symmetry

MDPI

Article

Enhancing Integer Time Series Model Estimations through Neural Network-Based Fuzzy Time Series Analysis

Mohammed H. El-Menshawy [1], Mohamed S. Eliwa [2,3], Laila A. Al-Essa [4], Mahmoud El-Morshedy [5,6,*] and Rashad M. EL-Sagheer [1,7]

[1] Mathematics Department, Faculty of Science, Al-Azhar University, Naser City, Cairo 11884, Egypt; m.h.elmenshawy@azhar.edu.eg (M.H.E.-M.); rashadmath@azhar.edu.eg (R.M.E.-S.)
[2] Department of Statistics and Operations Research, College of Science, Qassim University, Buraydah 51482, Saudi Arabia; m.eliwa@qu.edu.sa
[3] Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt
[4] Department of Mathematical Sciences, College of Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[5] Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[6] Department of Statistics and Computer Science, Faculty of Science, Mansoura University, Mansoura 35516, Egypt
[7] High Institute of Computer and Management Information System, First Statement, Cairo 11865, Egypt
* Correspondence: m.elmorshedy@psau.edu.sa

Abstract: This investigation explores the effects of applying fuzzy time series (FTSs) based on neural network models for estimating a variety of spectral functions in integer time series models. The focus is particularly on the skew integer autoregressive of order one (NSINAR(1)) model. To support this estimation, a dataset consisting of NSINAR(1) realizations with a sample size of n = 1000 is created. These input values are then subjected to fuzzification via fuzzy logic. The prowess of artificial neural networks in pinpointing fuzzy relationships is harnessed to improve prediction accuracy by generating output values. The study meticulously analyzes the enhancement in smoothing of spectral function estimators for NSINAR(1) by utilizing both input and output values. The effectiveness of the output value estimates is evaluated by comparing them to input value estimates using a mean-squared error (MSE) analysis, which shows how much better the output value estimates perform.

Keywords: neural network; time series; NSINAR(1); spectrum; bispectrum; computer simulation; Daniell lag window; fuzzy inference; statistics and numerical data

check for updates

Citation: El-Menshawy, M.H.; Eliwa, M.S.; Al-Essa, L.A.; El-Morshedy, M.; EL-Sagheer, R.M. Enhancing Integer Time Series Model Estimations through Neural Network-Based Fuzzy Time Series Analysis. *Symmetry* **2024**, *16*, 660. https://doi.org/10.3390/sym16060660

Academic Editors: Jian-Qiang Wang, Hsien-Chung Wu and Sergei D. Odintsov

Received: 17 March 2024
Revised: 26 April 2024
Accepted: 15 May 2024
Published: 27 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, a variety of FTS methodologies have emerged in the literature. FTSs can be defined as employing fuzzy sets to represent its observations. Ref. [1] provided the initial description of FTSs and outlined its fundamental definitions. Fundamentally, an FTS is comprised of three consecutive steps: fuzzification, fuzzy relationship formation, and defuzzification. Numerous studies that have investigated these processes [2–5] have focused solely on the most recent process. Additionally, because the formation of fuzzy links is directly related to predicting, numerous studies have concentrated on it [6–10]. Artificial neural networks appear to be quite good at figuring out fuzzy relationships that increase the forecasting performance's accuracy. Generally, artificial neural network techniques have shown success in a wide range of applications as in [11–13], and researchers often employ neural networks to build fuzzy relationships of the FTS. Over time, there have been notable breakthroughs in the field of modeling integer-valued time series due to the substantial attention given to this topic. The integer autoregressive (INAR) model is one of the best among them for modeling counting series. In recent years, researchers have been striving

to improve the INAR model's capacity to accurately reproduce observed data. This effort has involved modifications to various aspects of INAR models, including adjustments to the marginal distribution (see [14–17]), others the ranking of the models as [18,19], and yet others, the thinning operators as [20–27]. Ref. [28] investigates certain statistical metrics for a number of INAR(1) models. To match specific data and more accurately characterize it, [29] provided an INAR(1) model based on two thinning operators. The fuzzy logic employed in [30] enhanced the estimation of all density functions. Ref. [31] conducted research on the FTS technique to enhance periodograms while keeping their statistical characteristics. Ref. [32] investigates a few statistical properties and all density functions for the ZTPINAR(1) process. Ref. [33] applies the FTS based on the Chen method to smoothing estimates for the DCGINAR(1) process. Ref. [34] used a novel technique of FTSs to improve the estimation of stationary processes' unknown parameters by traditional methods. Ref. [35] employed a fuzzy Markov chain to enhance the estimates of density functions with respect to the MCGINAR(1) process.

Since it focuses on building fuzzy relationships that lead to forecasting, [36] was the first to apply neural networks to FTS predictions. Ref. [37] proposes a novel method for handling high-order multivariate fuzzy time series that is based on artificial neural networks. Ref. [38] develops a fuzzy time series model based on neural networks to enhance the forecasting of observations. Ref. [39] provides a brand-new hybrid fuzzy time series technique that uses artificial neural networks for defuzzification and a fuzzy c-means (FCM) method for fuzzification. Ref. [40] creates and develops precise statistical forecasting models to predict the monthly API and assesses these models to track the state of the air quality. In a unique method, [41] created a PSNN to create fuzzy associations in high-order FTSs and modified PSO to train the network's weights. By combining a convectional neural network and FTS method, [42] suggested short-term load forecasting. Ref. [43] employed various techniques to forecast the air pollution index (API) of Kuala Lumpur, Malaysia, for the year 2017. These techniques included the use of artificial neural networks (ANNs), autoregressive integrated moving average (ARIMA), trigonometric regressors, Box–Cox transformation, ARMA errors, trend and seasonality (TBATS), and multiple fuzzy time series (FTS) models. Ref. [44] built an LSTM recurrent neural network based on trend fuzzy granulation for long-term time series forecasting. A unique multi-functional recurrent fuzzy neural network (MFRFNN) for time series prediction was developed in [45]. Ref. [46] suggested a shallow and deep neural network model for demand forecasting fuzzy time series pharmaceutical data.

The purpose of this research is to present fuzzy time series (FTSs), which are based on neural network models [36], and is intended to enhance estimates of density functions for integer time series. These functions are spectral, bispectral, and normalized bispectral density functions. The NSINAR(1) model is used in this strategy. All spectrum functions and their smoothed estimations for NSINAR(1) are computed for this purpose. We use neural network-based FTSs and forecast realizations to generate the output values for NSINAR(1) observational "input values". All density functions are estimated with input and output values. The contribution of the output values of neural network-based FTSs to smoothing of these estimations is investigated by contrasting the two cases using the results.

2. Developing the Forecasting Models

The following are the procedures for establishing a forecasting model: data preparation, evaluation, and selection; neural network construction (in terms of input variable selection, transfer function, structure, etc. [47].

2.1. Preparing Data

A number of crucial decisions, including data preparation, input variable selection, network type and design, the transfer function, training methodology, and model validation, assessment, and selection, must be made by the neural network forecaster. Some of these

choices can be made while building the model, but others must be carefully thought through before beginning any modeling work. Given that neural networks operate on data-driven principles, data preparation emerges as a pivotal initial step in crafting an effective neural network model. Indeed, the creation of a practical and predictive neural network model heavily relies on the availability of a robust, sufficient, and representative dataset. Consequently, the quality of the data significantly influences the reliability of neural network models. Moreover, ample data are indispensable for the training process of neural networks. Numerous studies have utilized a practical ratio ranging from 70%:30% to 90%:10% for distinguish between in-samples and out-of-samples [47]. As a result, we use NSINAR(1) to construct a series of size n=1000. For our estimation (the in-sample), we chose the data from the first observation to the 800th, and for our forecast (the out-of-sample), we used the data from the 801st and 1000th. In other words, the ratio is around $\frac{800}{1000} : \frac{200}{1000} = 80\% : 20\%$ falling among the two categories of samples.

2.2. Setup of a Neural Network

The model used in forecasting the most is a multilayer feedforward structure [48]. As a result, backpropagation [49] was selected as the model, and *PC Neuron Institutional*, a backpropagation software package, was selected as the method for creating the forecasting model. In the setup, our goal was to first build (or train) the fuzzy associations between each FLR before forecasting. An FLR is a 1-1 relationship. Hence, there is one input layer and one output layer with one node each. The majority of forecasting applications have employed only one hidden layer and a suitable number of hidden nodes, even if there are several criteria for selecting the number of hidden layers and hidden nodes in each layer [50–52]. A minimal neural network was used [53] to avoid over-fitting. As a result, we employed two hidden nodes and one hidden layer. As a result, we created the neural network structure shown in Figure 1. Figure 2 shows the purpose of each node in the hidden and output layers. The previous layer's node(t) r, like X_r in the diagram, provides input(t) to the node s. A weight, Wrs, which represents the strength of the connection between each connection from node r to s, is attached to each one. The following formula is used to calculate node s'output, Y_t [30].

$$Y_t = f(\sum W_{rs} \times X_r - \theta_r), \tag{1}$$

$$f(z) = \frac{1}{1 + exp(-z)}, \tag{2}$$

where f(z) is a sigmoid function.

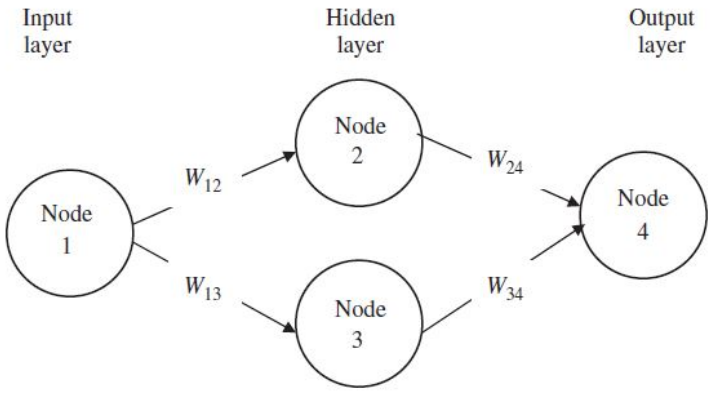


Figure 1. Neural network structure.

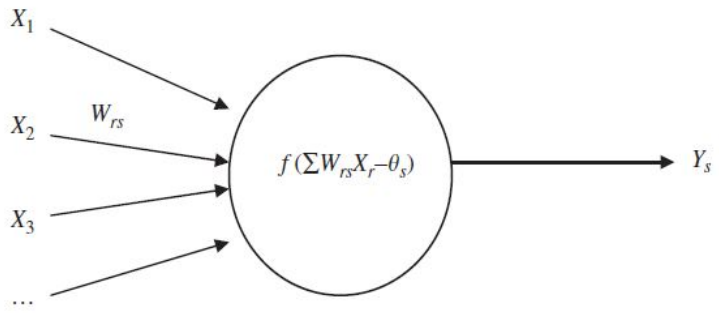


Figure 2. A node in a neural network.

2.3. Model Selection and Evaluation

A pertinent study [54] states that no single approach has proven to be consistently best for all kinds of applications, including neural network applications. Thus, we used a simple model, as well as a hybrid one. An in-sample and an out-of-sample category were created for the observations. A further division into known and unknown patterns was made for the out-of-sample observations. In contrast to the unidentified patterns that constituted the majority of the data outside of the sample, the observations that were present in both outside of the sample and inside the sample were regarded as established patterns. With regard to their respective original functions, we used the mean-squared error (MSE) to compare the performance of the three estimators in each of the three scenarios. For details on the calculation of the MSE, please refer to [32,33], which results in a different evaluation of the three different types of observations.

3. Fuzzy Time Series Models Utilizing Neural Networks

This study applied a backpropagation neural network to forecast the fuzzy observations with respect to the FTS of NSINAR(1). There are six steps in this study, which can be reported as follows:

1. Defining and partitioning the universe of discourse: According to the problem domain of NSINAR(1)' series, the universe of discourse for observations, U =[starting, ending], is defined. After the length of the intervals, l, is determined, U can be partitioned into equal-length intervals $u_1, u_2, u_3, \ldots, u_b, b = 1, \ldots$ and their corresponding midpoints $m_1, m_2, m_3, \ldots, m_b$: respectively.

$$u_b = [starting + (b-1) \times l_a, starting + b \times l_a], \quad (3)$$

$$m_b = \frac{1}{2} \times [starting + (b-1) \times l_a, starting + b \times l_a]. \quad (4)$$

2. Defining fuzzy sets for observations: Each linguistic observation, A_i, can be defined by the intervals $u_1, u_2, u_3, \ldots, u_b$, where $A_i = f_{Ai}(u_1)/u_1 + f_{Ai}(u_2)/u_2 + \ldots + f_{Ai}(u_b)/u_b$.
3. Fuzzifying the observations: A fuzzy set is created by fuzzifying each observation. An observation is fuzzified to A_i if its greatest degree of membership is in A_i, just as in [6,55].
4. Establishing the fuzzy relationship (neural network training): The fuzzy associations in these **FLR**s were built (or trained) using a backpropagation neural network. Index i served as the input, and index j served as the appropriate output for each FLR, $A_i \rightarrow A_j$. These **FLR**s became the input and output patterns for neural network training.
5. Forecasting: A description of the hybrid and basic models can be found below. The basic model uses a neural network methodology to forecast each piece of data, while the hybrid model uses the same neural network approach to forecast the known patterns together with a simple strategy to anticipate the unknown patterns.

Model 1 (basic mode): Assume $F(t-1) = A_{i'}$. We chose i' as the forecast input in order to make the calculations easier. Assume that j' is the neural network's output. The hazy forecast is $A_{j'}$, we say. In other words,

$$F(t) = A_{j'}. \tag{5}$$

Model 2 (hybrid mode): Assume $F(t-1) = A_{i'}$. If $A_{i'}$ is a recognized pattern, the basic model is used to obtain the fuzzy forecast. If $A_{i'}$ is an unknown pattern, then we merely take Ai as the fuzzily predicted value for $F(t)$, in accordance with Chen's model [6]. That is,

$$F(t) = A_{i'}. \tag{6}$$

6. Defuzzifying: No matter whatever model is used, the defuzzified forecast is always equal to the fuzzy forecast's midpoint. Assume $A_{k'}$ is the fuzzy prediction for F(t). The forecast that has been defuzzed corresponds to $A_{k'}$'s middle, i.e.,

$$forecast_t = m_{k'}. \tag{7}$$

For additional details on this methodology, see [36]. Therefore, the forecast observations that were obtained from the generated realizations from the NSINAR(1) process—known as the "input values"—are the "output values" of this approach.

4. The New Skew INAR(1) Model

The NSINAR(1) process was first defined by [56]. These model, also known as true integer autoregressive models, have become necessary for the modeling of count data with positive and negative values. It defines the NSINAR(1) as

$$X_t = \beta \star X_{t-1} + \eta_t \tag{8}$$

where $\beta \star X_{t-1} \stackrel{d}{=} \beta * A_{t-1} - \beta \circ B_{t-1}$ is the difference between the negative binomial thinning and binomial thinning operators, where the counting series $\beta * A_{t-1}$ and $\beta \circ B_{t-1}$ are independent random variables. $\beta * A = \sum_{i=0}^{A} W_i$ and $\beta \circ B = \sum_{i=0}^{B} U_i$, where W_i is a sequence of *i.i.d* geometric random variables and U_i is a sequence of *i.i.d*. Bernoulli random variables independent of W_i, $X_t = A_t - B_t$, where $A_t \sim$ geometric$(\frac{\gamma}{1+\gamma})$, $B_t \sim$ Poisson(λ), $\{X_t\}$ is a sequence of random variables having a geometric-Poisson (γ, λ), and η_t has the distribution of $\epsilon_t - \varepsilon_t$, where $\{\epsilon_t\}$ and $\{\varepsilon_t\}$ are independent r.v.s and X_t and η_{t-l} are independent for all $l \geq 1$. ε_t are i.i.d. r.v.s with a common Poisson$(\lambda(1-\beta))$ distribution, and ϵ_t is a mixture of two random variables with geometric $\gamma/(1+\gamma)$ and geometric$(\beta/(1+\beta))$ distributions. The condition of the stationarity of the process $\{X_t\}$ is $0 \leq \beta \leq \gamma/(1+\gamma)$, and the condition of the non-stationarity of the process $\{X_t\}$ is $\gamma/(1+\gamma) \leq \beta \leq 1$. Here, our study is restricted to the stationary case. Some properties of X_t are introduced:

The mean $\gamma_X = \gamma - \lambda$, the variance $\sigma_X^2 = \gamma(\gamma+1) + \lambda$, the second-order central moment $R_2(s) = \beta^s(\gamma^2 + \gamma + \lambda) = \beta^s R_2(0)$, and the third-order central moment are calculated as $R_3(0,0) = 2\gamma^3 + 3\gamma^2 + \gamma - \lambda$, $R_3(0,s) = \beta^s(2\gamma^3 + 3\gamma^2 + \gamma - \lambda) = \beta^s R_3(0,0)$, and $R_3(s,s) = \beta^{2s} R_3(0,0) - \beta R_2(0)\frac{\beta^s - \beta^{2s}}{(1-\beta)}$. Refer to [23] and [56] for detailed information on the thinning operator $\star$ and the NSINAR(1) model, respectively.

5. Spectral and Bispectral Density Functions

In time series analysis, statistical spectral analysis plays a number of roles, including the estimate, hypothesis testing, data reduction, and description. It is quite helpful to examine a time series in both the time and frequency domains when performing an analysis. The Fourier transform, which converts the time series from the time domain to the frequency domain, is the fundamental tool of spectral analysis. This field frequently contains a wealth of important information that requires additional research to locate. Second-order spectra

have played a very important role in the analysis of linear and Gaussian time series data and in signal processing. When the process is Gaussian, the second-order spectra contain all the necessary and useful information about the process, and we do not need to consider higher order spectra. Before analyzing the time series data, in most cases, we assume that the series is generated from a linear process and even from a Gaussian process for further simplified analysis. But, in reality, this is not the case for every process. The reason behind assuming the linearity of the process is that it is very easy to fit a linear model in comparison to fitting the actual non-linear model. In order to study non-linear and non-Gaussian processes, one needs to consider higher order spectra. Theoretically, it is possible to compute the spectrum of any order, but computationally, it is very costly. Hence, we will consider only the bispectrum, the simplest higher order spectrum. So, we can say that the second-order spectra will not adequately characterize the series (unless it is Gaussian), and hence, there is a need for higher order spectral analysis. The simplest type of higher order spectral analysis is the bispectral analysis. This leads us to the bispectral density function, which gives us important details regarding the process' nonlinearity. The normalized bispectrum's modulus for continuous non-Gaussian time series is flat.

The spectral density function "SDF", bispectral density function "BDF", and normalized bispectral density function "NBDF" of NSINAR(1) are provided in this section.

Theorem 1. *If (8) is satisfied by* $\{X_t\}$*, then the SDF, denoted by* $g(w)$*, is computed as Let* $\{X_t\}$ *satisfy (8), then the SDF, represented by* $g(w)$ *is calculated as*

$$g(w) = \frac{(1-\beta^2)(\gamma^2+\gamma+\lambda)}{2\pi(1+\beta^2-2\beta\, cos\, w)}, \quad -\pi \le w \le \pi. \tag{9}$$

Proof.

$$\begin{aligned} g(w) &= \frac{1}{2\pi}\sum_{t=-\infty}^{\infty} R_2(t)exp(-iwt) = \frac{1}{2\pi}[R_2(0) + \sum_{t=1}^{\infty}\beta^t R_2(0)exp(-iwt) + \sum_{t=1}^{\infty}\beta^t R_2(0)exp(iwt)] \\ &= \frac{R_2(0)}{2\pi}[1 + \frac{\beta exp(-iw)}{1-\beta exp(-iw)} + \frac{\beta exp(iw)}{1-\beta exp(iw)}] = \frac{R_2(0)[1-\beta^2]}{2\pi(1+\beta^2-2\beta cosw)}. \end{aligned}$$

□

Theorem 2. *If (8) is satisfied by* $\{X_t\}$*, then the BDF, denoted by* $g(w_1, w_2)$*, are computed as*

$$\begin{aligned} g(w_1,w_2) =& \frac{1}{(2\pi)^2}\Bigg[R_3(0,0)\{1 + F_1(-w_1) + F_1(-w_2) + F_1(w_1+w_2) + \left(R_3(0,0) + \frac{\beta^2 R_2(0)}{\beta(1-\beta)}\right)\{F_2(w_1) \\ &+ F_2(w_2) + F_2(-w_1-w_2)\} - \left(\frac{\beta^2 R_2(0)}{\beta(1-\beta)}\right)\{F_1(w_1) + F_1(w_2) + F_1(-w_1-w_2)\} \\ &+ F_1(-w_1)\{[F_2(-w_1-w_2) + F_2(w_2)](R_3(0,0) + \frac{\beta^2 R_2(0)}{\beta(1-\beta)}) - [F_1(-w_1-w_2) \\ &+ F_1(w_2)]\frac{\beta^2 R_2(0)}{\beta(1-\beta)}\} + F_1(-w_2)\{[F_2(-w_1-w_2) + F_2(w_1)](R_3(0,0) + \frac{\beta^2 R_2(0)}{\beta(1-\beta)}) \\ &- [F_1(-w_1-w_2) + F_1(w_1)]\frac{\beta^2 R_2(0)}{\beta(1-\beta)}\} + F_1(w_1+w_2)\{[F_2(w_1) + F_2(w_2)](R_3(0,0) \\ &+ \frac{\beta^2 R_2(0)}{\beta(1-\beta)}) - [F_1(w_1) + F_1(w_2)]\frac{\beta^2 R_2(0)}{\beta(1-\beta)}\}\Bigg], \quad -\pi \le w_1, w_2 \le \pi, \end{aligned} \tag{10}$$

$$\text{where } F_1(w_k) = \tfrac{\beta exp(iw_k)}{1-\beta exp(iw_k)} \text{ and } F_2(w_k) = \tfrac{\beta^2 exp(iw_k)}{1-\beta^2 exp(iw_k)}, k = 1, 2.$$

Proof. We can write $g(w_1, w_2)$ as (see [17]) in the following formulae:

$$g(w_1,w_2) = \frac{1}{(2\pi)^2}[\sum_{t_1=0}^{\infty}\sum_{t_2=t_1}^{\infty} R_3(t_1,t_2)exp(-i(t_1w_1+t_2w_2)) + \sum_{t_2=0}^{\infty}\sum_{t_1=t_2+1}^{\infty} R_3(t_2,t_1)exp(-i(t_1w_1+t_2w_2))$$
$$+ \sum_{t_1=0}^{\infty}\sum_{t_2=-\infty}^{-1} R_3(-t_2,t_1-t_2)exp(-i(t_1w_1+t_2w_2)) + \sum_{t_1=-\infty}^{-1}\sum_{t_2=-\infty}^{t_1-1} R_3(t_1-t_2,-t_2)exp(-i(t_1w_1+t_2w_2))$$
$$+ \sum_{t_2=-\infty}^{-1}\sum_{t_1=-\infty}^{-t_2} R_3(t_2-t_1,-t_1)exp(-i(t_1w_1+t_2w_2)) + \sum_{t_1=-\infty}^{-1}\sum_{t_2=0}^{\infty} R_3(-t_1,t_2-t_1)exp(-i(t_1w_1+t_2w_2))].$$

Applying the symmetry characteristics of the third-order cumulants (see [57]), then

$$g(w_1,w_2) = \frac{1}{(2\pi)^2}[R_3(0,0) + \sum_{\tau=1}^{\infty} R_3(0,\tau)\{exp(-i\tau w_1) + exp(-i\tau w_2) + exp(i\tau(w_1+w_2))\}$$
$$+ \sum_{\tau=1}^{\infty} R_3(\tau,\tau)\{exp(i\tau w_1) + exp(i\tau w_2) + exp(-i\tau(w_1+w_2))\}$$
$$+ \sum_{t=1}^{\infty}\sum_{\tau=1}^{\infty} R_3(t,t+\tau)\{exp(-itw_1 - i(t+\tau)w_2) + exp(-itw_2 - i(t+\tau)w_1)$$
$$+ exp(itw_1 - i\tau w_2)exp(itw_2 - i\tau w_1) + exp(i\tau w_1 + i(t+\tau)w_2) + exp(i\tau w_2 + i(t+\tau)w_1)\}].$$

Using the expressions of $R_3(0,\tau), R_3(\tau,\tau)$ and $R_3(s,s+\tau)$, we obtain

$$g(w_1,w_2) = \frac{1}{(2\pi)^2}[R_3(0,0) + \sum_{\tau=1}^{\infty} \beta^{\tau} R_3(0,0)\{exp(-i\tau w_1) + exp(-i\tau w_2) + exp(i\tau(w_1+w_2))\}$$
$$+ \sum_{\tau=1}^{\infty}(\beta^{2\tau}R_3(0,0) - \beta^2 R_2(0)\frac{\beta^{\tau}-\beta^{2\tau}}{\beta(1-\beta)})\{exp(i\tau w_1) + exp(i\tau w_2) + exp(-i\tau(w_1+w_2))\}$$
$$+ \sum_{t=1}^{\infty}\sum_{\tau=1}^{\infty}\beta^{\tau}[\beta^{2t}R_3(0,0) - \beta^2 R_2(0)\frac{\beta^t-\beta^{2t}}{\beta(1-\beta)}]\{exp(-itw_1 - i(t+\tau)w_2) + exp(-itw_2 - i(t+\tau)w_1)$$
$$+ exp(itw_1 - i\tau w_2)exp(itw_2 - i\tau w_1) + exp(i\tau w_1 + i(t+\tau)w_2) + exp(i\tau w_2 + i(t+\tau)w_1)\}].$$

All these summations can be evaluated as follows, for example

$$\sum_{\tau=1}^{\infty}(\beta^{\tau}exp(-i\tau w_1)) = \sum_{\tau=1}^{\infty}(\beta exp(-iw_1))^{\tau} = \frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)} = F_1(-w_1),$$

Hence, following a few computations and summations, we have

$$
\begin{aligned}
g(w_1,w_2) = \frac{1}{(2\pi)^2}[R_3(0,0) + R_3(0,0)\{&\frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)} + \frac{\beta exp(-iw_2)}{1-\beta exp(-iw_2)} + \frac{\beta exp(i(w_1+w_2))}{1-\beta exp(i(w_1+w_2))}\}\\
+ (R_3(0,0) + \frac{\beta^2 R_2(0)}{\beta-\beta^2})\{&\frac{\beta^2 exp(iw_1)}{1-\beta^2 exp(iw_1)} + \frac{\beta^2 exp(iw_2)}{1-\beta^2 exp(iw_2)} + \frac{\beta^2 exp(-i(w_1+w_2))}{1-\beta^2 exp(-i(w_1+w_2))}\}\\
- \frac{\beta^2 R_2(0)}{\beta-\beta^2}\{&\frac{\beta exp(iw_1)}{1-\beta exp(iw_1)} + \frac{\beta exp(iw_2)}{1-\beta exp(iw_2)} + \frac{\beta exp(-i(w_1+w_2))}{1-\beta exp(-i(w_1+w_2))}\} + (R_3(0,0)\\
+ \frac{\beta^2 R_2(0)}{\beta-\beta^2}\{&\frac{\beta^2 exp(-i(w_1+w_2))}{1-\beta^2 exp(-i(w_1+w_2))}\frac{\beta exp(-iw_2)}{1-\beta exp(-iw_2)} + \frac{\beta^2 exp(-i(w_1+w_2))}{1-\beta^2 exp(-i(w_1+w_2))}\frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)}\\
+ &\frac{\beta^2 exp(iw_1)}{1-\beta^2 exp(iw_1)}\frac{\beta exp(-iw_2)}{1-\beta exp(-iw_2)} + \frac{\beta^2 exp(iw_2)}{1-\beta^2 exp(iw_2)}\frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)}\\
+ &\frac{\beta^2 exp(iw_1)}{1-\beta^2 exp(iw_1)}\frac{\beta exp(i(w_1+w_2))}{1-\beta exp(i(w_1+w_2))} + \frac{\beta^2 exp(iw_2)}{1-\beta^2 exp(iw_2)}\frac{\beta exp(i(w_1+w_2))}{1-\beta exp(i(w_1+w_2))}\}\\
+ \frac{\beta^2 R_2(0)}{\beta-\beta^2}\{&\frac{\beta exp(-i(w_1+w_2))}{1-\beta^{-i(w_1+w_2)}}\frac{\beta exp(-iw_2)}{1-\beta exp(-iw_2)} + \frac{\beta exp(-i(w_1+w_2))}{1-\beta exp(-i(w_1+w_2))}\frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)}\\
+ &\frac{\beta exp(iw_1)}{1-\beta exp(iw_1)}\frac{\beta exp(-iw_2)}{1-\beta exp(-iw_2)} + \frac{\beta exp(iw_2)}{1-\beta exp(iw_2)}\frac{\beta exp(-iw_1)}{1-\beta exp(-iw_1)}\\
+ &\frac{\beta exp(iw_1)}{1-\beta exp(iw_1)}\frac{\beta exp(i(w_1+w_2))}{1-\beta exp(i(w_1+w_2))} + \frac{\beta exp(iw_2)}{1-\beta exp(iw_2)}\frac{\beta exp(i(w_1+w_2))}{1-\beta exp(i(w_1+w_2))}\}],
\end{aligned}
$$

and by taking, $F_1(w_k) = \frac{\beta exp(iw_k)}{1-\beta exp(iw_k)}$ and $F_2(w_k) = \frac{\beta^2 exp(iw_k)}{1-(\beta^2 exp(iw_k)}, k = 1, 2.$, the proof is complete. The NBDF, represented by $f(w_1, w_2)$, is calculated as

$$f(w_1,w_2) = \frac{g(w_1,w_2)}{\sqrt{g(w_1)g(w_2)g(w_1+w_2)}}, \tag{11}$$

where $g(w_1, w_2)$ and $g(w)$ are obtained by (11) and (9), respectively. □

Figure 3 illustrates the generated observations from the NSINAR(1) process at $\lambda = 3$, $\beta = 0.25$, and $\gamma = 5$. Figures 4, 5, and 6 show, respectively, the SDF, BDF, and NBDF of NSINAR(1) at $\lambda = 3, \beta = 0.25$, and $\gamma = 5$. We infer that the model is linear from Figures 5 and 6 and the results in Tables 1 and 2, since the NBDF values, which fall between (0.73, 0.78), are flatter (constant, very tightly spaced apart) than the BDF values, which fall between (5, 20). The simulated series of the forecasted NSINAR(1) observations of the neural network-based FTS in both the basic model and the hybrid model are shown in Figure 7. Figure 7 shows that the shape and properties of the time series are maintained by the forecasted observations. From Figures 3 and 7, one can deduce that the series is stationary and its values are positive- and negative-valued, and this agrees with the definition of the NSINAR(1) model.

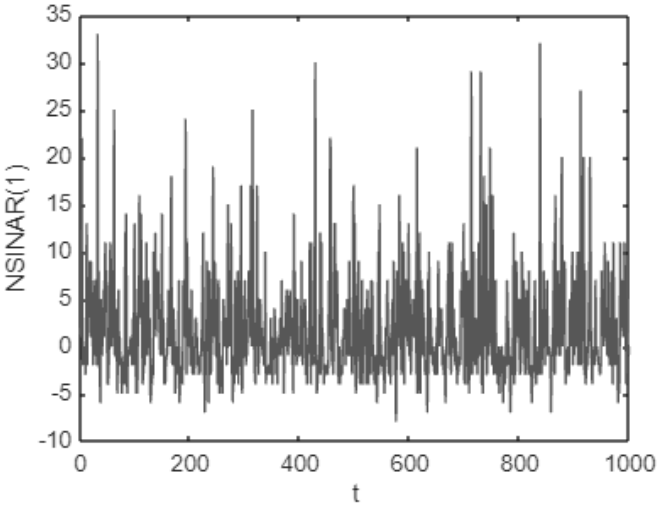


Figure 3. NSINAR(1) model simulations using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.

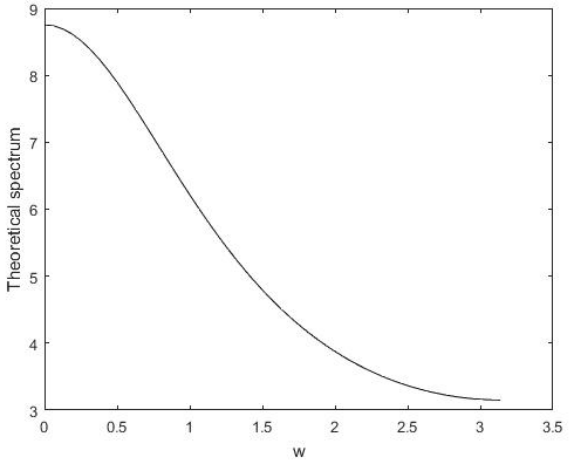


Figure 4. SDF of NSINAR(1) using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.

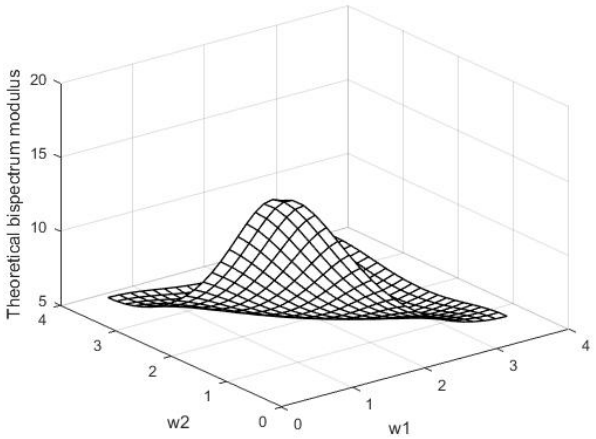


Figure 5. BDF of NSINAR(1) using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.

Table 1. The BDF of NSINAR(1) using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.

	w_2	**0.00** π	**0.05** π	**0.10** π	**0.15** π	**0.20** π	**0.25** π	**0.30** π	**0.35** π	**0.40** π	**0.45** π	**0.50** π	**0.55** π	**0.60** π	**0.65** π	**0.70** π	**0.75** π	**0.80** π	**0.85** π	**0.90** π	**0.95** π	π
w_1																						
0.00 π		18.956	18.759	18.196	17.344	16.301	15.167	14.024	12.932	11.925	11.022	10.228	9.541	8.954	8.459	8.049	7.715	7.451	7.252	7.112	7.030	7.002
0.05 π		18.759	18.379	17.672	16.726	15.642	14.508	13.397	12.354	11.405	10.562	9.827	9.194	8.657	8.208	7.839	7.542	7.312	7.144	7.034	6.979	6.979
0.10 π		18.196	17.672	16.874	15.890	14.814	13.721	12.671	11.697	10.821	10.047	9.376	8.802	8.317	7.915	7.587	7.328	7.132	6.995	6.914	6.887	6.914
0.15 π		17.344	16.726	15.890	14.918	13.888	12.864	11.893	11.001	10.203	9.502	8.897	8.382	7.950	7.594	7.308	7.085	6.923	6.816	6.763	6.763	6.816
0.20 π		16.301	15.642	14.814	13.888	12.929	11.990	11.106	10.300	9.582	8.954	8.414	7.956	7.574	7.263	7.016	6.829	6.698	6.621	6.595	6.621	6.698
0.25 π		15.167	14.508	13.721	12.864	11.990	11.141	10.347	9.625	8.984	8.425	7.946	7.542	7.207	6.938	6.728	6.574	6.473	6.423	6.423	6.473	6.574
0.30 π		14.024	13.397	12.671	11.893	11.106	10.347	9.638	8.996	8.427	7.932	7.509	7.154	6.863	6.632	6.457	6.334	6.261	6.237	6.261	6.334	6.457
0.35 π		12.932	12.354	11.697	11.001	10.300	9.625	8.996	8.426	7.922	7.485	7.113	6.803	6.552	6.356	6.212	6.118	6.071	6.071	6.118	6.212	6.356
0.40 π		11.925	11.405	10.821	10.203	9.582	8.984	8.427	7.922	7.476	7.091	6.764	6.495	6.280	6.116	6.001	5.932	5.909	5.932	6.001	6.116	6.280
0.45 π		11.022	10.562	10.047	9.502	8.954	8.425	7.932	7.485	7.091	6.751	6.465	6.232	6.049	5.914	5.825	5.781	5.781	5.825	5.914	6.049	6.232
0.50 π		10.228	9.827	9.376	8.897	8.414	7.946	7.509	7.113	6.764	6.465	6.216	6.015	5.861	5.753	5.688	5.667	5.688	5.753	5.861	6.015	6.216
0.55 π		9.541	9.194	8.802	8.382	7.956	7.542	7.154	6.803	6.495	6.232	6.015	5.843	5.716	5.632	5.590	5.590	5.632	5.716	5.843	6.015	6.232
0.60 π		8.954	8.657	8.317	7.950	7.574	7.207	6.863	6.552	6.280	6.049	5.861	5.716	5.613	5.551	5.531	5.551	5.613	5.716	5.861	6.049	6.280
0.65 π		8.459	8.208	7.915	7.594	7.263	6.938	6.632	6.356	6.116	5.914	5.753	5.632	5.551	5.511	5.511	5.551	5.632	5.753	5.914	6.116	6.356
0.70 π		8.049	7.839	7.587	7.308	7.016	6.728	6.457	6.212	6.001	5.825	5.688	5.590	5.531	5.511	5.531	5.590	5.688	5.825	6.001	6.212	6.457
0.75 π		7.715	7.542	7.328	7.085	6.829	6.574	6.334	6.118	5.932	5.781	5.667	5.590	5.551	5.551	5.590	5.667	5.781	5.932	6.118	6.334	6.574
0.80 π		7.451	7.312	7.132	6.923	6.698	6.473	6.261	6.071	5.909	5.781	5.688	5.632	5.613	5.632	5.688	5.781	5.909	6.071	6.261	6.473	6.698
0.85 π		7.252	7.144	6.995	6.816	6.621	6.423	6.237	6.071	5.932	5.825	5.753	5.716	5.716	5.753	5.825	5.932	6.071	6.237	6.423	6.621	6.816
0.90 π		7.112	7.034	6.914	6.763	6.595	6.423	6.261	6.118	6.001	5.914	5.861	5.843	5.861	5.914	6.001	6.118	6.261	6.423	6.595	6.763	6.914
0.95 π		7.030	6.979	6.887	6.763	6.621	6.473	6.334	6.212	6.116	6.049	6.015	6.015	6.049	6.116	6.212	6.334	6.473	6.621	6.763	6.887	6.979
π		7.002	6.979	6.914	6.816	6.698	6.574	6.457	6.356	6.280	6.232	6.216	6.232	6.280	6.356	6.457	6.574	6.698	6.816	6.914	6.979	7.002

Table 2. The NBDF of NSINAR(1) using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.

	w_2	**0.00** π	**0.05** π	**0.10** π	**0.15** π	**0.20** π	**0.25** π	**0.30** π	**0.35** π	**0.40** π	**0.45** π	**0.50** π	**0.55** π	**0.60** π	**0.65** π	**0.70** π	**0.75** π	**0.80** π	**0.85** π	**0.90** π	**0.95** π	π
w_1																						
0.00 π		0.7319	0.7322	0.7332	0.7346	0.7363	0.7381	0.7399	0.7417	0.7433	0.7447	0.7460	0.7471	0.7480	0.7488	0.7494	0.7499	0.7503	0.7507	0.7509	0.7510	0.7510
0.05 π		0.7322	0.7329	0.7340	0.7356	0.7373	0.7392	0.7410	0.7426	0.7442	0.7455	0.7467	0.7477	0.7485	0.7492	0.7498	0.7503	0.7507	0.7509	0.7511	0.7512	0.7512
0.10 π		0.7332	0.7340	0.7353	0.7369	0.7387	0.7405	0.7422	0.7438	0.7453	0.7465	0.7476	0.7485	0.7493	0.7500	0.7505	0.7509	0.7512	0.7515	0.7516	0.7516	0.7516
0.15 π		0.7346	0.7356	0.7369	0.7386	0.7403	0.7420	0.7437	0.7452	0.7465	0.7477	0.7487	0.7496	0.7503	0.7509	0.7514	0.7517	0.7520	0.7522	0.7523	0.7523	0.7522
0.20 π		0.7363	0.7373	0.7387	0.7403	0.7419	0.7436	0.7451	0.7466	0.7478	0.7489	0.7499	0.7507	0.7514	0.7519	0.7523	0.7527	0.7529	0.7530	0.7531	0.7530	0.7529
0.25 π		0.7381	0.7392	0.7405	0.7420	0.7436	0.7451	0.7466	0.7479	0.7491	0.7502	0.7511	0.7518	0.7524	0.7529	0.7533	0.7536	0.7538	0.7539	0.7539	0.7538	0.7536
0.30 π		0.7399	0.7410	0.7422	0.7437	0.7451	0.7466	0.7480	0.7493	0.7504	0.7514	0.7522	0.7529	0.7535	0.7539	0.7543	0.7545	0.7546	0.7547	0.7546	0.7545	0.7543
0.35 π		0.7417	0.7426	0.7438	0.7452	0.7466	0.7479	0.7493	0.7505	0.7515	0.7525	0.7532	0.7539	0.7544	0.7548	0.7551	0.7553	0.7554	0.7554	0.7553	0.7551	0.7548
0.40 π		0.7433	0.7442	0.7453	0.7465	0.7478	0.7491	0.7504	0.7515	0.7525	0.7534	0.7542	0.7548	0.7552	0.7556	0.7559	0.7560	0.7561	0.7560	0.7559	0.7556	0.7552
0.45 π		0.7447	0.7455	0.7465	0.7477	0.7489	0.7502	0.7514	0.7525	0.7534	0.7542	0.7549	0.7555	0.7559	0.7563	0.7565	0.7566	0.7566	0.7565	0.7563	0.7559	0.7555
0.50 π		0.7460	0.7467	0.7476	0.7487	0.7499	0.7511	0.7522	0.7532	0.7542	0.7549	0.7556	0.7561	0.7565	0.7568	0.7570	0.7570	0.7570	0.7568	0.7565	0.7561	0.7556
0.55 π		0.7471	0.7477	0.7485	0.7496	0.7507	0.7518	0.7529	0.7539	0.7548	0.7555	0.7561	0.7566	0.7569	0.7572	0.7573	0.7573	0.7572	0.7569	0.7566	0.7561	0.7555
0.60 π		0.7480	0.7485	0.7493	0.7503	0.7514	0.7524	0.7535	0.7544	0.7552	0.7559	0.7565	0.7569	0.7573	0.7574	0.7575	0.7574	0.7573	0.7569	0.7565	0.7559	0.7552
0.65 π		0.7488	0.7492	0.7500	0.7509	0.7519	0.7529	0.7539	0.7548	0.7556	0.7563	0.7568	0.7572	0.7574	0.7576	0.7576	0.7574	0.7572	0.7568	0.7563	0.7556	0.7548
0.70 π		0.7494	0.7498	0.7505	0.7514	0.7523	0.7533	0.7543	0.7551	0.7559	0.7565	0.7570	0.7573	0.7575	0.7576	0.7575	0.7573	0.7570	0.7565	0.7559	0.7551	0.7543
0.75 π		0.7499	0.7503	0.7509	0.7517	0.7527	0.7536	0.7545	0.7553	0.7560	0.7566	0.7570	0.7573	0.7574	0.7574	0.7573	0.7570	0.7566	0.7560	0.7553	0.7545	0.7536
0.80 π		0.7503	0.7507	0.7512	0.7520	0.7529	0.7538	0.7546	0.7554	0.7561	0.7566	0.7570	0.7572	0.7573	0.7572	0.7570	0.7566	0.7561	0.7554	0.7546	0.7538	0.7529
0.85 π		0.7507	0.7509	0.7515	0.7522	0.7530	0.7539	0.7547	0.7554	0.7560	0.7565	0.7568	0.7569	0.7569	0.7568	0.7565	0.7560	0.7554	0.7547	0.7539	0.7530	0.7522
0.90 π		0.7509	0.7511	0.7516	0.7523	0.7531	0.7539	0.7546	0.7553	0.7559	0.7563	0.7565	0.7566	0.7565	0.7563	0.7559	0.7553	0.7546	0.7539	0.7531	0.7523	0.7516
0.95 π		0.7510	0.7512	0.7516	0.7523	0.7530	0.7538	0.7545	0.7551	0.7556	0.7559	0.7561	0.7561	0.7559	0.7556	0.7551	0.7545	0.7538	0.7530	0.7523	0.7516	0.7512
π		0.7510	0.7512	0.7516	0.7522	0.7529	0.7536	0.7543	0.7548	0.7552	0.7555	0.7556	0.7555	0.7552	0.7548	0.7543	0.7536	0.7529	0.7522	0.7516	0.7512	0.7510

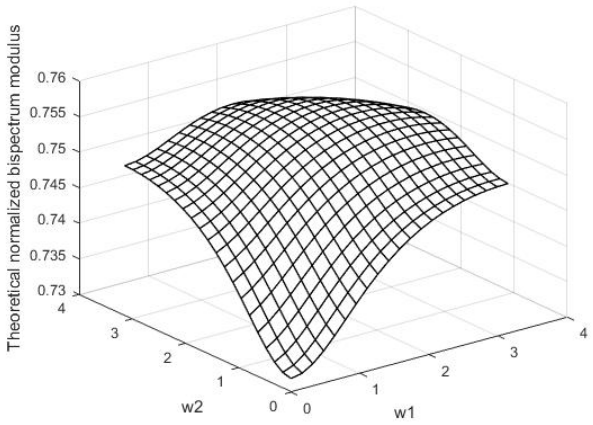
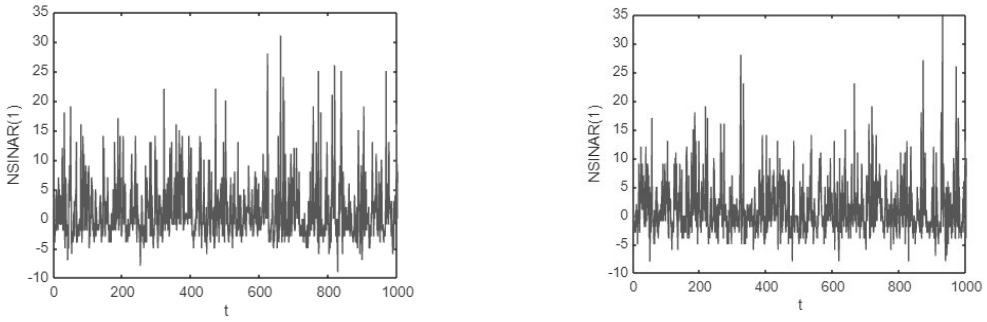


Figure 6. NBDF of NSINAR(1) using $\lambda = 0.3, \beta = 0.25$, and $\gamma = 5$.



(**a**) Basic model (**b**) Hybrid model

Figure 7. The simulated series of the forecasted NSINAR(1) observations of neural network-based FTS.

6. Estimations of SDF, BDF, and NBDF

The smoothed periodogram technique by the lag window is utilized to estimate the spectral density functions. The lag window is one-dimensional in the case of the SDF and two-dimensional lag in the case of the BDF. Generally, if $X_1, X_2, \ldots, X_N$ is the realizations of a real-valued third-order stationary process $\{X_t\}$ with mean γ, autocovariance $R_2(s)$, and third cumulant $R_3(s_1, s_2)$, the smoothed spectrum, smoothed bispectrum, and smoothed normalized bispectrum are, respectively, given by (see [28])

$$\begin{aligned}\hat{g}(w) &= \frac{1}{2\pi} \sum_{s=-(N-1)}^{N-1} \Phi(s)\hat{R}_2(s) \exp(-isw) \\ &= \frac{1}{2\pi} \sum_{s=-(N-1)}^{N-1} \Phi(s)\hat{R}_2(s) cosws, \end{aligned} \tag{12}$$

$$\hat{g}(w_1, w_2) = \frac{1}{4\pi^2} \sum_{s_1=-(N-1)}^{N-1} \sum_{s_2=-(N-1)}^{N-1} \Phi(s_1, s_2)\hat{R}_3(s_1, s_2) \exp(-is_1w_1 - is_2w_2), \tag{13}$$

$$\hat{f}(w_1, w_2) = \frac{\hat{g}(w_1, w_2)}{\sqrt{\hat{g}(w_1)\hat{g}(w_2)\hat{g}(w_1 + w_2)}}, \tag{14}$$

where $\hat{R}_2(t)$ and $\hat{R}_3(t_1, t_2)$, the natural estimators for $R_2(t)$ and $R_3(t_1, t_2)$ are, respectively, given by

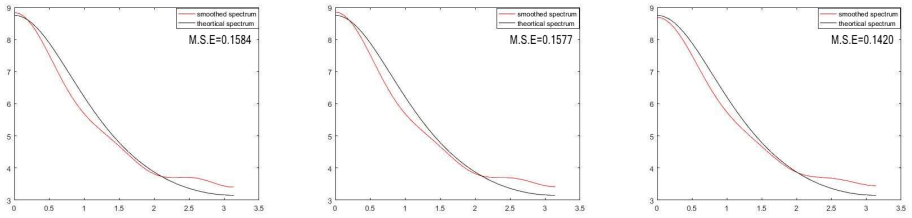
$$\hat{R}_2(s) = \frac{1}{N-s} \sum_{t=1}^{N-|s|} (X_t - \bar{X})(X_{t+|s|} - \bar{X}), \tag{15}$$

$$\bar{X} = \frac{1}{N} \sum_{t=1}^{N} X_t,$$

$$\hat{R}_3(s_1, s_2) = \frac{1}{N} \sum_{t=1}^{N-\beta} (X_t - \bar{X})(X_{t+s_1} - \bar{X})(X_{t+s_2} - \bar{X}), \tag{16}$$

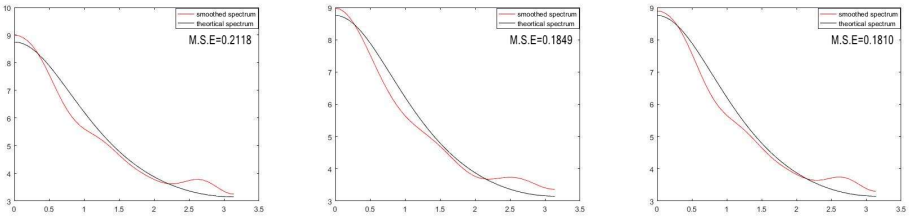
where $s_1, s_2 \geq 0, \beta = max(0, s_1, s_2), s = 0, \pm 1, \pm 2, ..., \pm(N-1), -\pi \leq w_1, w_2 \leq \pi$, "$\Phi(s)$" is a one-dimensional lag window and "$\Phi(s_1, s_2)$" $= \Phi(s_1 - s_2)\Phi(s_1)\Phi(s_2)$ is a two-dimensional lag window given by [58]. This section calculates estimates of the SDF, BDF, and NBDF by employing a smoothed periodogram approach that depends on the lag windows. The Daniell [59] window was chosen with a different number of frequencies M = 7 and M = 9. These estimates are calculated for three scenarios: (i) the generated realizations from NSINAR(1), which serve as the FTS's "input values", and (ii) and (iii), the forecasted observations, which serve as the output values of the FTS, which is based on a neural network.

Figures 8 and 9 illustrate the estimated SDF for the input and output values of the FTS using Daniell lag windows at M = 7 and M = 9, respectively. Figures 10 and 11 illustrate the estimated BDF for the input and output values of the FTS using Daniell lag windows at M = 7 and M = 9, respectively. Figures 12 and 13 illustrate the estimated NBDF for the input and output values of the FTS using Daniell lag windows at M = 7 and M = 9, respectively. Tables 3–5 display the estimated BDF obtained with the Daniell window when M = 7 for the input values for the FTS, as well as the output values of the FTS, which depend on a neural network (in the case of the basic and hybrid models).



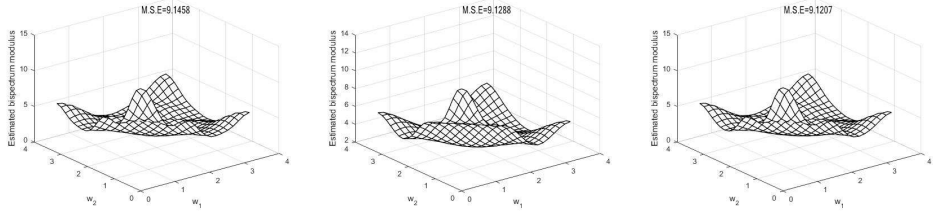
(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 8. The SDF and estimated SDF obtained with the Daniell window when M = 7, both in the case of the input and output values of the FTS.



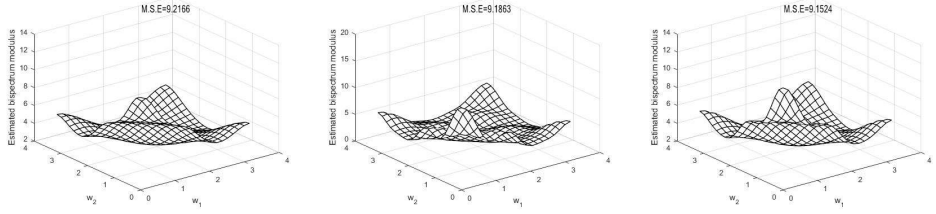
(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 9. The SDF and estimated SDF obtained with the Daniell window when M = 9, both in the case of the input and output values.



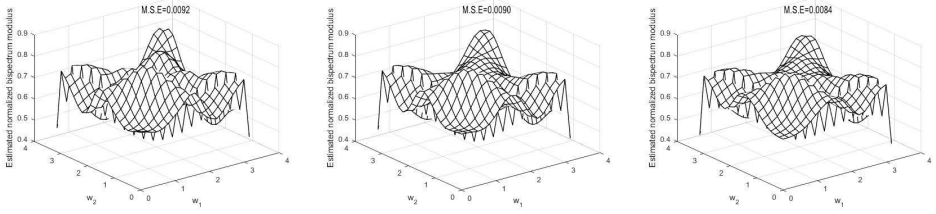
(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 10. Estimated BDF obtained with the Daniell window when M = 7, both in the case of the input and output values.



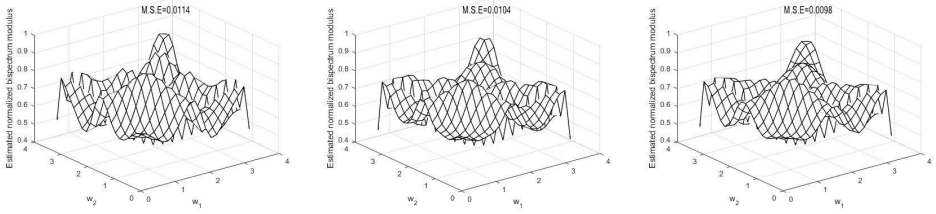
(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 11. Estimated BDF obtained with the Daniell window when M = 9 both in the case of the input and output values.



(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 12. Estimated NBDF obtained with the Daniell window when M = 7 both in the case of the input and output values.



(**i**) input values (**ii**) output values "Model 1" (**iii**) output values "Model 2"

Figure 13. Estimated NBDF obtained with the Daniell window when M = 9 both in the case of the input and output values.

Table 3. Estimated BDF of NSINAR(1) using the Daniell window at M=7 by the input values of the FTS.

	w_2	**0.00** π	**0.05** π	**0.10** π	**0.15** π	**0.20** π	**0.25** π	**0.30** π	**0.35** π	**0.40** π	**0.45** π	**0.50** π	**0.55** π	**0.60** π	**0.65** π	**0.70** π	**0.75** π	**0.80** π	**0.85** π	**0.90** π	**0.95** π	π
w_1																						
0.0 π		19.583	19.380	18.624	17.139	15.161	13.305	12.036	11.269	10.580	9.750	8.976	8.532	8.359	8.102	7.505	6.668	5.878	5.307	4.935	4.700	4.613
0.05 π		19.380	18.903	17.784	16.010	13.980	12.286	11.214	10.534	9.863	9.142	8.610	8.375	8.205	7.779	7.041	6.217	5.544	5.074	4.751	4.568	4.568
0.10 π		18.624	17.784	16.357	14.430	12.406	10.813	9.867	9.322	8.859	8.459	8.238	8.106	7.807	7.213	6.457	5.758	5.207	4.770	4.445	4.317	4.445
0.15 π		17.139	16.010	14.430	12.495	10.627	9.291	8.593	8.258	8.025	7.872	7.772	7.552	7.083	6.444	5.816	5.277	4.777	4.301	3.980	3.980	4.301
0.20 π		15.161	13.980	12.406	10.627	9.132	8.228	7.808	7.575	7.379	7.221	7.012	6.629	6.103	5.587	5.149	4.704	4.175	3.675	3.462	3.675	4.175
0.25 π		13.305	12.286	10.813	9.291	8.228	7.688	7.376	7.040	6.670	6.314	5.922	5.473	5.060	4.745	4.439	4.001	3.464	3.076	3.076	3.464	4.001
0.30 π		12.036	11.214	9.867	8.593	7.808	7.376	6.947	6.374	5.752	5.187	4.709	4.352	4.135	3.970	3.699	3.270	2.852	2.681	2.852	3.270	3.699
0.35 π		11.269	10.534	9.322	8.258	7.575	7.040	6.374	5.551	4.733	4.081	3.671	3.487	3.420	3.303	3.035	2.690	2.457	2.457	2.690	3.035	3.303
0.40 π		10.580	9.863	8.859	8.025	7.379	6.670	5.752	4.733	3.842	3.253	2.995	2.948	2.930	2.805	2.561	2.319	2.220	2.319	2.561	2.805	2.930
0.45 π		9.750	9.142	8.459	7.872	7.221	6.314	5.187	4.081	3.253	2.811	2.679	2.676	2.638	2.491	2.269	2.099	2.099	2.269	2.491	2.638	2.676
0.50 π		8.976	8.610	8.238	7.772	7.012	5.922	4.709	3.671	2.995	2.679	2.587	2.555	2.467	2.288	2.092	2.008	2.092	2.288	2.467	2.555	2.587
0.55 π		8.532	8.375	8.106	7.552	6.629	5.473	4.352	3.487	2.948	2.676	2.555	2.463	2.314	2.138	2.032	2.032	2.138	2.314	2.463	2.555	2.676
0.60 π		8.359	8.205	7.807	7.083	6.103	5.060	4.135	3.420	2.930	2.638	2.467	2.314	2.162	2.089	2.084	2.089	2.162	2.314	2.467	2.638	2.930
0.65 π		8.102	7.779	7.213	6.444	5.587	4.745	3.970	3.303	2.805	2.491	2.288	2.138	2.089	2.130	2.130	2.089	2.138	2.288	2.491	2.805	3.303
0.70 π		7.505	7.041	6.457	5.816	5.149	4.439	3.699	3.035	2.561	2.269	2.092	2.032	2.084	2.130	2.084	2.032	2.092	2.269	2.561	3.035	3.699
0.75 π		6.668	6.217	5.758	5.277	4.704	4.001	3.270	2.690	2.319	2.099	2.008	2.032	2.089	2.089	2.032	2.008	2.099	2.319	2.690	3.270	4.001
0.80 π		5.878	5.544	5.207	4.777	4.175	3.464	2.852	2.457	2.220	2.099	2.092	2.138	2.162	2.138	2.092	2.099	2.220	2.457	2.852	3.464	4.175
0.85 π		5.307	5.074	4.770	4.301	3.675	3.076	2.681	2.457	2.319	2.269	2.288	2.314	2.314	2.288	2.269	2.319	2.457	2.681	3.076	3.675	4.301
0.90 π		4.935	4.751	4.445	3.980	3.462	3.076	2.852	2.690	2.561	2.491	2.467	2.463	2.467	2.491	2.561	2.690	2.852	3.076	3.462	3.980	4.445
0.95 π		4.700	4.568	4.317	3.980	3.675	3.464	3.270	3.035	2.805	2.638	2.555	2.555	2.638	2.805	3.035	3.270	3.464	3.675	3.980	4.317	4.568
π		4.613	4.568	4.445	4.301	4.175	4.001	3.699	3.303	2.930	2.676	2.587	2.676	2.930	3.303	3.699	4.001	4.175	4.301	4.445	4.568	4.613

Table 4. Estimated BDF of NSINAR(1) using the Daniell window at M=7 by the output values of the neural network-based FTS "Model 1".

	w_2	**0.00** π	**0.05** π	**0.10** π	**0.15** π	**0.20** π	**0.25** π	**0.30** π	**0.35** π	**0.40** π	**0.45** π	**0.50** π	**0.55** π	**0.60** π	**0.65** π	**0.70** π	**0.75** π	**0.80** π	**0.85** π	**0.90** π	**0.95** π	π
w_1																						
0.0 π		19.523	19.160	18.115	16.535	14.695	12.941	11.551	10.610	10.005	9.545	9.109	8.679	8.274	7.859	7.350	6.701	5.983	5.344	4.906	4.685	4.624
0.05 π		19.160	18.460	17.165	15.475	13.672	12.054	10.835	10.043	9.534	9.133	8.746	8.367	7.987	7.547	6.980	6.295	5.604	5.054	4.721	4.581	4.581
0.10 π		18.115	17.165	15.785	14.150	12.455	10.948	9.839	9.162	8.774	8.489	8.206	7.895	7.528	7.060	6.465	5.795	5.171	4.713	4.459	4.382	4.459
0.15 π		16.535	15.475	14.150	12.630	11.061	9.701	8.761	8.251	7.995	7.800	7.565	7.263	6.879	6.396	5.820	5.208	4.664	4.280	4.089	4.089	4.280
0.20 π		14.695	13.672	12.455	11.061	9.678	8.573	7.884	7.531	7.315	7.082	6.786	6.437	6.042	5.590	5.077	4.544	4.079	3.765	3.655	3.765	4.079
0.25 π		12.941	12.054	10.948	9.701	8.573	7.773	7.296	6.975	6.640	6.245	5.835	5.456	5.103	4.730	4.303	3.854	3.476	3.261	3.261	3.476	3.854
0.30 π		11.551	10.835	9.839	8.761	7.884	7.296	6.865	6.400	5.838	5.265	4.797	4.465	4.207	3.929	3.586	3.230	2.967	2.873	2.967	3.230	3.586
0.35 π		10.610	10.043	9.162	8.251	7.531	6.975	6.400	5.692	4.926	4.279	3.861	3.637	3.483	3.280	3.008	2.751	2.605	2.605	2.751	3.008	3.280
0.40 π		10.005	9.534	8.774	7.995	7.315	6.640	5.838	4.926	4.084	3.494	3.194	3.075	2.982	2.813	2.592	2.412	2.346	2.412	2.592	2.813	2.982
0.45 π		9.545	9.133	8.489	7.800	7.082	6.245	5.265	4.279	3.494	3.026	2.831	2.760	2.666	2.494	2.294	2.164	2.164	2.294	2.494	2.666	2.760
0.50 π		9.109	8.746	8.206	7.565	6.786	5.835	4.797	3.861	3.194	2.831	2.678	2.586	2.450	2.263	2.096	2.028	2.096	2.263	2.450	2.586	2.678
0.55 π		8.679	8.367	7.895	7.263	6.437	5.456	4.465	3.637	3.075	2.760	2.586	2.440	2.274	2.120	2.029	2.029	2.120	2.274	2.440	2.586	2.760
0.60 π		8.274	7.987	7.528	6.879	6.042	5.103	4.207	3.483	2.982	2.666	2.450	2.274	2.145	2.084	2.070	2.084	2.145	2.274	2.450	2.666	2.982
0.65 π		7.859	7.547	7.060	6.396	5.590	4.730	3.929	3.280	2.813	2.494	2.263	2.120	2.084	2.105	2.105	2.084	2.120	2.263	2.494	2.813	3.280
0.70 π		7.350	6.980	6.465	5.820	5.077	4.303	3.586	3.008	2.592	2.294	2.096	2.029	2.070	2.105	2.070	2.029	2.096	2.294	2.592	3.008	3.586
0.75 π		6.701	6.295	5.795	5.208	4.544	3.854	3.230	2.751	2.412	2.164	2.028	2.029	2.084	2.084	2.029	2.028	2.164	2.412	2.751	3.230	3.854
0.80 π		5.983	5.604	5.171	4.664	4.079	3.476	2.967	2.605	2.346	2.164	2.096	2.120	2.145	2.120	2.096	2.164	2.346	2.605	2.967	3.476	4.079
0.85 π		5.344	5.054	4.713	4.280	3.765	3.261	2.873	2.605	2.412	2.294	2.263	2.274	2.274	2.263	2.294	2.412	2.605	2.873	3.261	3.765	4.280
0.90 π		4.906	4.721	4.459	4.089	3.655	3.261	2.967	2.751	2.592	2.494	2.450	2.440	2.450	2.494	2.592	2.751	2.967	3.261	3.655	4.089	4.459
0.95 π		4.685	4.581	4.382	4.089	3.765	3.476	3.230	3.008	2.813	2.666	2.586	2.586	2.666	2.813	3.008	3.230	3.476	3.765	4.089	4.382	4.581
π		4.624	4.581	4.459	4.280	4.079	3.854	3.586	3.280	2.982	2.760	2.678	2.760	2.982	3.280	3.586	3.854	4.079	4.280	4.459	4.581	4.624

Table 5. Estimated BDF of NSINAR(1) using the Daniell window at M=7 by the outputvalues of the neural network-based FTS "Model 2".

	w_2	**0.00** π	**0.05** π	**0.10** π	**0.15** π	**0.20** π	**0.25** π	**0.30** π	**0.35** π	**0.40** π	**0.45** π	**0.50** π	**0.55** π	**0.60** π	**0.65** π	**0.70** π	**0.75** π	**0.80** π	**0.85** π	**0.90** π	**0.95** π	π
w_1																						
0.0 π		18.710	18.588	18.004	16.674	14.807	13.023	11.736	10.844	10.064	9.349	8.834	8.526	8.245	7.834	7.284	6.648	5.975	5.355	4.914	4.701	4.649
0.05 π		18.588	18.230	17.235	15.555	13.638	12.054	10.975	10.179	9.473	8.909	8.563	8.322	7.991	7.502	6.907	6.259	5.609	5.064	4.728	4.595	4.595
0.10 π		18.004	17.235	15.819	13.965	12.165	10.802	9.884	9.213	8.695	8.369	8.182	7.944	7.532	6.986	6.389	5.775	5.187	4.722	4.453	4.371	4.453
0.15 π		16.674	15.555	13.965	12.212	10.688	9.585	8.839	8.313	7.976	7.804	7.643	7.323	6.836	6.294	5.759	5.220	4.698	4.283	4.058	4.058	4.283
0.20 π		14.807	13.638	12.165	10.688	9.495	8.648	8.043	7.600	7.321	7.131	6.864	6.444	5.957	5.504	5.071	4.602	4.125	3.757	3.618	3.757	4.125
0.25 π		13.023	12.054	10.802	9.585	8.648	7.969	7.422	6.970	6.613	6.277	5.870	5.424	5.032	4.706	4.359	3.935	3.513	3.249	3.249	3.513	3.935
0.30 π		11.736	10.975	9.884	8.839	8.043	7.422	6.856	6.316	5.800	5.295	4.821	4.447	4.190	3.963	3.660	3.287	2.972	2.851	2.972	3.287	3.660
0.35 π		10.844	10.179	9.213	8.313	7.600	6.970	6.316	5.624	4.940	4.342	3.908	3.655	3.501	3.319	3.048	2.757	2.573	2.573	2.757	3.048	3.319
0.40 π		10.064	9.473	8.695	7.976	7.321	6.613	5.800	4.940	4.161	3.585	3.255	3.101	2.991	2.824	2.598	2.398	2.320	2.398	2.598	2.824	2.991
0.45 π		9.349	8.909	8.369	7.804	7.131	6.277	5.295	4.342	3.585	3.108	2.873	2.764	2.661	2.503	2.316	2.185	2.185	2.316	2.503	2.661	2.764
0.50 π		8.834	8.563	8.182	7.643	6.864	5.870	4.821	3.908	3.255	2.873	2.686	2.583	2.468	2.305	2.143	2.073	2.143	2.305	2.468	2.583	2.686
0.55 π		8.526	8.322	7.944	7.323	6.444	5.424	4.447	3.655	3.101	2.764	2.583	2.463	2.324	2.160	2.048	2.048	2.160	2.324	2.463	2.583	2.764
0.60 π		8.245	7.991	7.532	6.836	5.957	5.032	4.190	3.501	2.991	2.661	2.468	2.324	2.178	2.072	2.040	2.072	2.178	2.324	2.468	2.661	2.991
0.65 π		7.834	7.502	6.986	6.294	5.504	4.706	3.963	3.319	2.824	2.503	2.305	2.160	2.072	2.060	2.060	2.072	2.160	2.305	2.503	2.824	3.319
0.70 π		7.284	6.907	6.389	5.759	5.071	4.359	3.660	3.048	2.598	2.316	2.143	2.048	2.040	2.060	2.040	2.048	2.143	2.316	2.598	3.048	3.660
0.75 π		6.648	6.259	5.775	5.220	4.602	3.935	3.287	2.757	2.398	2.185	2.073	2.048	2.072	2.072	2.048	2.073	2.185	2.398	2.757	3.287	3.935
0.80 π		5.975	5.609	5.187	4.698	4.125	3.513	2.972	2.573	2.320	2.185	2.143	2.160	2.178	2.160	2.143	2.185	2.320	2.573	2.972	3.513	4.125
0.85 π		5.355	5.064	4.722	4.283	3.757	3.249	2.851	2.573	2.398	2.316	2.305	2.324	2.324	2.305	2.316	2.398	2.573	2.851	3.249	3.757	4.283
0.90 π		4.914	4.728	4.453	4.058	3.618	3.249	2.972	2.757	2.598	2.503	2.468	2.463	2.468	2.503	2.598	2.757	2.972	3.249	3.618	4.058	4.453
0.95 π		4.701	4.595	4.371	4.058	3.757	3.513	3.287	3.048	2.824	2.661	2.583	2.583	2.661	2.824	3.048	3.287	3.513	3.757	4.058	4.371	4.595
π		4.649	4.595	4.453	4.283	4.125	3.935	3.660	3.319	2.991	2.764	2.686	2.764	2.991	3.319	3.660	3.935	4.125	4.283	4.453	4.595	4.649

6.1. Discussion of Results

In contrast to the input values for the FTS and the output values of the FTS, which depend on a neural network (in the case of the basic and hybrid models), and depending on the M.S.E at the top of each image, we discover the following:

- Figures 8–13 illustrate the preference for the hybrid model, followed by the basic model, over the input values in estimating the SDF, BDF, and NBDF, as the hybrid model's outputs had the lowest mean-squared error (refer to [32] and [33] to find out how to compute the MSE). In general, this indicates that the neural network-based FTS (whether used with the basic model or the hybrid model) significantly improved the smoothing of density functions' estimates.
- The Daniell window at M = 7 performs better than that at M = 9 in estimating the SDF for all three situations (input values and output values for the basic and hybrid models), according to a comparison of Figures 8 and 9.
- The Daniell window at M = 7 performs better than that at M = 9 in estimating the BDF for all three situations (input values and output values for the basic and hybrid models), according to a comparison of Figures 10 and 11.
- The Daniell window at M = 7 performs better than that at M = 9 in estimating the NBDF for all three situations (input values and output values for the basic and hybrid models), according to a comparison of Figures 12 and 13.

6.2. Empirical Analysis

To create a series with size n = 1000, we employed NSINAR(1). Data ranging from the first observation to the 800th were utilized for our estimation (the in-sample), whereas data spanning the 801st and 1000th were used for our forecast (the out-of-sample). These observations were generated from NSINAR(1) fifteen times, and thus, fifteen time series are available. The proposed method is applied to each time series generated from NSINAR(1). In the following scenarios, the estimators of the spectral density functions were found: for the observations produced by the NSINAR(1) model and for the observations estimated by the basic and hybrid models using the suggested methodology. The MSE was used to compare these estimators in the three scenarios. The input values were compared to the respective performances of the two models. Based on the mean-squared errors (MSEs) of each of the fifteen series, we discovered that the hybrid model outperformed the basic model in terms of performance. When comparing all the series, the hybrid model outperformed the input values in 12 out of the 15 series, while the basic model outperformed the input values in 9 out of the 15 series.

7. Conclusions

More precise smoothing estimates of the SDF, BDF, and NBDF for the INAR(1) models were suggested using a unique strategy. A neural network-based FTS was employed in this method. Consequently, the SDF, BDF, and NBDF for a recognized process (NSINAR(1)) were computed. The observations produced by this process served as the FTS input values. In order to anticipate the observations of NSINAR(1)'s observations, two models were employed: the basic and hybrid models, both applying neural networks. These forecasted observations served as the output values of the neural network-based FTS. Both the input and output values were used for calculating the estimates of the SDF, BDF, and NBDF of NSINAR(1). A definite improvement in the estimation for the hybrid model over the basic model, as well as the input values was discovered by comparing all density functions with their estimations in each scenario. As a result, the neural network-based FTS (either via the basic model or the hybrid model) helped further improve the smoothing of the INAR(1) estimates. Future research will attempt to improve the aforementioned technique using pi-sigma artificial neural networks, despite the fact that this requires far more work than this method to achieve the best estimate smoothing when compared to the findings reached here.

Author Contributions: Methodology, M.E.-M., M.S.E. and R.M.E.-S.; Software, M.E.-M. and M.H.E.-M.; Validation, M.S.E. and M.H.E.-M.; Formal analysis, M.S.E. and R.M.E.-S.; Investigation, M.E., M.E.-M. and L.A.A.-E.; Resources, M.E.-M. and L.A.A.-E.; Data curation, M.H.E.-M., M.S.E. and M.E.-M.; Writing—original draft, M.E.-M.; Writing—review & editing, M.S.E. and R.M.E.-S.; Visualization, M.H.E.-M., M.S.E., L.A.A.-E. and M.E.-M.; Project administration, M.E.-M and R.M.E.-S.; Funding acquisition, L.A.A.-E. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R443), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. This study is supported via funding from Prince sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Song, Q.; Chissom, B.S. Fuzzy time series and its models. *Fuzzy Sets Syst.* **1993**, *54*, 269–277. [CrossRef]
2. Huarng, K. Effective lengths of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets Syst.* **2001**, *123*, 387–394. [CrossRef]
3. Yu, T.H. A refined fuzzy time-series model for forecasting. *Phys. A Stat. Mech. Its Appl.* **2005**, *346*, 657–681. [CrossRef]
4. Huarng, K.; Yu, T.H. Ratio-based lengths of intervals to improve enrollment forecasting. In Proceedings of the Ninth International Conference on Fuzzy Theory and Technology, Cary, NC, USA, 23 September 2003.
5. Huarng, K.; Yu, T.H. A dynamic approach to adjusting lengths of intervals in fuzzy time series forecasting. *Intell. Data Anal.* **2004**, *8*, 3–27. [CrossRef]
6. Chen, S.M. Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst.* **1996**, *81*, 311–319. [CrossRef]
7. Chen, S.M. Forecasting enrollments based on high-order fuzzy time series. *Cybern. Syst.* **2002**, *33*, 1–16. [CrossRef]
8. Hwang, J.R.; Chen, S.M.; Lee, C.H. Handling forecasting problems using fuzzy time series. *Fuzzy Sets Syst.* **1998**, *100*, 217–228. [CrossRef]
9. Song, Q.; Chissom, B.S. Forecasting enrollments with fuzzy time series-Part II. *Fuzzy Sets Syst.* **1994**, *62*, 1–8. [CrossRef]
10. Huarng, K. Heuristic models of fuzzy time series for forecasting. *Fuzzy Sets Syst.* **2001**, *123*, 369–386. [CrossRef]
11. Kitchens, F.L.; Johnson, J.D.; Gupta, J.N. Predicting automobile insurance losses using artificial neural networks. In *Neural Networks in Business: Techniques and Applications*; IGI Global: Hershey, PA, USA, 2002; pp. 167–187.
12. Widrow, B.; Rumelhart, D.E.; Lehr, M.A. Neural networks: Applications in industry, business and science. *Commun. ACM* **1994**, *37*, 93–106. [CrossRef]
13. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]
14. McKenzie, E. Autoregressive moving-average processes with negative-binomial and geometric marginal distributions. *Adv. Appl. Probab.* **1986**, *18*, 679–705. [CrossRef]
15. Al-Osh, M.A.; Aly, E.A.A. First order autoregressive time series with negative binomial and geometric marginals. *Commun. Stat.-Theory Methods* **1992**, *21*, 2483–2492. [CrossRef]
16. Alzaid, A.; Al-Osh, M. First-order integer-valued autoregressive (INAR(1)) process: Distributional and regression properties. *Stat. Neerl.* **1988**, *42*, 53–61. [CrossRef]
17. Bakouch, H.S.; Ristić, M.M. Zero truncated Poisson integer-valued AR(1) model. *Metrika* **2010**, *72*, 265–280. [CrossRef]
18. Alzaid, A.A.; Al-Osh, M.A. Some autoregressive moving average processes with generalized Poisson marginal distributions. *Ann. Inst. Stat. Math.* **1993**, *45*, 223–232. [CrossRef]
19. Jin-Guan, D.; Yuan, L. The integer-valued autoregressive (INAR(p)) model. *J. Time Ser. Anal.* **1991**, *12*, 129–142. [CrossRef]
20. Aly, E.A.A.; Bouzar, N. Explicit stationary distributions for some Galton-Watson processes with immigration. *Stoch. Model.* **1994**, *10*, 499–517. [CrossRef]
21. Ristić, M.M.; Nastić, A.S.; Miletić, A.V. A geometric time series model with dependent Bernoulli counting series. *J. Time Ser. Anal.* **2013**, *34*, 466–476. [CrossRef]
22. Latour, A. Existence and stochastic structure of a non-negative integer-valued autoregressive process. *J. Time Ser. Anal.* **1998**, *19*, 439–455. [CrossRef]
23. Ristić, M.M.; Bakouch, H.S.; Nastić, A.S. A new geometric first-order integer-valued autoregressive (NGINAR(1)) process. *J. Stat. Plan. Inference* **2009**, *139*, 2218–2226. [CrossRef]
24. Ristić, M.M.; Nastić, A.S.; Bakouch, H.S. Estimation in an integer-valued autoregressive process with negative binomial marginals (NBINAR(1)). *Commun. Stat.-Theory Methods* **2012**, *41*, 606–618. [CrossRef]
25. Ristić, M.M.; Nastić, A.S.; Jayakumar, K.; Bakouch, H.S. A bivariate INAR(1) time series model with geometric marginals. *Appl. Math. Lett.* **2012**, *25*, 481–485. [CrossRef]

26. Nastić, A.S.; Ristić, M.M. Some geometric mixed integer-valued autoregressive (INAR) models. *Stat. Probab. Lett.* **2012**, *82*, 805–811. [CrossRef]
27. Nastić, A.S.; Ristić, M.M.; Bakouch, H.S. A combined geometric INAR(p) model based on negative binomial thinning. *Math. Comput. Model.* **2012**, *55*, 1665–1672. [CrossRef]
28. Gabr, M.M.; El-Desouky, B.S.; Shiha, F.A.; El-Hadidy, S.M. Higher Order Moments, Spectral and Bispectral Density Functions for INAR(1). *Int. J. Comput. Appl.* **2018**, *182*, 0975–8887.
29. Miletić, A.V.; Ristić, M.M.; Nastić, A.S.; Bakouch, H.S. An INAR(1) model based on a mixed dependent and independent counting series. *J. Stat. Comput. Simul.* **2018**, *88*, 290–304. [CrossRef]
30. Teamah, A.A.M.; Faied, H.M.; El-Menshawy, M.H. Using the Fuzzy Time Series Technique to Improve the Estimation of the Spectral Density Function. *J. Stat. Adv. Theory Appl.* **2018**, *19*, 151–170. [CrossRef]
31. Teamah, A.A.M.; Faied, H.M.; El-Menshawy, M.H. Effect of Fuzzy Time Series Technique on Estimators of Spectral Analysis. *Recent Adv. Math. Res. Comput. Sci.* **2022**, *6*, 29–38.
32. El-menshawy, M.H.; Teamah, A.A.M.; Abu-Youssef, S.E.; Faied, H.M. Higher Order Moments, Cumulants, Spectral and Bispectral Density Functions of the ZTPINAR(1) Process. *Appl. Math* **2022**, *16*, 213–225.
33. El-Morshedy, M.; El-Menshawy, M.H.; Almazah, M.M.A.; El-Sagheer, R.M.; Eliwa, M.S. Effect of fuzzy time series on smoothing estimation of the INAR(1) process. *Axioms* **2022**, *11*, 423. [CrossRef]
34. Alqahtani, K.M.; El-Menshawy, M.H.; Eliwa, M.S.; El-Morshedy, M.; EL-Sagheer, R.M. Fuzzy Time Series Inference for Stationary Linear Processes: Features and Algorithms With Simulation. *Appl. Math* **2023**, *17*, 405–416.
35. El-Menshawy, M.H.; Teamah, A.E.M.A.; Eliwa, M.S.; Al-Essa, L.A.; El-Morshedy, M.; EL-Sagheer, R.M. A New Statistical Technique to Enhance MCGINAR (1) Process Estimates under Symmetric and Asymmetric Data: Fuzzy Time Series Markov Chain and Its Characteristics. *Symmetry* **2023**, *15*, 1577. [CrossRef]
36. Huarng, K.; Yu, T.H. The application of neural networks to forecast fuzzy time series. *Phys. A Stat. Mech. Its Appl.* **2006**, *363*, 481–491. [CrossRef]
37. Egrioglu, E.; Aladag, C.H.; Yolcu, U.; Uslu, V.R.; Basaran, M.A. A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Expert Syst. Appl.* **2009**, *36*, 10589–10594. [CrossRef]
38. Yu, T.H.K.; Huarng, K.H. A neural network-based fuzzy time series model to improve forecasting. *Expert Syst. Appl.* **2010**, *37*, 3366–3372. [CrossRef]
39. Egrioglu, E.; Aladag, C.H.; Yolcu, U. Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. *Expert Syst. Appl.* **2013**, *40*, 854–857. [CrossRef]
40. Rahman, N.H.A.; Lee, M.H.; Suhartono; Latif, M.T. Artificial neural networks and fuzzy time series forecasting: An application to air quality. *Qual. Quant.* **2015**, *49*, 2633–2647. [CrossRef]
41. Bas, E.; Grosan, C.; Egrioglu, E.; Yolcu, U. High order fuzzy time series method based on pi-sigma neural network. *Eng. Appl. Artif. Intell.* **2018**, *72*, 350–356. [CrossRef]
42. Sadaei, H.J.; Silva, P.C.; Guimaraes, F.G.; Lee, M.H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* **2019**, *175*, 365–377. [CrossRef]
43. Koo, J.W.; Wong, S.W.; Selvachandran, G.; Long, H.V.; Son, L.H. Prediction of Air Pollution Index in Kuala Lumpur using fuzzy time series and statistical models. *Air Qual. Atmos. Health* **2020**, *13*, 77–88. [CrossRef]
44. Tang, Y.; Yu, F.; Pedrycz, W.; Yang, X.; Wang, J.; Liu, S. Building trend fuzzy granulation-based LSTM recurrent neural network for long-term time-series forecasting. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 1599–1613. [CrossRef]
45. Nasiri, H.; Ebadzadeh, M.M. MFRFNN: Multi-functional recurrent fuzzy neural network for chaotic time series prediction. *Neurocomputing* **2022**, *507*, 292–310. [CrossRef]
46. Rathipriya, R.; Abdul Rahman, A.A.; Dhamodharavadhani, S.; Meero, A.; Yoganandan, G. Demand forecasting model for time-series pharmaceutical data using shallow and deep neural network model. *Neural Comput. Appl.* **2023**, *35*, 1945–1957. [CrossRef] [PubMed]
47. Zhang, G.P. Business forecasting with artificial neural networks: An overview. In *Neural Networks in Business Forecasting*;2004; pp. 1–22.
48. Indro, D.C.; Jiang, C.X.; Patuwo, B.; Zhang, G. Predicting mutual fund performance using artificial neural networks. *Omega* **1999**, *27*, 373–380. [CrossRef]
49. Rumelhart, D.E.; McClelland, J.L.; Parallel Distributed Processing Research Group, C. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986.
50. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
51. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
52. Hornik, K. Some new results on neural network approximation. *Neural Netw.* **1993**, *6*, 1069–1072. [CrossRef]
53. Armstrong, J.S. *Principles of Forecasting: A Handbook for Researchers and Practitioners*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 30.
54. Zhang, G.P. *Neural Networks in Business Forecasting*; IGI Global: Hershey, PA, USA, 2004.
55. Song, Q.; Chissom, B.S. Forecasting enrollments with fuzzy time series—Part I. *Fuzzy Sets Syst.* **1993**, *54*, 1–9. [CrossRef]
56. Bourguignon, M.; Vasconcellos, K.L. A new skew integer valued time series process. *Stat. Methodol.* **2016**, *31*, 8–19. [CrossRef]

57. Eduarda Da Silva, M.; Oliveira, V.L. Difference equations for the higher-order moments and cumulants of the INAR (1) model. *J. Time Ser. Anal.* **2010**, *25*, 317–333. [CrossRef]
58. Rao, T.S.; Gabr, M.M. *An Introduction to Bispectral Analysis and Bilinear Time Series Models*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1984; Volume 24.
59. Daniell, P.J. Discussion on symposium on autocorrelation in time series. *J. R. Stat. Soc.* **1946**, *8*, 88–90.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.