song M, Li Y, Pedrycz W (2023) Time series prediction with granular neural networks. Neural Comput 546:126328

Song M, Wang R (2024) Complexity-aided time series modeling and forecasting under a decomposition-aggregation framework. Inf Sci 120352

Song M, Wang R, Li Y (2024) Hybrid time series interval prediction by granular neural network and ARIMA. Granul Comput 9(1):3

Tonellotto N, Gotta A, Nardini FM, Gadler D, Silvestri F (2021) Neural network quantization in federated learning at the edge. Inf Sci 575:417–436

Truong HT, Ta BP, Le QA, Nguyen DM, Le CT, Nguyen HX, Do HT, Nguyen HT, Tran KP (2022) Light-weight federated learning-based anomaly detection for time-series data in industrial control systems. Comput Ind 140:103692

Vovan T (2023) Building the forecasting model for interval time series based on the fuzzy clustering technique. Granul Comput 8(6):1341–1357

Xing H, Xiao Z, Qu R, Zhu Z, Zhao B (2022) An efficient federated distillation learning system for multitask time series classification. IEEE Trans on Instrum and Meas 71:1–2

Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 10(2):1–19

Zhang T, Gao L, He C, Zhang M, Krishnamachari B, Avestimehr AS (2022) Federated learning for the internet of things: applications, challenges, and opportunities. IEEE Internet Things Mag 5(1):24–29

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.