



Hybrid time series interval prediction by granular neural network and ARIMA

Mingli Song^{1,2} · Ruobing Wang¹ · Yan Li¹

Received: 29 September 2023 / Accepted: 8 November 2023 / Published online: 18 December 2023
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

Abstract

Granular Computing has been successfully applied in many fields to assist modeling while uncertainty exists due to its intrinsic flexibility and comprehensive evaluation on different formats of information granules. The study of residual modeling calls for some flexible tools like Granular Computing techniques to capture the high uncertainty intra the residual. In this study, we devote ourselves to designing a hybrid time series prediction model by developing linear series model and nonlinear (residual) series model separately. After that, a linear information granule aggregation strategy is adopted generate the final predicted result. We focus on the developing a granular model (neural network) to capture the pattern among the highly nonlinear residual and providing an interval residual with upper bound and lower bound which covers as many training data (real residual) as possible and keeps specific enough simultaneously. To verify the effectiveness of our method, a series of experimental studies are executed on several benchmark time series and compared with the ARIMA (Autoregressive Integrated Moving Average) model. Our predicted intervals show a more compact results under the same coverage.

Keywords Interval prediction · Granular neural networks · ARIMA · Cuckoo search · Time series · Residual

1 Introduction

In time series prediction related studies, many researchers prefer probabilistic methods due to the strong ability of probability theory and widely successful application in various fields. However, interval prediction using probabilistic methods only reflect one aspect of the predicted results, i.e., *coverage*, which describes the percentage of covered points of the output intervals. To make the interval prediction results more comprehensive, some researchers

propose Granular Computing based prediction framework which provides two aspects to illustrate the final interval: *coverage* and *specificity*. In fact, the most uncertainty factor of a time series is the residual and there is no study exploring modeling residuals using Granular Computing techniques which becomes the main motivation of this study.

Hence, in this study, we propose a hybrid time series prediction model including Autoregressive Moving Average Model (ARIMA) and a granular neural network which is used to construct a granular model on the residual. A comprehensive discussion on the power of granular neural networks on reflecting the uncertainty of the residual is shown. Please refer to Fig. 1. Assume that a time series can be divided into two parts: a linear subsequence and a nonlinear subsequence. We build a prediction model on the linear part using ARIMA method since it shows effective and strong fitting ability on linear curves (Ning et al. 2022; Luo and Gong 2023). The concomitant residual of ARIMA is often a highly nonlinear complex part, and its prediction becomes a challenge. Some studies start to build individual models on the residual (Wani et al. 2017). Hence, we focus

✉ Mingli Song
songmingli@cuc.edu.cn

Ruobing Wang
wangruobing@cuc.edu.cn

Yan Li
1592563121@qq.com

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

² State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China



Fig. 1 The development process of the hybrid time series prediction model

on developing a more comprehensive prediction model of the residual using a type of granular neural networks (interval granular neural networks). Compared with probabilistic based interval models, this constructed granular neural network comes with both *coverage* and *specificity* metrics which are more rational and in line with human beings' thinking.

In Song et al. (2019), we proposed a type of granular neural networks with granular connections in the network architecture. The main frameworks of information granules (granular) include intervals, fuzzy sets, rough sets, shadow sets, and others. Therefore, the granular neural networks are more robust and stable than the numeric ones. Since the residual of a time series is highly random and may have a large range between the maximum value and the minimum value, it deserves to take a thorough analysis on the prediction modeling process.

In brief, the main contributions of this study are threefold:

- We execute a thorough analysis on the performances of granular neural networks when being used to predict the residual of a time series from both architecture aspect and parameters aspect.
- A training–testing procedure guided by a granular aggregation strategy is designed for a one-step time series predicting problem. It mainly focuses on providing a flexible interval residual with upper bound and lower bound which covers as many training data (real residual) as possible and keeps specific enough simultaneously.
- Cuckoo search algorithm is adopted to optimize the complex information granularity allocation problem and be testified effective.

The remainder of this paper is organized as follows:

Section 2 discusses some studies about different types of output modes of time series prediction models. Then, Sect. 3 introduces an existing classical linear and nonlinear modeling combinations: ARIMA (autoregressive moving average models) plus NN (neural network). This combination is the fundamental architecture of our method. In Sect. 4, we will describe our method in detail from several aspects. Section 5 elaborates on the experimental design and results. Finally, the conclusion is present in Sect. 6.

2 Related work

In this section, we will introduce the development history of time series prediction studies from a perspective of the output mode, i.e., points prediction, probabilistic prediction, and information granules prediction.

2.1 Point forecasting related studies

Time series forecasting issues have been extensively studied in the literature, and from the modeling perspective, it can be classified into three main types (Han et al. 2019): (1) Statistical models, (2) Artificial intelligence models and (3) Hybrid models.

The main traditional statistical models include exponential smoothing, autoregressive moving average models (ARIMA) (Gencer and Başçiftçi 2021) and improved models including seasonal factors (SARIMA) (Dubey et al. 2021), generalized autoregressive conditional heteroskedasticity models (GARCH) (Amirshahi and Lahmiri 2023), etc. These models assume a linear correlation between historical data and the future, describing the relationship between historical data and forecast values in the form of a linear expression through the analysis of their past values. However, they require that the time series must be smooth and linear. Therefore, although these models have achieved good results in early studies, they become inadequate for modeling recent real-world time series when used alone for forecasting (Büyükşahin and Ertekin 2019).

The number of ARIMA research papers has increased rapidly in the last decade. For instance, in the last 3 years, ARIMA has been successfully applied in studying COVID-19 trends (Busari and Samson 2022; Kiarie et al. 2022). Therefore, we consider using ARIMA model to capture the linear character of a time series.

In recent years, the application of artificial intelligence models to solve nonlinear regression problems has shown good results. Various improved models of decision trees such as Random Forest (RF) (Qiu et al. 2017) and Gradient Boosted Trees (GBDT) (Jamei et al. 2023) and their combinations are frequently used for time series prediction, because they have good interpretability. Multi-layer perceptron (MLP), back propagation neural network (BPNN), support vector regression (SVR), and long short-term memory neural network (LSTM) also perform well in various forecasting tasks. Wang et al. (2015) proposed an improved BPNN using adaptive differential evolution algorithm and experimental results show effective improvement in prediction accuracy. Bray and Han (2004) explored the use of support vector machines (SVMs) for time series prediction. Recently, deep learning related methods are successfully applied to solve time series

prediction problems, especially in data-rich tasks. Chimmula and Zhang (2020) apply LSTM models to deal with infectious disease prediction issues.

Due to the limitations of a single model, Zhang (2003) proposed to decompose the time series into a linear part and a nonlinear part and realize modeling separately. Based on the similar concept, Zhang et al. (2019) proposed an ARIMA-SVR approach for predicting daily emergency patient flow, and experimentally demonstrated that the hybrid model has a promising approach with superior performance relative to the single model on this dataset. Büyüksahin and Ertekin (2019) combined the ARIMA-ANN hybrid model with the empirical mode decomposition (EMD) technique to make the generated prediction components more predictable.

No matter what kind of models mentioned above, they finally output a real number as the prediction result which seems not sufficient enough to evaluate the complex time series data sets nowadays. This phenomenon definitely invokes further exploration in this field and probabilistic studies are one of the most successful cases.

2.2 Probabilistic prediction related studies

Although a plethora of time series forecasting models have been studied in terms of point forecasting for target value forecasting, interval forecasting for time series uncertainty is important in many fields such as agriculture, economy, and risk management in the financial industry. In many studies, this uncertainty is described in terms of probabilistic forecasting. Machine learning methods and their combinations with other methods have been widely studied (Bhardwaj et al. 2020; Lv et al. 2022; Rhif et al. 2023). Deep neural networks are a hot topic in recent research to generate probabilistic forecasts by learning to generate the mean and variance of the forecasts. A method that mixes wavelet transform (WT), deep belief network (DBN), and quantile regression (QR) is proposed in (Wang et al. 2016). The method first decomposes the original data with WT to obtain sequences of different frequencies, and then uses DBN to extract different patterns of each frequency. Finally, the probability prediction is derived using quantile regression statistics. Experimental results show that the model has competitive performance. The same probability prediction method combining WT and convolutional neural network is used in Wang et al. (2017). The DeepAR model in Salinas et al. (2020), which is based on an autoregressive recurrent network, was trained with a large number of correlated time series to obtain a probabilistic prediction model.

2.3 Granular prediction related studies

Recent years witness an increasing study on developing granular (fuzzy) models to predict time series. Chen and Hsu (2008) present a method for forecasting the enrollments of the University of Alabama based on the high-order fuzzy time series. Chen and Wang (2010) proposed a method to predict financial time series based on fuzzy-trend logical relationship groups (FTLRGs). Leite et al. (2011) proposed a fuzzy granular evolving model to realize time series prediction which uses information granules in fuzzy rules. Al-Hmouz et al. (2015) tried to form information granules on the original time series and analyze it using fuzzy relation. Chen et al. (2013) proposed a method for fuzzy forecasting based on two-factors second-order FTLRGs and the probabilities of trends of fuzzy-trend logical relationships. Guo et al. (2018) construct information granules on the original time series and use Hidden Markov Models to realize long-term prediction. Chen et al. (2019) proposed a new fuzzy time series forecasting method based on the proportions of intervals and particle swarm optimization (PSO) techniques. Zeng et al. (2019) proposed a clustering-based fuzzy time series forecasting method based on linear combinations of independent variables, the subtractive clustering algorithm and the artificial bee colony (ABC) algorithm. Deep neural networks are also adopted to add information granules information in the structure for time series prediction (Tomasiello et al. 2021). Pant and Kumar (2022) proposed hesitant fuzzy sets-based hybrid time series forecasting method using particle swarm optimization and support vector machine. Vovan (2023) used improved fuzzy clustering technique to develop a predictive model for interval time series, and a new rule is created to interpolate the historical data and forecast the future. All those literatures concentrate on the entire time series, no matter short-term or long-term prediction, or single variate or multiple variates. There is no study about the individual granular modeling while splitting the original time series.

3 A prediction model comprising ARIMA and NN

In this section, we will introduce the hybrid prediction model comprising ARIMA and NN which serves as the fundamental framework of our proposed method. The overview of the entire architecture is displayed in Fig. 2.



Fig. 2 A development process of prediction models using ARIMA and NN

3.1 ARIMA

The ARIMA model is a classical statistical tool in time series analysis, which essentially uses autoregressive and moving average components to separate the signal from the noise in the data and take into account the differential factors. When analyzing time series data, we first do a white noise test on the data. If it is white noise, then ARIMA is considered unnecessary to continue the analysis. If not, do the smoothness test. During the smoothness test, ADF method is often used to test whether there is a unit root in the data. According to the significance test statistic, if it is less than 5% confidence level, there is 95% certainty to reject the original hypothesis. Thus, the original time series is smooth. The most critical issue in ARIMA is the model ordering. Common ways to determine the order are: (1) by observing the trailing truncated tails of the autoregressive and partial autoregressive plots. (2) According to the Akira pool information criterion (AIC) and Bayesian information criterion (BIC). The estimation of the model parameters is generally evaluated by least squares estimation and maximum likelihood estimation. The expression of the model after parameter estimation is as follows.

$$y_t = \mu + \sum_{i=1}^p y_i y_{t-i} + \alpha_t + \sum_{i=1}^q \theta_i \alpha_{t-i} \quad (1)$$

where y_t denotes the value of the series at time t , μ is the mean of the series, e_t is the random error, and p and q are the autoregressive and moving average terms, respectively. The ARIMA model produces a residual series, for which a significance test is also performed. It is generally believed that a proper modeling process will fully and effectively extract the effective information from the series, while making the residual series a white noise series.

3.2 NN

Extracting nonlinear patterns embedded in time series is very strenuous for traditional linear models. Therefore, when dealing with nonlinear sequences, a nonlinear model is needed. Artificial neural networks are one such model. The application of artificial neural networks to time series prediction is relatively mature, where single hidden layer feedforward networks are more widely used in time series modeling and prediction. No matter if shallow neural networks or deep neural networks, they are inspired by the neuronal cells of the human brain and try to fit the original series using a nonlinear strategy. Since we will adopt a shallow (one hidden layer for instance) as our basics, the operation and theory is displayed below.

$$z = \sum_j^n w_{ij} x_j - \theta \quad (2)$$

$$y_k = f(z) \quad (3)$$

where x_j is the j th input variable and y_k represents the k th output. The relationship between input x_j and output y_k is expressed in formulas (2) and (3), where w_{ij} is the connection weight, and θ is the threshold, also known as bias. It can be noted that no matter how many input signals go through the linear combination of connection weights w_{ij} and threshold θ , they are still linear. A simple linear combination cannot solve complex problems. Therefore, a nonlinear function $f()$ is needed as an activation function to offer the ability to solve nonlinear problems.

3.3 Aggregation

Since the objective of developing a prediction model is to predict future tendency (short time or long time), an aggregation strategy should be carefully designed. Now, we will introduce the simple linear aggregation mechanism to aggregate ARIMA and NN model.

Based on the idea of decomposing a time series into a linear sequence and a nonlinear sequence, the residual series e_t is obtained by making the difference between the results $\hat{L}_t$ predicted by the ARIMA model and the actual value Y_t as follows.

$$e_t = Y_t - \hat{L}_t \quad (4)$$

The NN is used to analyze and model the residual sequence e_t . The trained neural network is then used to predict the nonlinear part of the residual sequence, $\hat{N}_t$.

$$\hat{Y}_t = \hat{L}_t + \hat{N}_t \quad (5)$$

Finally, two parts of the prediction result are added to be the final prediction result.

4 A hybrid model comprising ARIMA and GNN

This section presents our hybrid interval prediction model based on a granular neural network and ARIMA. In the study of Zhang (2003), it was found reasonable that a time series comprising a linear autocorrelation structure and a nonlinear part. Based on this theory, we design a time series prediction model including two main phases.

Phase 1: ARIMA modeling. On the premise that none of the data sets used are white noise series, the original time series is differenced to the first or second order according to the ADF test results, and no differencing is required if the series is already smooth. The autocorrelation and

partial autocorrelation plots of the time series are then drawn to initialize the model order. Then parameter estimation is performed for each of the initially selected models. The AIC and BIC values are calculated for each obtained model. The model that is smaller under both AIC and BIC criteria is selected by comprehensive comparison. ARIMA modeling is completed.

Phase 2: GNN modeling. The residual series is obtained after ARIMA and needs to be transformed into a form that can be processed by GNN. The method of sliding window is used in this paper.

Each phase of the proposed model will be explained in detail in this section. In the following, we first provide the fundamental information about GNN, and then describe cuckoo search algorithm. Then the interval prediction method for the hybrid model of granular neural network and ARIMA is presented.

4.1 GNN

Granular neural networks are an extended study based on numeric neural networks proposed in the literature (Song and Wang 2016). It aims to transform traditional numerical weights and biases into information granules and then makes the architecture of a network more robust and general. A level of information granularity ε is assigned to the system and then allocated to each weight and bias of a numerical neural network. Assume all the weights and biases are stored in a vector $\mathbf{v} = [v_1, v_2, \dots, v_p]$. There will be p levels of information granularity ε_i ($i = 1, 2, \dots, p$) allocated onto $\mathbf{v}$. The protocols of allocation can be uniformly allocation or nonuniformly allocation which depends on the users' requirements or the complexity of the problem. However, they all should satisfy an equation:

$$\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_p = p * \varepsilon \quad (6)$$

One simple way to construct interval information granules is:

$$\tilde{v}_i \leftarrow [v_i - v_i * \varepsilon_i, v_i + v_i * \varepsilon_i] \quad (7)$$

In this case, a numeric weight is extended into an interval $\tilde{v}$. There also can be asymmetric strategies which is more complex than the symmetric case. Hence, we choose the symmetric strategy to show the entire development process as a starting point. In general, people hope to develop a well-performed model and therefore, an optimization tool is often required to optimize the set of ε_i . There is an equation constraint for the information granularity set which makes the optimization more challenging especially when the value of p is value.

4.2 Cuckoo search algorithm

We choose cuckoo search algorithm as our optimization algorithm because of its merits in searching for global optimal results (Ong and Zainuddin 2023). Cuckoo search algorithm was inspired by the breeding characteristics of cuckoos, which lay their eggs in the nests of other bird species. The key to the success of this breeding method is to select the most suitable host nest. A study of the cuckoo's host nest selection strategy suggests the following rules.

1. Lay one egg at a time and place it in a randomly selected nest.
2. Nests that successfully reproduce quality eggs will be passed on to the next generation.
3. The number of host nests is fixed and the probability that a cuckoo egg is likely to be found and destroyed.
4. The optimization of a set of parameters looks similar to the above nest selection strategy and many researchers adopt this method as their optimization tools (Zhang et al. 2023; Priya et al. 2022). We choose this method and let ε_i be each host nest. The objective function is designed accordingly in Sect. 4.3.

4.3 The objective function

Granular Computing based prediction framework provides two aspects to illustrate the final interval: *coverage* and *specificity*. The term "*specificity*" reveals the compact level of the final intervals, that is, large length of the interval means a small specificity and small length of the interval means a large specificity. The term "*coverage*" describes the percentage of covered points of the output intervals.

In the conventional time series prediction modeling process, the MSE (mean squared error) or other similar evaluation metrics are adopted. In probabilistic interval prediction modeling process, *coverage* is the single metric along with a certain probability (confidence interval). Later, some interval prediction models are proposed with two evaluation metrics from two different aspects: generality and specificity. Generality is often expressed by *coverage* whereas specificity is mostly computed by the average length of all intervals. The first study defined those two metrics are (Wang et al. 2020) in which Prediction Intervals Coverage Probability (PICP) is defined for *coverage* and Mean Prediction Interval Width (MPIW) is defined for the average length of all intervals.

$$\text{PICP} = \frac{c}{N} \quad (8)$$

where N is the total number of test data and c is the number of true values included in the prediction interval.

$$\text{MPIW} = \frac{1}{c} \sum_{i=1}^c (\tilde{y}_i^{(U)} - \tilde{y}_i^{(L)}) \quad (9)$$

where $\tilde{y}_i^{(U)}$ denotes the upper limit of the prediction interval for the i th sample point on the test set and $\tilde{y}_i^{(L)}$ the lower limit of the interval, and c denotes the number of true values successfully captured by the prediction interval. There is a critical issue about formula (9): it only considers the intervals that cover the real data and calculates their average length. However, sometimes, although an interval doesn't cover a real output, it was just that close. And in this case, the interval still deserves to be looked insight. Thus, in this study, we introduce another average length metric in the following formula:

$$\text{AL} = \frac{1}{N} \sum_{i=1}^N (\tilde{y}_i^{(U)} - \tilde{y}_i^{(L)}) \quad (10)$$

To obtain a better result, we need to follow two main principles. One is to make predicted intervals contain as many real data as possible. That is, larger value of PICP is preferred. The other is to keep the length of each prediction interval as small as possible. That is, smaller value of MPIW is preferred. To comprehensively consider those two factors simultaneously, we define one formula including the two metrics in (11).

$$Q = \frac{\text{PICP}}{\text{AL} + 1} \quad (11)$$

4.4 Algorithm

Assume a time series T containing n sample observations, with y_t denoting the t th observation. Here we only consider the one-step prediction premise. Five stages are designed:

Stage 1: Predict the linear components of the time series with the ARIMA model and obtains the linear part prediction results and the residual series.

Stage 2: Trains an artificial neural network with the residual series to get the network weights and biases.

Stage 3: Use the cuckoo search algorithm to allocate weight granularity to the numeric weights and biases to develop a suitable granular neural network.

Stage 4: Utilize the granular neural network to predict a set of intervals for the residual.

Stage 5: Aggregate the predicted results generated in the first and fourth stages with a linear combination function (summation) to form the final intervals.

Assume that the residual sequence is represented as $e : e_1, e_2, \dots, e_z$. The input data after the sliding window method can be represented as: $w_1, w_2, \dots, w_m$, where $w_1 = e_1, e_2, \dots, e_s$, $w_2 = e_2, e_3, \dots, e_{s+1}, \dots$, $w_m = e_m, e_{m+1}, \dots, e_{m+s-1}$, where s represents the length of the sliding window. The final predicted interval is obtained by adding the results of ARIMA and the interval of the granular neural network for the residual series. To better understand the third and fourth stages of the algorithm, the pseudo-code is provided in Algorithm 1.

Algorithm 1:

Input: A residual sequence $e_1, e_2, \dots, e_z$, system granularity ε

Output: a granular neural network

- 1: Reorganize the residual sequence into a series of sliding windows: $w_1 = e_1, e_2, \dots, e_s$, $w_2 = e_2, e_3, \dots, e_{s+1}, \dots$, $w_m = e_m, e_{m+1}, \dots, e_{m+s-1}$
 - 2: Train a numeric neural network using the sliding windows and return a weights and biases matrix W (the total number of entries in W is equal to p)
 - 3: Initialize a matrix of granularities $E_{n \times p} = [\theta_1; \theta_2; \dots, \theta_n]$, where $\theta_k = [\varepsilon_{k1}, \varepsilon_{k2}, \dots, \varepsilon_{kp}]$, and $\sum_{i=1}^p \varepsilon_{ki} = \varepsilon \times p$
 - 4: For each θ_k , allocate each entry in it onto each entry of W and make each entry of W become an interval, then obtain a granular matrix $\tilde{W}$. Calculate the interval output of the granular neural network with the sliding windows and $\tilde{W}$. Compute the objective function value and record it as q_k . $Q = \{q_k, k = 1, 2, \dots, n\}$
 - 5: Apply Cuckoo search algorithm to update $E_{n \times p}$ according to Q . Then repeat step 4 until stop condition is satisfied.
 - 6: Record the best individual in the last iteration as θ_{best} . At this time, develop the corresponding granular neural network GNN and return.
-

Now, we have completed the entire training process of a GNN. When we face a prediction task, a linear output is obtained through utilizing ARIMA and the input sliding window. A nonlinear part, i.e., the interval residual is obtained by sending the sliding window to the GNN. The aggregation can be simply realized through linear combination of the linear output and the interval residual. Since the residual is often highly nonlinear and there is no obvious pattern to be captured, an effective and powerful tool is needed. In the following experiments section, we will concentrate on testing the performance of GNN from as many as possible aspects.

5 Experimental studies

The experiments were implemented under two programming environments: the ARIMA statistical model was implemented with the help of the R programming language, while the basic part of the numerical neural network was developed using the third-party library keras in python. The experiments were run on a 2.90 GHz CPU.

All the experiments are done under the one-step prediction requirements and can be equally applied to the multiple-step case.

5.1 Datasets and parameters setting

The datasets used in this study include: Wolfe sunspot data, Canada lynx data, and Globtemp data. These datasets have different statistical characteristics and have been extensively studied in the neural networks related literature and statistical literature. This study uses these datasets to demonstrate the validity of the model and compare with the existing studies.

The sunspot dataset has monthly observation samples and annual observation samples. The annual data are used in this experiment, where a total of 221 sample points from 1700 to 1920 are used as training data, while 67 sample points from 1921 to 1987 are used as test data.

The Canada lynx dataset records the number of Lynx captured each year in the Mackenzie River region of northern Canada. The dataset records a total of 114 observations from 1821 to 1934, with observations from 1821 to 1920 used as training data to train the model, and observations from 1921 and 1934 used as test data. It is worth noting that these data were processed with a logarithm based on 10 before our analysis of the Canada lynx data set.

The third dataset is the exchange rate between the British pound and the U.S. dollar. Predicting exchange rates is difficult in the field of time series. Prediction of exchange rates has both academic and practical value for

international trade. This study uses weekly observations from 1980 to 1993, with 731 data points, of which the first 679 data points are used as training data and the last 52 as test data. Following (Zhang 2003), this study uses natural logarithms to preprocess the data.

In the process of establishing numerical neural networks on the residual, a single hidden layer neural network structure is used, in which the sigmoid function is used as the activation function of the hidden neurons and the number of neurons in the output layer is one. Only one hidden layer is enough for small size time series data sets. When the size increases, we can enlarge the number of hidden layers and hidden neurons. The optimizer uses adaptive moment estimation (Adam) and MSE as the loss function. Facing different data epoch is set within 10,000 to make sure the performance of the learning process converges to a stable state.

As a comparison experiment for the ARIMA model we use the same model order as in (Zhang 2003). The sunspot dataset uses ARIMA (9, 0, 0), and the Canada lynx dataset uses ARIMA (12, 0, 0), and the exchange rate dataset is designed with the model ARIMA (1, 1, 0). Those hyperparameters are determined according to the experiments of ARIMA which show comparable better results on those parameters.

5.2 Granular modeling on the residual

In the course of many studies, researchers focus on the accuracy of the model. However, the exploration of the effect of “uncertainty” is neglected, and this phenomenon is more obvious in residual series forecasting. As we mentioned earlier, an important basis for determining whether a model has fully extracted the valid information from the data is to do a white noise test on the residual series. If the residual series passes the white noise test, it means that valuable signals have been extracted, and what remains is noise and interference. However, this conclusion only shows that it is not possible to make accurate point

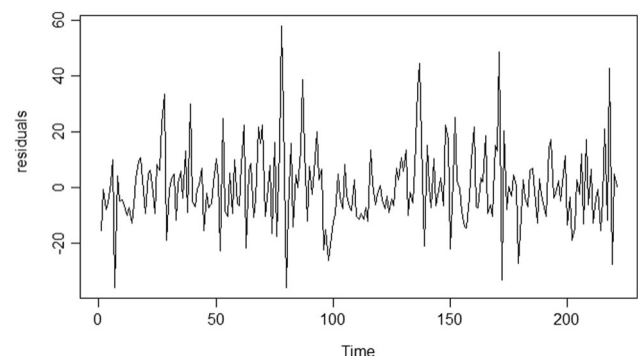


Fig. 3 ARIMA residuals of the Sunspots dataset

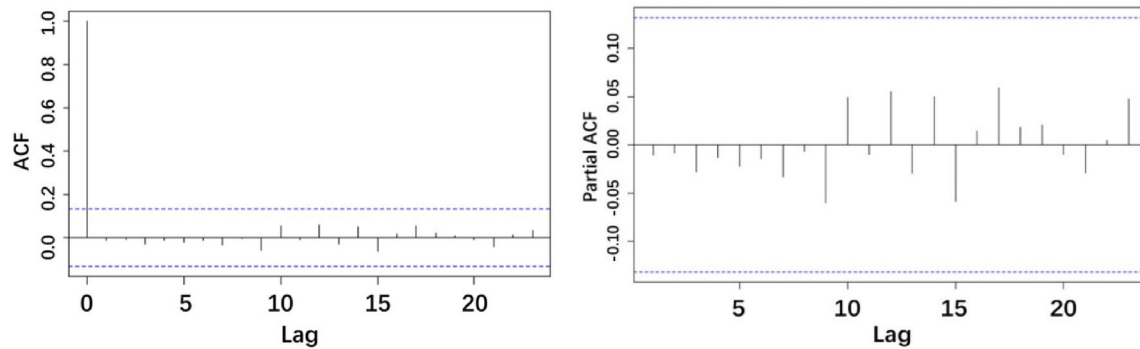


Fig. 4 ACF and PACF plots of the Sunspots residual sequence

predictions on the residual series. Here an attempt is made to account for this uncertainty in the form of an interval prediction.

Taking the Sunspots dataset as an example. The residual series generated after the ARIMA model prediction is shown in Fig. 3. A direct observation shows that the series is irregularly noisy. The autocorrelation plot of the residual series (Fig. 4.) shows that the autocorrelation coefficients of the delayed k -order samples are nearly zero except for the 0-order autocorrelation coefficient which is 1 (due to the presence of random perturbations, the autocorrelation coefficient is not strictly equal to 0). It can be considered to a greater extent that there is no autocorrelation.

The Box–Pierce test on the residual series can more accurately illustrate the white noise characteristics of the residual series. The p value of the test result is 0.8695, which is significantly larger than the critical value (such as 0.05) at the selected confidence level, and the original hypothesis cannot be rejected, i.e., we can consider the series as white noise. The critical value will be further used in the comparison study which is often set as 0.2 and 0.05.

With these settings, the confidence intervals are 80% and 95%.

In this experiment, a granular neural network is used to make interval predictions for the residual series to describe the uncertainty intra the residual series. To observe and analyze the effectiveness of the GNN on the residual modeling, we draw the predicted intervals of two time series data sets in Figs. 5 and 6. For each data set, a set of different levels of information granularity are used to run the program and here we only draw two cases. As to the Sunspot data set, the upper bound and lower bound are more smoothly when the system granularity is larger (0.15). In addition, they cover more real output data points. As to the Lynx time series, the smaller value of system granularity (0.12) can cover almost all data points although there are some sharp breaks in the curves (the bounds and the real output). In addition, the bounds are more compact whilst covering enough real output points.

Together with the predicted interval residual, we add the linear part and outcome the final output. The final upper bound and lower bound along with the real time series of

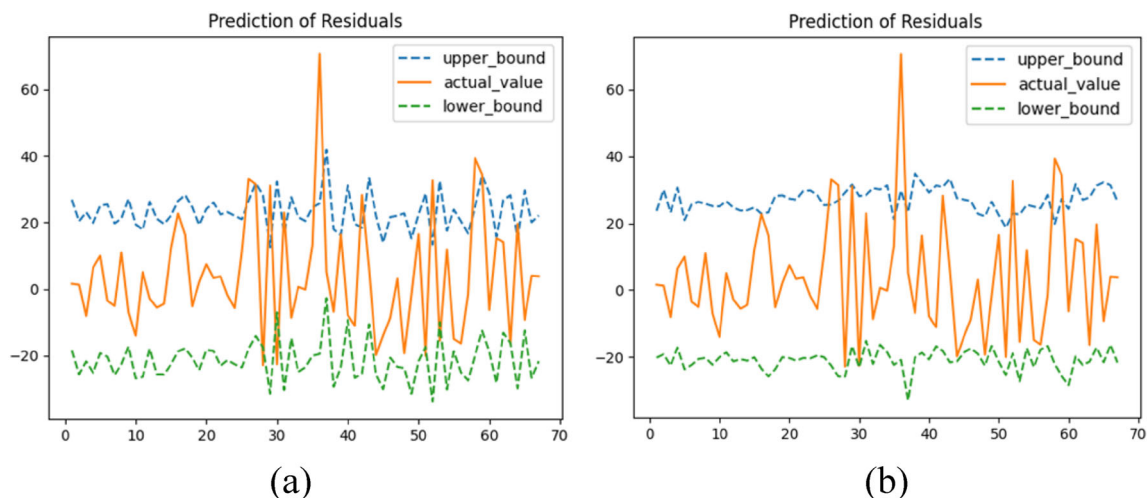


Fig. 5 The output upper bound and lower bound of the residual, Sunspot data, **a** $\varepsilon = 0.12$; **b** $\varepsilon = 0.15$

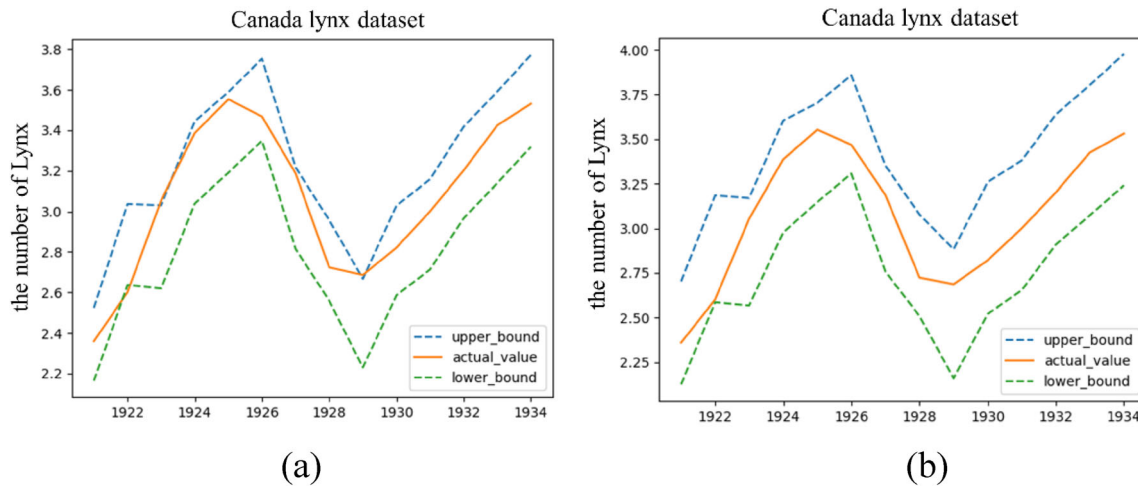


Fig. 6 The output upper bound and lower bound of the residual, Lynx data, **a** $\varepsilon = 0.12$; **b** $\varepsilon = 0.18$

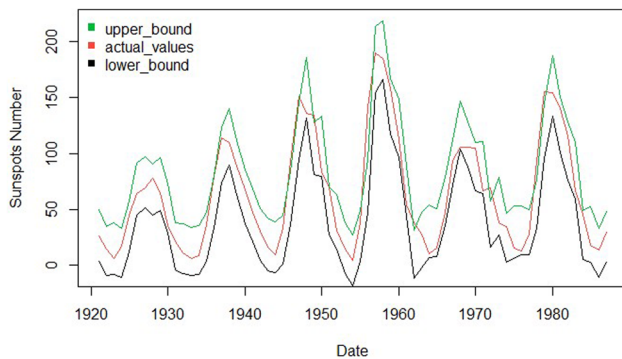


Fig. 7 Prediction results of the proposed model on the Sunspots dataset

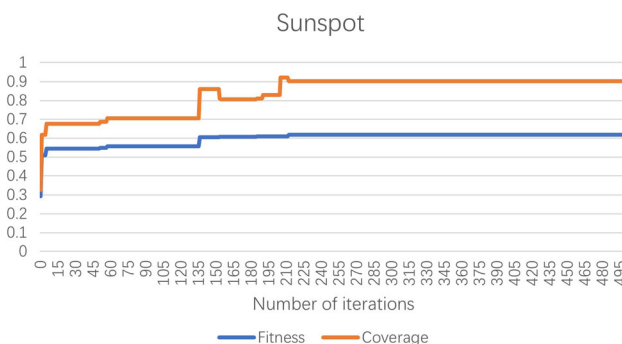


Fig. 8 Trend of fitness and *coverage* versus number of iterations. Note that fitness is the same as Q

Sunspot data set are illustrated in Fig. 7. It can be seen that almost all the data points are covered by the two bounds. This again shows the effectiveness of GNN and our method. In addition, the width of the bounds is flexible and controllable by users according to the complexity of the problem at hand.

5.3 Iterative process for granularity allocation optimization

After training a numerical neural network, it is crucial to granularize the weights and biases. To ensure that the fitness function is valid, we plot Fig. 8 to show how the fitness function and the *coverage* change as the number of iterations increases during the iterative process of the cuckoo search algorithm. It is important to note that there is a decreasing coverage phase during the iterative process. This is because the fitness function has to take into account both the *coverage* and the average width. To ensure that the coverage is as high as possible while the average interval width is small, the selection of an effective optimization tool is critical.

5.4 Comparison with ARIMA prediction intervals

The prediction intervals given by probabilistic prediction models in most studies are generally based on a few common confidence levels fixed in statistics, such as 80% or 95% confidence level. The interval prediction, however,

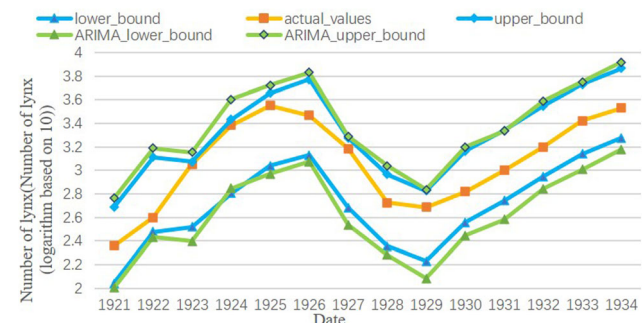


Fig. 9 The predicted intervals of our model and the 95% confidence interval of the ARIMA model (Sunspot data set)

Table 1 Prediction results of ARIMA and our method on three data sets

	80% level		95% level		Ours		
	Coverage	AL	Coverage	AL	Coverage	AL	ε
Sunspots	0.761	36.477	0.896	55.787	0.902	44.076	0.12
Lynx	0.857	0.492	1	0.753	0.986	0.492	0.01
Globtemp	0.704	0.253	0.926	0.387	0.963	0.33	0.12

can be useful in different fields and can serve as an informative opinion for decision makers. Therefore, we show the results for the prediction intervals of the model in this paper and the 95% confidence intervals of the ARIMA model, as shown in Fig. 9 (using Sunspot data set). The model proposed in this paper is clearly better than the probability intervals of the benchmark ARIMA model, being 34.7% narrower in width than ARIMA.

In Fig. 9, the interval of our hybrid model is almost completely covered by the ARIMA prediction interval, which indicates that the hybrid model gives a more accurate range of the future trend of the data. For the same volatility data, the model that gives a narrower interval has a stronger resistance to disturbances.

To further demonstrate better performances of our method over the benchmark probabilistic prediction model, a comparison with the ARIMA model on the Sunspots dataset, Lynx dataset, and Globtemp rate dataset is given in this section. The prediction results at 80% and 95% confidence levels of ARIMA are presented in Table 1.

It is obvious that our proposed model shows a much narrower intervals with similar coverage value or even larger *coverage* on the three time series data sets.

6 Conclusion

The modeling of the residual is crucial to time series prediction problems. A residual comprises the most stochastic factors intra a time series which is closely related to the concept of uncertainty. Therefore, developing an effective tool to deal with a residual with high uncertainty becomes a challenging. In this paper, a hybrid time series forecasting method based on granular neural network and ARIMA is proposed. The main contribution of this method is to develop a more compact prediction model on the residual series to control uncertainty. This hybrid framework has a convenient and adjustable flexible structure for the interval output and outputs a high coverage of the prediction interval. Through the comparison experiments with the classical benchmark probabilistic forecasting model ARIMA, our method shows better performance. In addition, the method provides some aspects for the interval forecasting field. In the future, we will try to explore the multivariate time series prediction using the granular

neural networks and ARIMA. Thus, how to effectively use all the variates and how to aggregate them to form a comprehensive model becomes the core challenge.

Acknowledgements Support from the Fundamental Research Funds for the Central Universities (CUC23ZDTJ001) are gratefully appreciated.

Author contributions MS wrote the main manuscript text. RW did experiments (Figs 1, 2, 5, 6) and prepared the literature. YL prepared Figs. 3, 4, 7, 8, 9. All authors reviewed the manuscript.

Data availability The data sets used in this paper can be found from the following links: Wolfe sunspot data source: <https://www.kaggle.com/datasets/robervalt/sunspots>. Canada lynx data source: https://www.researchgate.net/figure/Canadian-lynx-data-series-1821-1934_fig1_258379422. Globtemp data source: <https://www.poundsterlinglive.com/bank-of-england-spot/historical-spot-exchange-rates/gbp/GBP-to-USD-1993>.

Declarations

Conflict of interest The authors declare no competing interests.

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
- Amirshahi B, Lahmiri S (2023) Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies. *MLWA* 12(15):100465
- Bhardwaj S, Chandrasekhar E, Padiyar P, Gadre VM (2020) A comparative study of wavelet-based ANN and classical techniques for geophysical time-series forecasting. *Comput Geosci* 138:104461
- Bray M, Han D (2004) Identification of support vector machines for runoff modelling. *J Hydroinform* 6(4):265–280
- Busari SI, Samson TK (2022) Modelling and forecasting new cases of Covid-19 in Nigeria: comparison of regression, ARIMA and machine learning models. *Sci Afr* 18:e01404
- Büyüksahin ÜÇ, Ertekin S (2019) Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. *Neurocomputing* 361:151–163
- Chen SM, Hsu CC (2008) A new approach for handling forecasting problems using high-order fuzzy time series. *Intell Autom Soft Comput* 1(14):29–43
- Chen SM, Wang NY (2010) Fuzzy forecasting based on fuzzy-trend logical relationship groups. *IEEE Trans Syst Man Cybern* 4(5):1343–1358
- Chen SM, Manalu GMT, Pan JS, Liu HC (2013) Fuzzy forecasting based on two-factors second-order fuzzy-trend logical

- relationship groups and particle swarm optimization techniques. *IEEE Trans Cybern* 43(3):1102–1117
- 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
- Chimmula VKR, Zhang L (2020) Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fractals* 135:109864
- Dubey AK, Kumar A, García-Díaz V, Sharma AK, Kanhaiya K (2021) Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustain Energy Technol Assess* 47:101474
- Gencer K, Başçiftçi F (2021) Time series forecast modeling of vulnerabilities in the android operating system using ARIMA and deep learning methods. *Sustain Comput Inform* 30:100515
- Guo HY, Pedrycz W, Liu XD (2018) Hidden Markov models based approaches to long-term prediction for granular time series. *IEEE Trans Fuzzy Syst* 26(5):2807–2817
- Han Z, Zhao J, Leung H, Ma KF, Wang W (2019) A review of deep learning models for time series prediction. *IEEE Sens J* 21(6):7833–7848
- Jamei M, Ali M, Karbasi M, Sharma E, Jamei M, Chu X, Yaseen ZM (2023) A high dimensional features-based cascaded forward neural network coupled with MVMD and Boruta-GBDT for multi-step ahead forecasting of surface soil moisture. *Eng Appl Artif Intell* 120:105895
- Kiarie J, Mwalili S, Mbogo R (2022) Forecasting the spread of the COVID-19 pandemic in Kenya using SEIR and ARIMA models. *IDM* 7(2):179–188
- Leite D, Gomide F, Ballini R, Costa P (2011) Fuzzy granular evolving modeling for time series prediction. In: 2011 IEEE international conference on fuzzy systems (FUZZ-IEEE 2011), p 12327771
- Luo J, Gong Y (2023) Air pollutant prediction based on ARIMA-WOA-LSTM model. *Atmos Pollut Res* 14:101761
- Lv SX, Peng L, Hu H, Wang L (2022) Effective machine learning model combination based on selective ensemble strategy for time series forecasting. *Inf Sci* 612:994–1023
- Ning Y, Kazemi H, Tahmasebi P (2022) A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet. *Comput Geosci* 164:105126
- Ong P, Zainuddin Z (2023) An optimized wavelet neural networks using cuckoo search algorithm for function approximation and chaotic time series prediction. *Decis Anal* 6:100188
- Pant M, Kumar S (2022) Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. *Granul Comput* 7:861–879
- Priya RD, Sivaraj R, Anitha N, Devisurya V (2022) Tri-staged feature selection in multi-class heterogeneous datasets using memetic algorithm and cuckoo search optimization. *Expert Syst Appl* 209:118286
- Qiu X, Zhang L, Suganthan PN, Amaratunga GA (2017) Oblique random forest ensemble via least square estimation for time series forecasting. *Inf Sci* 420:249–262
- Rhif M, Abbes AB, Martínez B, Farah IR (2023) Veg-W2TCN: a parallel hybrid forecasting framework for non-stationary time series using wavelet and temporal convolution network model. *Appl Soft Comput* 137:110172
- Salinas D et al (2020) DeepAR: probabilistic forecasting with autoregressive recurrent networks. *Int J Forecast* 36(3):1181–1191
- Song M, Wang Y (2016) A study of granular computing in the agenda of growth of artificial neural networks. *Granul Comput* 1(4):247–257
- Song M, Jing Y, Pedrycz W (2019) Granular neural networks: a study of optimizing allocation of information granularity in input space. *Appl Soft Comput* 77:67–75
- Tomasiello S, Loia V, Khaliq A (2021) A granular recurrent neural network for multiple time series prediction. *Neural Comput Appl* 33:10293–10310
- Vovan T (2023) Building the forecasting model for interval time series based on the fuzzy clustering technique. *Granul Comput* 8:1341–1357
- Wang L, Zeng Y, Chen T (2015) Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Syst Appl* 42(2):855–863
- Wang HZ, Wang GB, Li GQ, Peng JC, Liu YT (2016) Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Appl Energy* 182(15):80–93
- Wang HZ et al (2017) Deep learning based ensemble approach for probabilistic wind power forecasting. *Appl Energy* 188:56–70
- Wang B, Li T, Yan Z, Zhang G, Lu J (2020) DeepPIPE: a distribution-free uncertainty quantification approach for time series forecasting. *Neurocomputing* 397:11–19
- Wani O, Beckers JV, Weerts AH, Solomatine DP (2017) Residual uncertainty estimation using instance-based learning with applications to hydrologic forecasting. *Hydrol Earth Syst Sci* 21(8):4021–4036
- Zeng SZ, Chen SM, Teng MO (2019) Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm. *Inf Sci* 484:350–366
- Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50:159–175
- Zhang YM, Luo L, Yang JC, Liu DH, Kong RX, Feng YB (2019) A hybrid ARIMA-SVR approach for forecasting emergency patient flow. *J Ambient Intell Human Comput* 10(3):3315–3323
- Zhang ML et al (2023) OCRUN: an oppositional Runge Kutta optimizer with cuckoo search for global optimization and feature selection. *Appl Soft Comput* 146:110664

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.