A Stacking Neuro-Fuzzy framework for time series forecasting from distributed data sources: The case of Rainfall-Runoff modeling

Universidad de Valparaíso
CHILE

Facultad de Ciencias
Departamento de Estadística

Marvin Isaac Querales Carrasquel

Advisor: Dr. Rodrigo Salas
Co-Advisor: Dr. Harvey Rosas

This dissertation is submitted for the degree of
Doctor en Estadística

February 2021

Executive summary

Time series is a sequence of data collected at a regular and specific periods like hourly, daily, weekly, monthly or yearly [1]; it appears in many different areas and may involve a dynamic phenomena [2]. For example, forecasting the Runoff-Rainfall relationship (forecasting runoff values from previous values of both runoff and rainfall among other variables) is a main field in hydrology because it provides information to engineers for the urban and environmental planning such as land use, flood and water resources management of a watershed, among others [3]. In economy, time series forecasting in stock price prediction, investment, and financial sector helps investors, fund managers, and insurance companies to make decisions for better returns [4].

There exist several approaches for developing a forecasting model. Two of the most common are, on the one hand, time series methods that encompass techniques such as Autoregression (AR), Moving Average (MA), Autoregression Moving Average (ARMA), Autoregressive Integrated Moving Average Model (ARIMA), Autoregressive Conditional Heteroscedasticity model (ARCH) and Generalized Autoregressive Conditional Heteroscedasticity model (GARCH), among others [5]. On the other hand, machine learning techniques such as support vector regression (SVR) and Artificial Neural Networks (ANNs) have received a lot of attention in time series forecasting [2], because they outperform the classical methods. ANNs can learn a non-linear relationship between input and output to approximate any continuous function (beneficial for complex data) [6]. There are a lot of algorithms but the feed-forward backpropagation type is frequently applied due to having less complexity than other architectures [7].

ANNs are effective for handling datasets with noisy characteristics, especially in cases where the underlying physical relationships are not clear. In uncertain environments, fuzzy logic has emerged as a useful approach to model time series. Neuro-Fuzzy model is a structure in which the neural network uses the training data to determine the membership functions (a curve that defines how each point in the input space is mapped to a degree of membership) and fuzzy rules of the fuzzy logic system [8]. The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a fuzzy system to represent knowledge in an interpretative manner and the learning ability of a neural network to optimize its parameters. Thus, the drawbacks of both of the individual approaches - the black-box behavior of neural networks and the problems of finding suitable membership values for fuzzy systems- could be mitigated [9].

Many Neuro-Fuzzy systems for function approximation are based on Takagi-Sugeno fuzzy systems, because they allow the application of gradient descent learning if a differentiable membership function (e.g. Gaussian) is used [10]. The first and still one of the popular Neuro-Fuzzy systems is Jang's ANFIS model proposed in 1991 [11]. ANFIS (adaptive network-based fuzzy inference system) is a Neuro-Fuzzy method to determine the parameters of a Sugeno-type fuzzy model which is represented as a special feed-forward network. This model has been used to predict real-world time-series data such as water level prediction [12], weather forecasting [13], Worldwide Interoperability for Microwave Access (WiMAX) traffic prediction [14] and runoff prediction ([15, 16]), among others. However, ANFIS employs offline learning which suffers from increased computational time and requires retraining to capture recent changes in the system. Moreover, the number of rules in ANFIS is predefined by the user, and overfitting is a common problem when the data are overtrained by ANFIS [17].

Therefore, several models have been proposed for improving the accuracy of the forecast. Various hybrid models, like the Fuzzy information systems with ARIMA [18], Fuzzy wavelet neural networks with GARCH [19], or fuzzy systems with recurrent component [20], have been successfully applied in time series prediction. On the other hand, an extension of ANFIS model called Self-Organization Neuro-Fuzzy Inference System (SONFIS), has been proposed by [21, 22]. Its basic structure and functionality are similar to the ANFIS model but, during the learning procedure, the SONFIS self organizes its architecture to automatically identify the number of premises and consequences needed to model the available complex and high dimensional data.

An aspect to consider in forecasting models is that information can be obtained from multiple types of instruments, experimental designs, and other types of sources. As phenomena usually tend to be studied from different perspectives, it is unusual for a single method of data acquisition to provide a complete understanding of it. Besides, studying the different sources of information provides a better description of the event than doing it with each measure separately [23]. Data fusion and stacking models use a combination of multiple sources to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor [24]. The interest in stacking arises from two main benefits it can provide. First, having access to multiple modalities that observe the same phenomenon may allow for more robust predictions. Second, having access to multiple modalities might allow us to capture complementary information, something that is not visible in individual modalities on their own [25].

Data acquired from multiple sensors can be fused or stacked at a variety of levels: the raw data level, the feature level, or the decision level. At the raw data level, Diao et al. [26] proposed a novel algorithm to reduce the amount of incoming data by combining several multivariate time series into one. In the feature level, Guo et al. [27], developed a forecasting model to merge features based on financial time series. They extracted different features from time series that were then combined to obtain the next day's price. Fusion at the decision level is the most common approach. Many studies use models for stacking the results of different classifiers to obtain a global output [28, 29]. This level implies the usage of an statistical approach such as simple and weighted average, median, selection of the best initial results, multiple linear regression [30], among others, or to develop a model based on the concept of stacking generalization, a type of ensemble learning that combines the output of a number of single models or base learners through another learner such as ANN or another machine learning approach [31, 32, 30, 33, 34, 35, 36, 37]

Most fusion studies on time series at different levels are developed to obtain a good accuracy in the classification task and there are only a few of proposals that focus on the regression task. Since few studies combine Neuro-Fuzzy models with fusion or stacking techniques for time series forecasting and considering that stacking would have lower prediction error than the single forecasts, in this work, a new framework combining a Neuro-Fuzzy model and a stacking approach to improve time series forecasting is proposed. The novelty of this work focuses on the generation of a Neuro-Fuzzy framework that allows single predictions and stacking of various sources of information to perform a regression task on time series.

This thesis is organized as follow

Chapter 1 presents an extended introduction to the work and the general and specific objectives are described. Chapter 2 includes a review of the literature associated with the state of the art, including the principles of time series and fuzzy logic, Neuro-Fuzzy models, and a review of data fusion approaches on time series forecasting. Chapter 3 covers the proposal for the new Stacking Neuro-Fuzzy framework for time series, including a new Neuro-Fuzzy framework called Self- Identification Neuro Fuzzy Inference Model (SINFIM) for single time series forecasting and the alternative applied as stacking approach. Chapter 4 focuses on experimental results, using a hydrological application for modeling the Rainfall-Runoff relationship in two Chilean watersheds as case study. Chapter 5 presents some discussion. Finally, the Chapter 6 concludes with some remarks and future work.

"You can’t put a limit on anything. The more you dream, the farther you get"
Michael Phelps

Acknowledgements

First of all, I want to deeply thank my Advisor, Dr. Rodrigo Salas, for agreeing to guide me in this work, for all the knowledge imparted, and for his unconditional support throughout my academic life in the doctoral program. Likewise, to my co-Advisor, Dr. Harvey Rosas, for guiding me in the development of the work and for teaching me clearly and precisely important aspects of this work.

In the same way, I would like to thank Dr. Héctor Allende-Cid for providing me with the key knowledge for the development of this doctoral thesis and Prof. Yerel Morales, who with her advice on hydrology, was able to guide me adequately in the applied aspects of the proposed model .

Also, I want to thank the Department of Statistics of the Universidad de Valparaíso for allowing me to have entered the doctoral program and have been trained with a high technical and scientific quality.

Finally, I would like to thank in a special way my fellow students for both the Doctorate in Statistics, as well as the Doctorate in Sciences mention Neuroscience and Master in Engineering Sciences mention Biomedical Engineering of the Universidad de Valparaíso. They became a fundamental base of my academic training and without them, I would not have been able to achieve this goal. I will be enormously grateful and I wish them a million blessings.

Abstract

Neuro-fuzzy models have emerged as a useful approach for time series forecasting given the ability of neural networks to optimize parameters in a fuzzy system. However, learning structures have to consider that information can be obtained from different types of instruments, experimental designs, and other types of sources. Considering that most stacking approaches are based on the classification task, the aim of this work was to propose a new stacking Neuro-Fuzzy framework for the regression task in time series forecasting. This proposal was developed by taking the Rainfall-Runoff modeling phenomenon as a case study. The proposed framework involves two modules: the new Self-Identification Neuro-Fuzzy Inference model (SINFIM) developed for single forecasting and the Self-Organizing Neuro-Fuzzy Inference System (SONFIS) as stacking approach. Data from two Chilean watersheds (the Diguillín river (Ñuble region) and Colorado river (Maule region)) and average daily runoff and average daily rainfall recorded from eighteen years were collected from the Chilean directorate of water resources (DGA). The experimental results showed a good fit in most single forecasting of runoff, with meteorological stations showing both fit and efficiency indexes greater than 80% in the validation set, being able to efficiently predict both high and low runoff values. However, better results were obtained with the stacking model, being higher than single runoff predictions and other approaches such as average, multiple linear regression (MLP), Artificial Neural Network (ANN), and Adaptive Neuro-Inference System (ANFIS). Therefore, the general framework proposed represents a solid alternative to forecast (in the study case) the runoff in a given watershed, since it obtains good performance rates and the fusion approach can improve the prediction and get more accurate values than single models.

Resumen

Los modelos Neuro-Difusos han surgido como un enfoque útil para el pronóstico de series de tiempo dada la capacidad de las redes neuronales para optimizar los parámetros en un sistema difuso. Sin embargo, las estructuras de aprendizaje deben considerar que la información se puede obtener de diferentes tipos de instrumentos, diseños experimentales y otros tipos de fuentes. Teniendo en cuenta que la mayoría de los enfoques de Stacking se basan en la tarea de clasificación, el objetivo de este trabajo fue proponer un nuevo modelo Stacking Neuro-Difuso para la tarea de regresión en el pronóstico de series de tiempo. Esta propuesta fue desarrollada tomando como caso de estudio el fenómeno de modelado de Precipitación-Caudal. El modelo propuesto comprende dos módulos: el nuevo método Self-Identification Neuro-Fuzzy Inference model (SINFIM) desarrollado para el pronóstico individual y el Self-Organizing Neuro-Fuzzy Inference System (SONFIS) como herramienta de stacking. Los datos de dos cuencas hidrográficas chilenas (el río Diguillín (Región de Ñuble) y el río Colorado (Región de Maule)) y la escorrentía diaria promedio y la precipitación diaria promedio registrada de dieciocho años fueron recolectados de la Dirección General de Aguas de Chile (DGA). Los resultados experimentales mostraron un buen ajuste en las predicciones individuales de caudal, con estaciones meteorológicas mostrando índices de ajuste y eficiencia superiores al 80 % en el conjunto de validación, pudiendo predecir eficientemente tanto valores altos como bajos de escorrentías. Sin embargo, se obtuvieron mejores resultados con el modelo de Stacking, siendo superior a las predicciones de escorrentía individuales y otros enfoques como el promedio, regresión lineal múltiple (MLP), Red neuronal artificial (ANN) y Sistema de Neuro-Inferencia adaptativa (ANFIS). Por lo tanto, el marco general propuesto representa una alternativa sólida para pronosticar (en el caso de estudio) la escorrentía en una cuenca determinada, ya que obtiene buenos índices de desempeño y el enfoque de fusión puede mejorar la predicción y obtener valores más precisos que los modelos individuales.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

A time series is a set of observations x_t, each one being recorded at a specific time t. These observations are made sequentially at specific period like hourly, daily, weekly, monthly or yearly [1], and are found in a wide variety of areas such as economics and finance, environmental modeling, meteorology and hydrology, demographics, medicine, engineering, quality control, among others [2, 3, 4]. Time series modeling is a dynamic process that studies the past observations of a time series to develop an appropriate model that describes the inherent structure of the series. This model is then used to generate future values for the series, that is, to make forecasts. Therefore, time series forecasting can be termed as the act of predicting the future by understanding the past, and which success depends on an appropriate model fitting [39].

There exist several approaches to develop models for time series forecasting that could be grouped in two categories: classic statistical models and machine learning approaches. In the first group, there are linear methods such like Autoregressive (AR) and Moving Average (MA) models. Combining these two, the Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) models have been proposed in literature. The Autoregressive Fractionally Integrated Moving Average (ARFIMA) model generalizes ARMA and ARIMA models [40]. For seasonal time series forecasting, a variation of ARIMA, the Seasonal Autoregressive Integrated Moving Average (SARIMA) model is used. However many practical time series show non-linear patterns such as, for example, the volatility changes in economic and financial time series. Considering this fact, various nonlinear statistical models have been suggested such as the Autoregressive Conditional Heteroskedasticity (ARCH) model and its variations like Generalized ARCH (GARCH) and Exponential Generalized ARCH (EGARCH) [9], the Threshold Autoregressive (TAR) model, the Non-linear Autoregressive (NAR) model, and the Nonlinear Moving Average (NMA) model, among others [5].

In the second group, machine learning techniques such as Support Vector Regression (SVR) and Artificial Neural Networks (ANNs) are received recently a lot of attention in time series forecasting [2] because these approaches outperform the classical methods. The main feature of ANNs, when applied to time series forecasting problems is their inherent capability of non-linear modeling, without any presumption about the statistical distribution followed by the observations. The appropriate model is adaptively formed based on the given data [6]. There are a lot of algorithms but the feed-forward backpropagation type is frequently applied due to having less complexity than other architectures [7]. ANNs are effective for handling high dimensional datasets with noisy characteristics, especially in cases where the underlying physical relationships are not clear.

In uncertain environments, fuzzy logic has emerged as a useful approach to model time series. A Neuro-Fuzzy model combines ANN and fuzzy logic and effectively integrates the learning capability of neural networks into the development process of a fuzzy inference system. That is, it helps to determine the membership functions and fuzzy rules through learning from the data using the neural network [8, 41]. In this way, the accuracy of modeling by the fuzzy system can be greatly enhanced. Thus, the drawbacks of both of the individual approaches - the black-box behavior of neural networks and the problems of finding suitable membership values for fuzzy systems- could be mitigated [9]

Due to the different connections between ANN and the fuzzy system, a number of Neuro-Fuzzy models have been proposed. The Adaptive-Network-Based Fuzzy Inference System (ANFIS) proposed by Jang [11] was one of the first and still one of the popular Neuro-Fuzzy systems, and it represents a Sugeno-type fuzzy inference system implemented in the framework of an adaptive neural network with supervised learning capability. It has been widely adopted in many real world applications such as such as water level prediction [12], weather forecasting [13], Worldwide Interoperability for Microwave Access (WiMAX) traffic prediction [14] and runoff prediction ([15, 16]), among others.

However, ANFIS employs offline learning which suffers from increased computational time and requires retraining to capture recent changes in the system. Moreover, the number of rules in ANFIS is predefined by the user, and overfitting is a common problem when the data are overtrained by ANFIS. In fact, every data set that is trained using ANFIS has its maximum number of epochs before overfitting occurs [17]. In order to overcome these drawbacks, hybrid algorithms have been proposed using techniques such as Principal Component Analysis (ANFIS-PCA), wavelet transform (WANFIS), Particle Swarm Optimization (ANFIS-PSO), or Grey Wolf Optimizer (ANFIS-GWO), among others [42, 43, 44, 45]. Another method is Self-Organization Neuro-Fuzzy Inference System (SONFIS), an extension of ANFIS proposed by Allende-Cid et al. [21, 22], where the network automatically organizes itself to identify the set of Takagi-Sugeno-Kang type rules necessary to model the data set, overcoming the problem to set it by user.

An aspect to consider in forecasting models is that information can be obtained from different types of instruments, experimental designs, and other types of sources. As phenomena usually tend to be studied from different perspectives, it is unusual for a single method of data acquisition to provide a complete understanding of it [23]. Thus, a suitable combination of various time series models could provide a more complete and accurate representation of phenomena than each individual model [30]. An alternative is to develop a model based on the concept of stacking generalization, a type of ensemble learning that combines the output of a number of single models or base learners through another learner [31, 46]. This integration of single models achieves improved accuracy and more specific inferences than using a single sensor [24], and it might allow to capture complementary information, something that it is not visible in individual modalities on their own [25].

Considering this idea of integrating the output of different models, several schemes have been applied for many forecasting areas, using various statistical methods such as simple and weighted average, median, selection of the best initial results, multiple linear regression or ANN, among others [30, 33, 34, 35, 36, 37]. In most of these studies, stacking methods showed more accurate forecasts than single models, but ANN outperformed the other approaches. Based on this, some research have proposed specific types of ANN for stacking outputs such as radial basis function neural networks (RBFNN) [47, 36], back propagation neural network (BPNN), Elman neural network (ENN) [48] or hybrid methods based on continuity equation [49] or Grey relational analysis (GRA) [50]. Despite good performance obtained using ANN, some researchers suggest it may not be able to model non-stationary data, mainly if a pre-processing of input and/or output step is not performed [51, 52, 53]. Thus, fuzzy approaches have been used as combination models showing good results [54, 55]. However, Neuro-Fuzzy models have not been applied as stacking alternatives, only as single predictive methods, even considering its advantages in terms of explainability and modeling phenomena associated with degrees of uncertainties [10].

Most fusion studies in time series at different levels are developed to obtain good accuracy in classification task and there are few kinds of research that focus on the regression task. The last is important when data is distributed due to privacy, security, or ethical concerns and sharing data is not possible [56]. In other cases, it can be obtained a global model that allows predicting particular events at points that do not have enough data. Since few studies combine Neuro-Fuzzy models with fusion techniques for time series forecasting, in this work, a new structure combining a Neuro-Fuzzy model and a data fusion approach (based on the concept of stacking generalization) to improve time series forecasting is proposed. The novelty of this work focuses on the generation of a new Neuro-Fuzzy framework that allows the incoming and stacking of various sources of information to perform a regression task in time series.

As case study, we work with Rainfall-Runoff modeling phenomenon, a key tools in hydrological processes that represents a set of equations to predict the runoff of a watershed from rainfall, previous values of streamflow and, depending on the type of model, or other characteristics such as evapotranspiration and watershed variables (topography, drainage area or soil properties, among others) [57, 58]. Although there is a cause and effect relationship between both processes, rainfall-runoff modeling with high accuracy is affected by its non-linear behavior associated with the complex characteristics of the water cycle [59]. In Chile, conceptual methods have been mainly used to model this relationship, despite their limitations associated with the assumption of environmental standard conditions [60]. Besides, there are no studies aimed at data combination, even though many investigations suggest the use of more than one meteorological station as strategy to reduce the uncertainty in runoff forecasting. The framework developed in this work is our main contribution to professionals involved in the management of watershed resources to make appropriate decisions in environmental planning. Considering the aim of this work, the following objectives were proposed:

General

To improve the performance of time series forecasting by developing a new Stacking Neuro-Fuzzy framework from distributed data sources, in comparison with single approaches.

Specifics

1. To improve the performance of single time series forecasting by managing the uncertainty of selection of lags and hyperparameters in the regression task with a new Self-Identification Neuro-Fuzzy Inference Model.
2. To enhance time series forecasting with the development of a Self-Organizing Neuro-Fuzzy Inference System as stacking approach considering a distributed data sources.

3. To improve the performance of Runoff forecasting from distributed meteorological stations by applying the proposed Stacking Neuro-Fuzzy framework in comparison with single approaches.

Chapter 2

The state of art

2.1 Time series overview

2.1.1 Stochastic processes

A time series is a set of observations x_t, each one being recorded at a specific time t. The observations can be observed at a finite number of times, and in that case the underlying sequence of random variables $(X_1, X_2, \cdots, X_n)$ is just a n-dimensional random variable. Often, however, it is convenient to allow the number of observations to be infinite. In that case $\{X_t,\ t = 1, 2, \cdots\}$ is called a stochastic process [61].

Formally, a stochastic process is a family of indexed random variables

$$\{X(t,w);\ t \in T;\ w \in \Omega\} \tag{2.1}$$

defined on a probability space (Ω, β, P) where T is an arbitrary set. If T is a countable finite or infinite set, X is called a process discreet time; On the other hand, if T is an interval, X is called a process continuous time.

A stochastic process is a function of two variables defined by

$$X_t = X_t(w),\ w \in \Omega \tag{2.2}$$

On the other hand, for a random result, $w \in \Omega$ is a function of time

$$X_t = X_t(w),\ t \in T \tag{2.3}$$

2.1.2 Stationarity

A large number of stochastic processes have the property that their average statistical properties are independent of where they are formed along the time axis. Such stochastic processes are said to have various types of stationary properties. In other words, the statistical properties such as mean and variance of a stationary process do not depend upon time. It is a necessary condition for building a time series model that is useful for future forecasting.

A process $\{x(t),\ t = 0,1,2,\cdots\}$ is Strongly Stationary or Strictly Stationary if the joint probability distribution function of $\{x_{t-s}, x_{t-s+1}, \cdots, x_t, \cdots, x_{t+s-1}, x_{t+s}\}$ is independent of t for all s. Thus for a strong stationary process the joint distribution of any possible set of random variables from the process is independent of time [62].

A stochastic process is said to be Weakly Stationary of order k if the statistical moments of the process up to that order depend only on time differences and not upon the time of occurrences of the data being used to estimate the moments. For example a stochastic process $\{x(t), t = 0,1,2,\cdots\}$ is second order stationary if it has time independent mean and variance and the covariance values $Cov(x_t, x_{t-s})$ depend only on s [63].

2.1.3 Autoregressive Integrated Moving Average Model (ARIMA)

An ARIMA (p,d,q) model consists of three parts namely:

$$ARIMA(p,d,q) = AR(p)\ +\ I(d)\ +\ MA(q) \tag{2.4}$$

Where p is the order of autoregressive term, q is the order of moving average term and d is the number of times the series is differenced in order to make it stationary. If the series is already stationary, no differencing is required, thus $d = 0$ and it is known as ARMA model. $AR(p)$ represents that y_t, the value to be predicted depends on previous p values of y. $AR(p)$ is represented as:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + e_t \tag{2.5}$$

here y_t and e_t are respectively the actual value and random error (or random shock) at time period t, ϕ_i, $i = 1,2,\cdots,p$ are model parameters and c is a constant. The integer constant p is known as the order of the model. Sometimes the constant term is omitted for simplicity.

$MA(q)$ represents that y_t, the value to be predicted depends on previous q error terms. $MA(q)$ is represented as:

$$y_t = c + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{2.6}$$

Therefore, the equation of ARIMA model can be written as:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{2.7}$$

The coefficients of AR Model can be estimated using Ordinary Least Square (OLS), Maximum Likelihood Estimation (MLE) or Yule-Walker method. To estimate the coefficients of the MA model an ad-hoc method or innovation algorithm is used. The ad-hoc method consists of following steps:

1. Convert the $MA(q)$ to an $AR(\infty)$.
2. Cut off the $AR(\infty)$ term to some suitable order making it a finite term.
3. Use any of the three methods mentioned above to estimate AR parameters.
4. After finding the AR parameters, the MA parameters can be obtained from the relationship $ARMA(p,q) = AR(\infty)$ or $ARMA(p,q) = MA(\infty)$.

The ARIMA modeling procedures are determined by the Box-Jenkins model building methodology whose steps are:

1. Identifying the degree of differencing to transform the time series data into stationary.
2. Estimating the model parameters by autocorrelation function (ACF) and partial autocorrelation function (PACF).
3. Diagnostic checking which involves checking the degree of accuracy.

The first step of Box-Jenkins methodology is to make the data stationary. This can be done by removing the trend and seasonality present in the data. A test can also be carried out to check the stationarity known as Dickey-Fuller Test. The second step is to estimate the parameters of the model, this is, to estimate the values of p and q using ACF and PACF plot. PACF helps in determining the order of $AR(p)$ term and ACF in determining $MA(q)$ term. These p and q are used as initial values. Further, for improvement in the model the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values can be used. As the model with least AIC and BIC values is the best one. Thus, selecting the best model comes under the third step that is Diagnostic Checking, including also the diagnosis of errors and white noise. The limitation of ARIMA model is that it can remove the trend in the data by simply differencing it but cannot model the seasonality. Therefore, the SARIMA model was introduced, which extends the ARIMA model by adding the seasonality term to it [64].

Sometimes, external factors can affect time series forecasting, and, therefore, they should be included in the model prediction. The inclusion of one or more time series in a forecasting model can be done by the transfer function. ARIMAX is a Transfer Function Model which involves the inclusion of any other time series as an input variable in ARIMA. Assume two stationary time series denoted by Y_t and X_t, the transfer function model (TFM) is then represented as

$$Y_t = k + v(B)X_t + N_t \tag{2.8}$$

where Y_t is output variable, X_t is input variable, K is a constant, N_t is the noise at time t, B is backshift operator and $v(B)X_t$ is the transfer function which helps in conversion of input to output. Actually, this conversion of input is done with a linear filter. This conversion is not instantaneous but involves all values of the input in the form of summation with different weights v_i on x_t and, thus it can be expressed as

$$v(B)X_t = (v_0 + v_1 B + v_2 B^2 + \cdots)X_t \tag{2.9}$$

The weight can be either positive or negative. The weight v_i defines the degree of influence of exogenous variable on output variable.

2.1.4 Autoregressive Conditional Heteroskedasticity (ARCH) Model

Assume a univariate serially uncorrelated, zero mean process, u_t. The u_t is said to follow an Autoregressive Conditionally Heteroscedastic process of order v (or ARCH(v)) if the conditional distribution of u_t, given its past observations, $u_{t-1}, u_{t-2}, \cdots$, has zero mean and conditional variance

$$\sigma_t^2|t-1 = Var(u_t|\Omega_{t-1}) = E(u_t^2|\Omega_{t-1}) = \alpha_0 + \alpha_1 u_{t-1}^2 + \cdots + \alpha_v u_{t-v}^2 = \alpha_0 + \sum_{i=1}^{v} \alpha_i u_{t-i}^2 \tag{2.10}$$

where $u_t|\Omega_{t-1} \sim N(0, \sigma_t)$ and Ω_{t-1} is the conditional distribution function, σ_t^2 is the conditional variance and α is the model parameter. This ARCH process was first introduced by Engle [65] and allows the conditional variance change over time as a function of past observations.

The above ARCH model was then developed by Bollerslev [66] to the Generalized ARCH (GARCH) model by entering the lagged conditional variance. The GARCH model is then written as follows

$$\sigma_t^2|t-1 = \alpha_0 + \sum_{i=1}^{v} \alpha_i u_{t-i}^2 + \sum_{j=1}^{m} \beta_j \sigma_{t-j}^2 \tag{2.11}$$

where β_j is the model parameter to be estimated. For the above GARCH model, the higher the $\alpha+\beta$, the higher the intensity of persistence in volatility of the time series. It should also be noted that the ARCH parameter, α, and the GARCH parameter, β, indicate short run and long run persistence, respectively. There was also a limitation with GARCH model: It treats both the negative and positive variance with same magnitude alike. To remove this and improve the prediction, several extensions of GARCH were proposed such as EARCH or TGARCH, among others [64].

2.2 Fuzzy logic

Fuzzy logic is an analysis method developed to incorporate uncertainty into a decision model, considering reasoning that is approximate rather than precise [67]. This is, instead of precision and accuracy, fuzzy logic carries a level of tolerance for imprecision. Therefore, it is a useful tool to try to imitate the modes of reasoning of the human mind that are approximate instead of exact [68]. For a proper understanding, it is necessary to address certain concepts such as fuzzy measures, fuzzy sets, and linguistic variables.

2.2.1 Fuzzy Measures

A measure (classical measure) in a measurable space $(X, \mathscr{A})$, where $\mathscr{A}$ is a σ-algebra, is a map $\mu : \mathscr{A} \to [0, \infty]$ that satisfies the properties

1. $\mu(\emptyset) = 0$
2. σ-additivity: if $A_1, \cdots, A_n \in \mathscr{A}$ is a family of two or two disjoint sets, then $\mu(\bigcup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} \mu(A_n)$

The last property constitutes the main characteristic of the classic measurements. Although the σ-additivity can be very effective and convenient in certain applications like statistics and economics it can also be too inflexible and rigid in other contexts, such as the artificial intelligence, neural networks, image processing, among others, in which it is useful to define non-additive measures (fuzzy measures) [69]. Fuzzy measures are characterized by the weakening of the σ-additivity property of classical measurements, which can be replaced by a weaker condition, known as monotony, therefore, fuzzy measures are also called non-additive measures. A formal definition is as follows.

A fuzzy measure μ on the space X is a set function, $\mu : 2^{\mathrm{X}} \rightarrow [0,1]$. It map the subsets of X into the unit interval and has the following properties

1. $\mu(\emptyset) = 0$
2. $\mu(\mathrm{X}) = 1$
3. $\mu(\mathrm{A}) \geq \mu(\mathrm{B})$ if $\mathrm{A} \supseteq \mathrm{B}$ (Monotonicity)

The term fuzzy measure was introduced by Sugeno [70] and provides a very general structure for the representation of knowledge about any uncertain variable [71]. In probability theory an important idea is the probability of an event, where an event A is crisp or fuzzy subset of the domain of the uncertain variable.

Here, since the idea of measure is broader than that of probability, we use the less specific terminology of "anticipation" and speak of the anticipation of an event in an analogous manner to speaking of the probability of an event [72]. Let V be a variable taking its value in the space X. By using μ to express our knowledge about the value of the variable X we provide the following interpretation. For any subset A of X we have that $\mu(\mathrm{A})$ indicates our "anticipation" that the value of V lies in A. Then, $\mu(\emptyset) = 0$ reflects the fact that we do not anticipate finding the value of V in the null set. The property $\mu(\mathrm{X}) = 1$ reflects the fact that we completely anticipate finding the value of V in the set X. Finally the monotonicity of μ reflects the fact that we cannot be more confident of finding the value of V in the set A the any set that contains A. Here we shall use the expression V is μ to denote the situation where our knowledge about V is carried by the set measure μ.

2.2.2 Fuzzy sets

Since, a measure μ on a space X associates with each crisp subset of X a value $\mu(\mathrm{A})$ in the unit interval, the result is a set of classes called fuzzy set. Otherwise, let X be the universe of discourse and x be its elements. According to the classical set theory, crisp set A of X is defined by the characteristic function $f_{\mathrm{A}}(x)$ of set A

$$f_{\mathrm{A}}(x) : \mathrm{X} \rightarrow [0,1]$$

where

$$f_{\mathrm{A}}(x) = \begin{cases} 1 & \text{if } x \in \mathrm{A}, \\ 0 & \text{if } x \notin \mathrm{A} \end{cases}$$

Based on the fuzzy set theory, fuzzy set A of X is defined by its membership function $\mu_{\mathrm{A}}(x)$

$$\mu_{\mathrm{A}}(x) : \mathrm{X} \to [0, 1]$$

where

$$\mu_{\mathrm{A}}(x) = \begin{cases} 1 & \text{if } x \text{ is totally in A}, \\ 0 & \text{if } x \text{ is not in A}, \\ 0 < u < 1 & \text{if } x \text{ is partly in A} \end{cases}$$

For any element x of universe X, membership function $\mu_{\mathrm{A}}(x)$ equals the degree to which x is an element of set A. This degree represents the degree of membership, also known as the membership value of element x in set A. The most commonly used membership functions are triangular, trapezoidal, piecewise linear and Gaussian functions because they are easily prepared and computationally fast. The choice of membership functions is largely arbitrary because there is no theoretical justification for using one rather than another [73]. The number of membership functions is user-defined. More membership functions can achieve greater resolution but also cause greater computational complexity.

Fuzzy set operations are a generalization of crisp set operations, each of which is a fuzzy set operation. In fuzzy logic, three operations, including fuzzy complement, fuzzy intersection and fuzzy union, are the most commonly used. Let fuzzy sets A and B be described by their membership functions $\mu_{\mathrm{A}}(x)$ and $\mu_{\mathrm{B}}(x)$. The three fuzzy set operations are defined below.

- **Fuzzy complement:** The complement of a fuzzy set is the opposite of the set in question. The fuzzy complement of fuzzy sets A can be represented as

  $\mu_{\sim\mathrm{A}}(x) = 1 - \mu_{\mathrm{A}}(x)$

- **Fuzzy intersection:** Fuzzy intersection is the fuzzy operation for creating the intersection of fuzzy sets A and B on the universe of discourse X, which can be obtained as

  $\mu_{\mathrm{A}\bigcap\mathrm{B}}(x) = \min(\mu_{\mathrm{A}}(x), \mu_{\mathrm{B}}(x)) = \mu_{\mathrm{A}}(x) \bigcap \mu_{\mathrm{B}}(x), \text{where } x \in \mathrm{X}$

- **Fuzzy union:** The union of two fuzzy sets is the reverse of their intersection. That is, the fuzzy union is the largest membership value of the element in either set. The fuzzy union for forming the union of fuzzy sets A and B on the universe of discourse X can be given as

$\mu_{A \bigcup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \bigcup \mu_B(x)$, where $x \in X$

2.2.3 Linguistic Variables

A linguistic variable is a variable whose values are words or sentences in a natural or artificial language [38, 74]. For example, Age is a linguistic variable X if its values (term-set, $T(X)$ are assume to be "young", "not young", "very young", "more or less young", among others

$$T(\text{age}) = \{\text{"young", "old", "very young", "not young"}, \cdots\}$$

where each of the terms in $T(\text{age})$ is a label of a fuzzy subset of a universe of discourse, say $U = [0, 100]$.

A linguistic variable is associated with two rules:

1. A syntactic rule, which defines the well-formed sentences in $T(X)$;
2. A semantic rule, by which the meaning of the terms in $T(X)$ may be determined. If x is a term in $T(X)$, then its meaning (in a denotational sense) is a subset of $T(U)$. A primary term in $T(X)$ is a term whose meaning is a primary fuzzy set, that is, a term whose meaning must be defined a priori, and which serve as a basis for the computation of the meaning of the non-primary terms in $T(X)$.

For example, the primary terms in $T(\text{age})$ are "young" and "old", whose meaning must be defined by their respective membership function μ_{young} and μ_{old}. From these, the meaning of the non-primary terms may be computed by the application of a semantic rule such, for example, $\mu_{\text{very young}} = (\mu_{\text{young}})^2$ or $\mu_{\text{more or less old}} = (\mu_{\text{old}})^{1/2}$. For illustration, plots of the membership functions of these terms are shown in Figure 2.1.

2.2.4 Fuzzy If-Then rules and fuzzy systems

Fuzzy If-Then rules are expressions of the form IF A THEN B, where A and B are labels of fuzzy sets characterized by appropriate membership functions [11]. An example that describes a simple fact is

IF pressure is high THEN volume is small

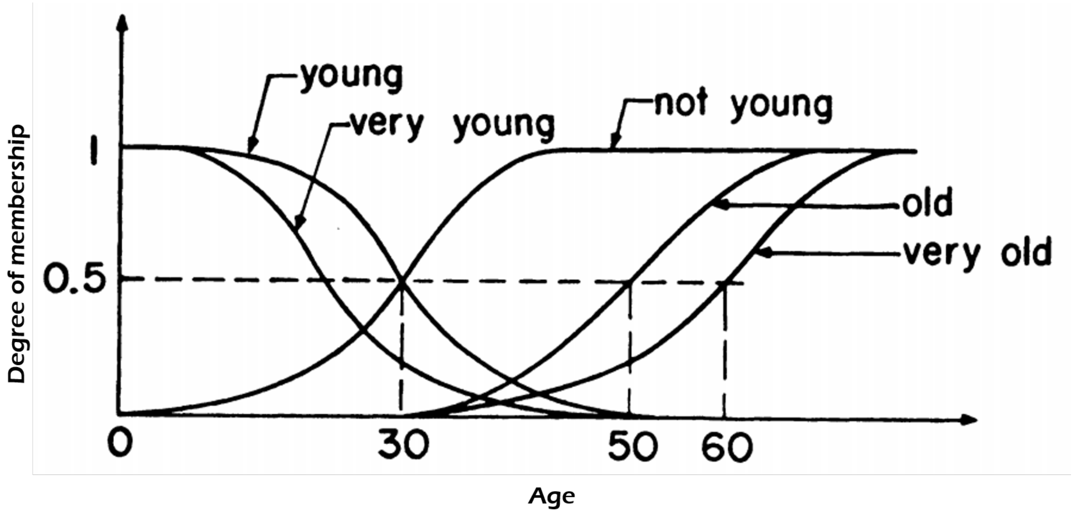


Fig. 2.1 Degrees of Membership for linguistic variable "Age". Zadeh [38].

where *pressure* and *volume* are linguistic variables and, *high* and *small* are linguistic values or labels that are characterized by membership functions.

Another form of fuzzy If-Then rule, proposed by Takagi and Sugeno [75], Moraga and Salas [76], has fuzzy sets involved only in the premise part. By using Takagi and Sugeno's fuzzy If-Then rule, we can describe the resistant force on a moving object as follows

$$\text{IF velocity is high THEN } force = k * (velocity)^2$$

where, again, *high* in the premise part is a linguistic label characterized by an appropriate membership functions. However, the consequent part is described by a non-fuzzy equation of the input variable, *velocity*.

Regardless of the form, fuzzy If-Then rules are the core part of the fuzzy systems. Basically a fuzzy system (or fuzzy model) is composed of five functional blocks (Figure 2.2):

- **Rule base**, containing a number of fuzzy If-Then rules;
- **Database**, which defines the membership functions.
- **Decision-making unit**, which performs the inference operations son the rules;
- **Fuzzification interface**, which transforms the crisp inputs into degrees of match with linguistic values;
- **Defuzzification interface**, which transform the fuzzy results of the inference into a crisp output.

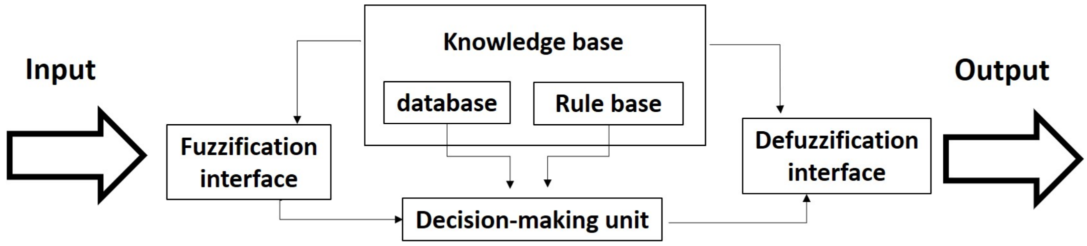


Fig. 2.2 Fuzzy system structure. Jang [11].

The steps of fuzzy reasoning (inference operations upon fuzzy If-Then rules) performed by fuzzy systems are:

1. **Fuzzification**: Compare the input variables with the membership functions on the premise part to obtain the membership values of each linguistic label.
2. **Combination**: Combine (through a specific T-norm operator) the membership values on the premise part to get firing strength (weight) of each rule.
3. **Generation**: Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength.
4. **Defuzzification**: aggregate the qualified consequents to produce a crisp output.

2.3 Neuro-fuzzy models

Neuro-fuzzy model is an architecture in which the neural network uses the training data to determine the membership functions and fuzzy rules of the fuzzy logic system [8, 76, 77]. One of the first and still one of the popular neuro-fuzzy systems is Jang's ANFIS model proposed in 1991 [11], and described below.

2.3.1 Adaptive Network-based Fuzzy Inference System

The fundamental axis of the Adaptive Network-based Fuzzy Inference System (ANFIS) (Figure 2.3) is to model the phenomenon based on the fuzzy Takagi-Sugeno rules of the form

$$R_r : \textbf{If } x_1 \textbf{ is } \mu_{j_1}^{(1)} \text{ and} \cdots \text{and } x_n \textbf{ is } \mu_{j_n}^{(n)} \textbf{ then } y = \beta_o^{(r)} + \beta_1^{(r)} x_1 + \cdots + \beta_n^{(r)} x_n. \tag{2.12}$$

where μ_{j_i} is the degree to which an input x_i satisfies a linguistic quantifier, being calculated through a certain membership function that involves a set of parameters called premises.

Depending on this degree of membership, the output y is calculated and its parameters β in the linear equation are called consequents.

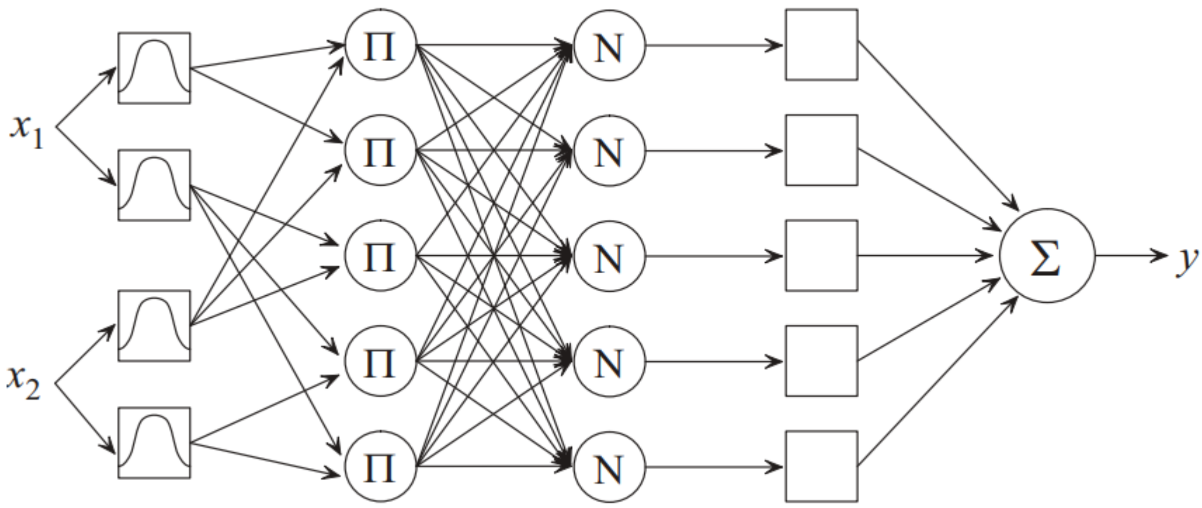


Fig. 2.3 ANFIS model architecture. Jang [11].

ANFIS structure has five layers. The nodes of layer 1 (*Fuzzification Layer*) compute the degree to which a given input x_i satisfies the linguistic quantifier $A_i^{(k)}$. The output of the node is given by the membership function $\mu_{A_i^{(k)}}(x_i)$. In this work we will use the gaussian-type membership function

$$\mu_{A_i^{(k)}}(x_i;\eta_i^{(k)}) = exp\left[-\left(\frac{x_i - v_i^{(k)}}{\sigma_i^{(k)}}\right)^2\right];\ i = 1,\cdots,n \tag{2.13}$$

where $\eta_i^{(k)} = \{v_i^{(k)}, \sigma_i^{(k)}\}$ are the premise parameters that should be estimated for the linguistic label $A_i^{(k)}$, the $v_i^{(k)}$ parameter stands for the location while $\sigma_i^{(k)}$ is the width of the membership function of the linguistic operator $A_i^{(k)}$.

The nodes of layer 2 (*Generalized "AND Layer"*) consist in *T-norm* operators that perform the generalized AND. Each node of this layer represents the firing strength of some specific rule. We will use the product *T-norm*, which is a gradient descend algorithm

$$w_k = w_k(\mathrm{x};\eta^{(k)}) = \mu_{A_1}^{(k)}(x_1,\eta_1^{(k)})\cdot\ldots\cdot\mu_{A_d}^{(k)}(x_1,\eta_d^{(k)}) \tag{2.14}$$

for $k = 1 \ldots K$.

The layer 3 (*Normalization layer*) computes the *normalizing firing strengths* of the weights of the previous layer

$$w_k = \bar{w}_k(\mathrm{x};\eta^{(k)}) = \frac{w_k}{\sum_{j=1}^{K} w_j} \tag{2.15}$$

for $k = 1...K$.

The nodes of layer 4 (*Consequent Layer*) compute the weighted hyperplane that approximates the nonlinear mapping, i.e.,

$$\bar{f}_k(\mathrm{x};\eta^{(k)},\Theta_k) = \bar{w}_k(\mathrm{x};\eta^{(k)})f_k(\mathrm{x};\Theta_k) = \bar{w}_k\Theta_k^{'}\mathrm{x} \tag{2.16}$$

where $\bar{w}_k$ is the output of the k-the node of layer 3 and $\Theta_k = (\Theta_1^{(k)},...,\Theta_d^{(k)},\Theta_{d+1}^{(k)})'$ is the consequent parameter.

And, The layer 5 (*Network output*) consists in a single node that computes the overall output as the summation of all the incoming signals

$$g(\mathrm{x};\eta,\Theta) = \sum_{k=1}^{K} \bar{w}_k(\mathrm{x};\eta^{(k)})f_k(\mathrm{x};\Theta_k) \tag{2.17}$$

where $\eta = (\eta_1^{(1)},...,\eta_d^{(K)})'$ and $\Theta = (\Theta_1,...,\Theta_K)'$ correspond to the premise and consequent set of parameters respectively.

To estimate the parameters, the classical ANFIS employs an hybrid learning that uses backpropagation learning algorithm to determine premise parameters η and the Least Mean Square estimation procedure to determine the consequent parameters $\Theta_k, k = 1,\cdots,K$, the premise parameters remains fixed. The updating rule is given by

$$\Theta_k(t+1) = (A'\cdot A)^{-1}\cdot A'\cdot B \tag{2.18}$$

where $A = W_k * X$ and $B = (Y - g(\mathrm{X};\eta,\Theta)) + (A*\Theta_k(t))' * W_k$ is a diagonal matrix where the k-th element of the diagonal is $\bar{w}_k$, and X is a matrix where each row corresponds to a data sample x, Y is a vector containing the output training sample and $g(\mathrm{X};\eta,\Theta)$ is the output of the ANFIS for all training data. The premise parameters $\eta_i^{(k)} = v_i^{(k)}, \sigma_i^{(k)}$ are estimated iteratively by the following updating rules

$$v_i^{(k)}(t+1) = v_i^{(k)}(t) + 4\alpha(t)\frac{1}{(\sigma_i^{(k)})^2}(x_i - v_i^{(k)})\bar{w}_k(f_k - g)(y-g) \tag{2.19}$$

$$\sigma_i^{(k)}(t+1) = \sigma_i^{(k)}(t) + 4\alpha(t)\frac{1}{(\sigma_i^{(k)})^3}(x_i - v_i^{(k)})^2\bar{w}_k(f_k - g)(y-g) \tag{2.20}$$

where the consequents parameters remains fixed and $\alpha(t)$ is the learning rate function.

ANFIS does not learn the structure of the fuzzy system, but it simply creates rules from all possible combinations of input fuzzy sets. Initial fuzzy partitions have to be specified and the consequent parameters are initialised by small random numbers. This model can be use both classification and forecasting time series and there are a lot of models based on it

2.3.2 Self-Organizing Neuro-Fuzzy Inference System

Self-Organization Neuro-Fuzzy Inference System (SONFIS) was proposed by [21, 22], and its basic structure has the same components of the ANFIS system. In other words, SONFIS, like the ANFIS model, has five layers in its neural network (Figure 2.4). Layer 1 associates each input to a certain fuzzy set through a Gaussian-type membership function. In layer 2, the T-norm *AND* operator is used to obtain the strength of each of the rules specified in layer 1, which will later be normalized in layer 3. The layer 4 computes the contribution of each rule using both consequent parameters and layer 3 output vector. Finally, the layer 5 computes the weighted global output of all the incoming signals.

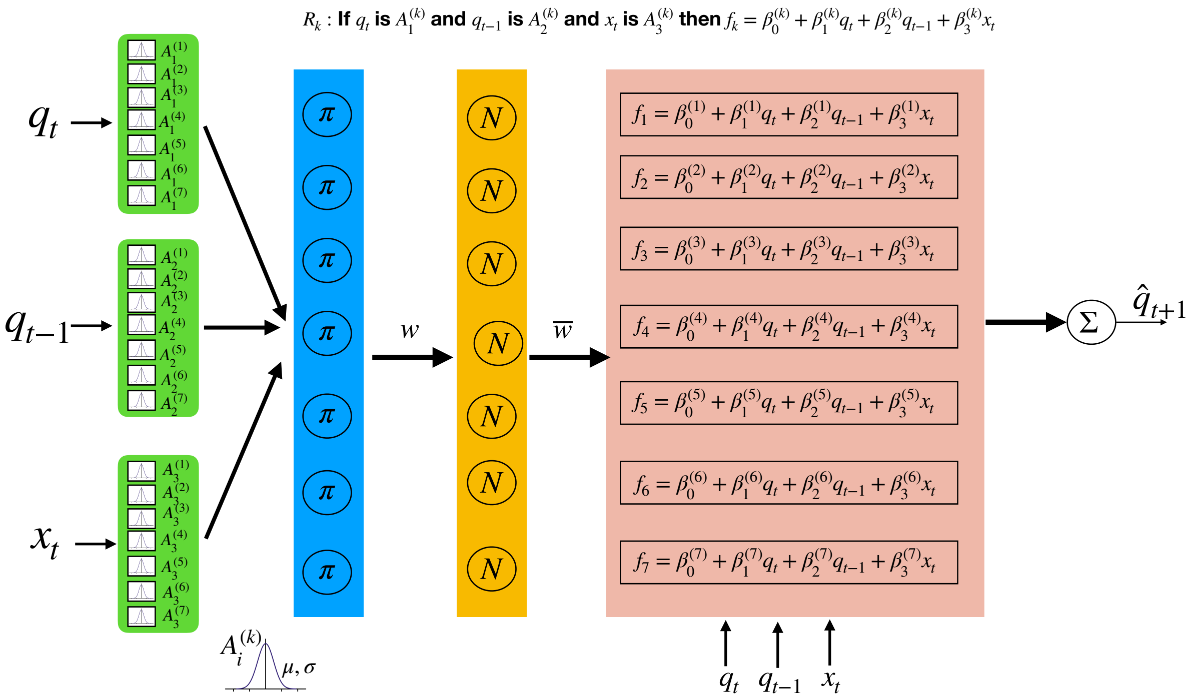


Fig. 2.4 SONFIS model. Its structure involves five layers of neurons and can automatically identifying the number of rules necessary to model the phenomenon during learning phase, being able to create or vanish new rules.

In addition to using the hybrid learning algorithm of the ANFIS model (namely, back-propagation algorithm to estimate the premise parameters and ordinary least squares method to obtain the consequent parameters), SONFIS uses iteratively a series of mathematical operators to automatically identify the number of rules required to model the dataset: *GrowNet*, *Split Membership Functions* and *Vanish Membership Functions* (Figure 2.5).

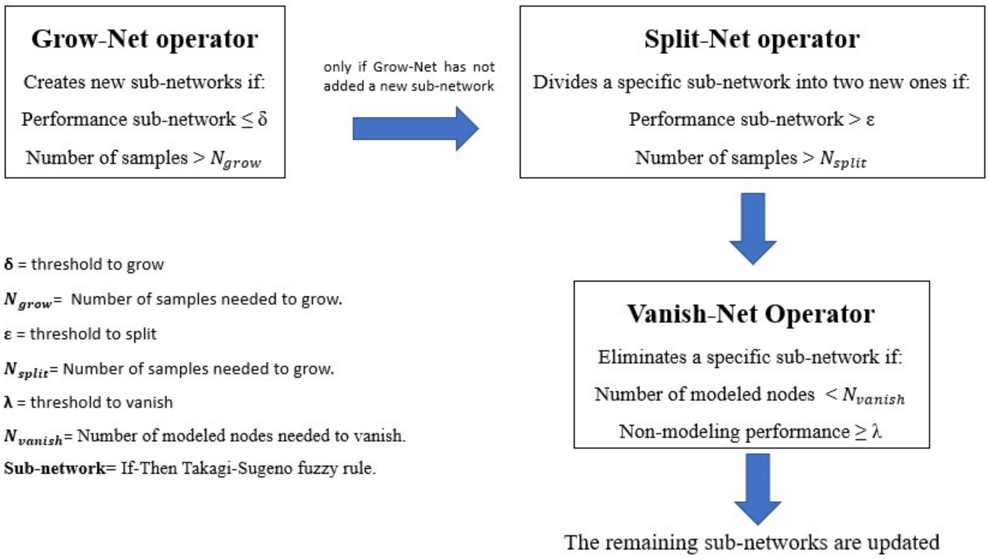


Fig. 2.5 Mathematical operators used in SONFIS model. They are applied iteratively to identify the number of fuzzy rules required to model the training data-set.

Before applying any operator, the current base model is frozen meaning that none of the parameters can be any longer update. SONFIS model can add, split or vanish a fuzzy if-then rule that has the following elements: a set of nodes of the first layer that are Gaussian-type functions, one node for each dimension; a node of the second layer that computes the product of the incoming values; a normalization node of the third layer; the weighted regression line modeled by the node of the fourth layer; and all the incoming and outgoing links of aforementioned nodes.

The SONFIS algorithm works as follows: At first the model must be initialized. The initialization consists in starting from a predefined number of nodes (if there is no prior knowledge the default is 1) and setting the user-defined thresholds. This step will provide an initial SONFIS architecture. After that, the algorithm will proceed with the iterations to establish a final parsimonious architecture.

In each iteration the current architecture must be frozen, so that the current nodes remain stable. The first operator to be evaluated is the GrowNet operator. This operator decides if the current number of nodes are having a good performance with all the training data. If not, this operator creates new subnetworks (if-then Takagi-Sugeno Fuzzy rules). The next step evaluates the SplitNet operator only if GrowNet has not added a new node. This operator splits a specific sub-network into 2 if it has a poor performance based on a defined threshold and if it does not model a minimum of data points.

The following step evaluates the VanishNet operator. This operator deletes a sub-network that has a poor performance or is modeling not a sufficient amount of data points. This step is crucial because it may eliminate nodes that could have been created by the other 2 operators and that are not making a relevant contribution to the performance of the model. After that, the next step adjusts the membership functions and hyperplane parameters of the sub-networks that the operators have created. The iterations stop when the operators fail to create new sub-networks. Finally the algorithm unfreezes the parameters of all the nodes of the SONFIS architecture and proceeds to adjust the premise and consequent parameters of all the nodes.

2.3.3 Other approaches

Several authors have proposed several approaches based on the neuro-fuzzy system. Some of them are:

- **FuNe-I**. This model, proposed in 1992, is based on the architecture of a feed-forward neural network [78]. Compared to other neuro-fuzzy approaches, the FuNe-I model is special because it uses three kinds of rules: the antecedents can be conjunctions or disjunctions, and there are rules with only one variable as antecedent (simple rules). A unit computes its activation, depending on the kind of rule it represents, by either a differentiable soft minimum, a differentiable soft maximum, or the identity function. This model The FuNe-I model provides algorithms for structure and parameter learning and is one of the first neuro-fuzzy approaches that also considers rule learning.

- **NEFCLASS**. Proposed in 1995, it is one of the firsts neuro-fuzzy approach that was able to handle missing values, both numeric and symbolic data in the same data set and to determine a rule-base fully automatically [79]. NEFCLASS is also based on the idea of a generic fuzzy perceptron and focuses on creating small interpretable fuzzy rule bases. In parameter learning the fuzzy sets are tuned by a simple backpropagation-like procedure that is based on a simple heuristics instead of a gradient descent approach. After learning NEFCLASS uses pruning strategies to reduce the number of rules as much as possible.

- **Fuzzy RuleNet**. This model is based on the structure of an radial basis function (RBF) network, but instead of the usual radial basis functions, which represent hyperellipsoids, RuleNet uses hyperboxes for classification. Fuzzy RuleNet allows hyperboxes to overlap. Each hyperbox represents a multidimensional fuzzy set given by a membership function in form of a hyperpyramid. By projecting the multidimensional fuzzy sets onto the individual dimensions we obtain triangular or trapezoidal fuzzy sets that describe the pattern features. The fuzzy classification rules obtained this way are equivalent to the multidimensional fuzzy sets, i.e. there is no loss of information as it would be in the case of hyperellipsoids used in fuzzy cluster analysis [80].

- **FALCON**. The Fuzzy Adaptive Learning Control Network (FALCON) is an architecture of five layers [81]. There are two linguistics nodes for each output. One is for the patterns and the other is for the real output of the FALCON. The first hidden layer is responsible for the mapping of the input variables relatively to each membership functions. The second hidden layer defines the antecedents of the rules followed by the consequents in the third hidden layer. FALCON uses an hybrid learning algorithm composed by a unsupervised learning to define the initial membership functions and initial rule base and it uses a learning algorithm based on the gradient descent to optimise/adjust the final parameters of the membership functions to produce the desired output.

- **NEFCON**. The Neural Fuzzy Controller (NEFCON) was drawn to implement a Mamdani type inference fuzzy system [82]. The connections in this architecture are weighted with fuzzy sets and rules using the same antecedents (called shared weights), which are represented by the drawn ellipses. They assure the integrity of the base of rules. The input units assume the function of fuzzyfication interface, the logical interface is represented by the propagation function and the output unit is responsible for the defuzzyfication interface. The process of learning in architecture NEFCON is based in a mixture of reinforcement learning with backpropagation algorithm. It has two variants NEFPROX (for function approximation) and NEFCLASS (for classification tasks) [83].

- **EFuNN**. In Evolving Neural Fuzzy Network EFuNN all nodes are created during the learning phase. The first layer passes data to the second layer that calculates the degrees of compatibility in relation to the predefined membership functions. The third layer contains fuzzy rule nodes representing prototypes of input- output data as an association of hyper-spheres from the fuzzy input and fuzzy output spaces. Each rule node is defined by two vectors of connection weights, which are adjusted through a hybrid learning technique. The fourth layer calculates the degree to which output membership functions are matched the input data and the fifth layer carries out the defuzzyfication and calculates the numerical value for the output variable.

2.4 Multimodal fusion in time series

Data fusion is a combination of multiple sources to obtain improved information; in this context, improved information means less expensive, higher quality, or more relevant information [24].

In time series, there are some models based on fusion techniques. To obtain more information on extracted features, Diao et al. [26] proposed a data fusion algorithm of multivariate time series with application in electroencephalographic signals. This framework was based on the local weighted linear prediction algorithm. On the other hand, Hsiao et al. [28] proposed a temporal data fusion in multisensor systems usign Dynamic time Warping (DTW), which is a general time alignment and a measure of similarity scheme for two time sequences. Guo et al. [27] proposed a feature fusion based forecasting model for financial time series. In this architecture, independent component analysis (ICA) and its multivariate framework (MICA) were used to extract two feature sets of time series, then both were fused with a framework adopting the idea of canonical correlation analysis, before applying SVR to forecast a future value in Shanghai stock market index and the Dow Jones index.

The previous models are based on merging the input data or the features extracted from the time series to make the prediction. But, in cases where the time series of different systems cannot be combined, there is another approach to the merger. Distributed regression is a suite of methods that enable researchers to conduct multidatabase regression analysis without the need to centrally combine all individual-level data from participating sites. It performs the same numeric algorithm as standard regression that is based on individual-level data but uses only summary statistics for computation. By following the same computation process, distributed regression and pooled individual level data analysis should theoretically produce statistically equivalent results [56].

Several researchers treat distributed learning systems as a centralized ensemble-based method, where a new data point is predicted from the predictions of all local sites using ensemble methods such as stacking, boosting, majority voting, simple average, or winner-takes-all methods [84]. In 2013, Allende-Cid et al. [85] present a distributed regression approach able to detect different contexts in the input space, thus improving the performance of local models in the task of regression from distributed sources. On the other hand, Since the average process of the typical distributed regression will decrease the variance, not the bias, bias correction is expected to improve the learning performance if the base regression algorithm is a biased one. Based on this, Guo et al. [86] proposed a bias corrected version of regularization kernel network.

Stacked generalization is an ensemble method that allows researchers to combine several different prediction algorithms into one [87]. It was introduced by Wolpert [88], and represents a combination of the base learners using another learner, usually referred to as the combiner learner. In the context of probabilistic forecasts, ensemble learning stands for the use of multiple ML algorithms to obtain individual probabilistic forecasts, and their subsequent combination (through a combiner learner) to obtain prediction intervals [31]. As stacking method, several schemes have been applied, including approaches such as average, median, selection of the best initial results, multiple linear regression or ANN, among others [30, 33, 34, 35, 36, 37].

Other stacking approaches used are the fuzzy integrals. Such is case fuzzy integrals Choquet and Sugeno [29]. The Choquet Integral is a generalization form of an integral using fuzzy measures. In a discrete setting, if $g : 2^N \to [0,1]$ (where $N = 1,...,n$) is a fuzzy measure (a set-valued increasing function such that $g(\emptyset) = 0$ and $g(N) = 1$, the Choquet integral is an aggregate function $C_g : [0,1]^n \to [0,1]$ defined, for each $x = (x_1,...,x_n) \in [0,1]^n$ and each membership function h, as:

$$C_g(h) = \sum_{i=1}^{n} (h(x_{(i)}) - h(x_{(i-1)})) \cdot g(A_{(i)}), \tag{2.21}$$

where $0 \leq h(x_{(1)}) \leq ... \leq h(x_{(n)})$, with the convention that $h(x_{(0)}) = 0$, and $A_{(i)} = (i),...,(n)$ is the subset of indices of $n-i+1$ largest components of $(h(x_{x_1}),...,h(x_n))$.

On the other hand, the Sugeno integral over the set $A = x_1,...,x_n$ of a membership function h with respect to the fuzzy measure (confidence) g is defined as

$$S_g(h) = \sup_{\alpha \in [0,1]} [\min(\alpha, g(A \cap F_\alpha))], \tag{2.22}$$

where $F_\alpha = x | h(x) \geq \alpha$.

Chapter 3

A Stacking Neuro-Fuzzy framework for time series forecasting from distributed data sources

In this chapter the proposed framework for time series forecasting from distributed data source is presented (Figure 3.1).

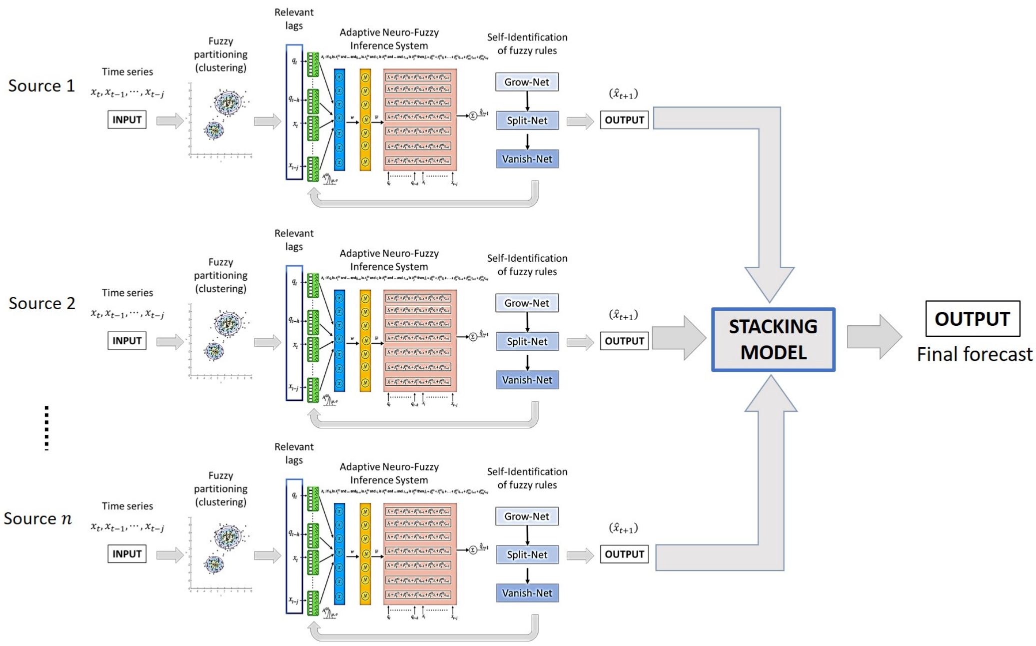


Fig. 3.1 Stacking model scheme. This framework involves the SINFIM method as single time series forecasting and the Self-Organizing Inference system as a stacking approach to provide the final output.

3.1 Single time series forecasting using a Self- Identification Neuro-Fuzzy Inference Model (SINFIM)

Self-Identification Neuro-Fuzzy Inference Model (SINFIM) is presented as method to single time series forecasting. This framework reduces the uncertainty associated with the lags and hyperparameters selection in a fuzzy system (Figure 3.2).

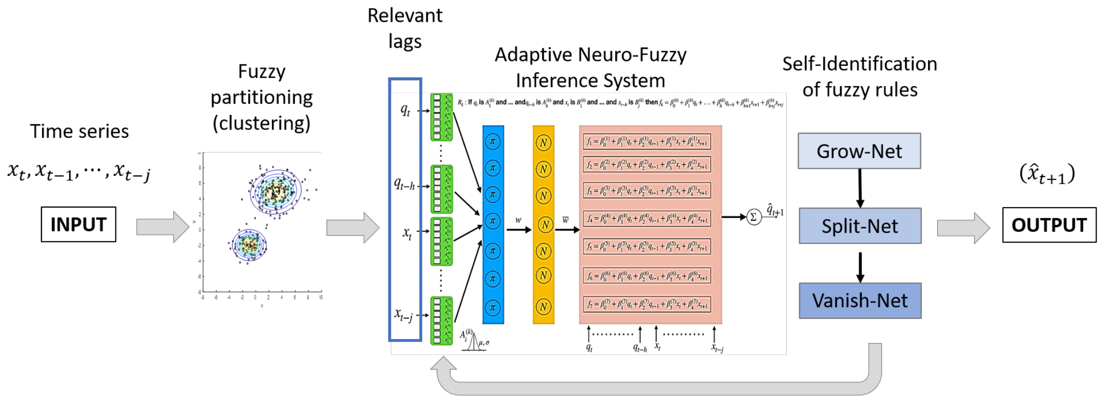


Fig. 3.2 Overview of the proposed framework to forecast day-ahead time series.

The proposed algorithms involve the following components:

1. **Self-Identification of the lags of time series**: using a model based on a fuzzy partitioning or clustering of each input (uni-dimensional time series), the more relevant lags of the time series to predict the day-ahead value are selected.
2. **Adaptive Neuro-Fuzzy Inference System**: Relevant lags selected in the previous step, are the inputs for an Adaptive Neuro-Fuzzy inference system, which involves a fuzzy model with a hybrid learning algorithm and a predefined number of membership functions. This model serves as the fundamental structure to predict the day-ahead value.
3. **Self-Identification of fuzzy rules**: Once the base Neuro-Fuzzy model is obtained, a series of mathematical operators are iteratively applied in order to identify the number of fuzzy rules required to model the phenomenon. Then, the consequent parameters are updated and, finally, the output is predicted.

3.1.1 Self-Identification of lags

The proposed framework begins with the identification of the most influential lags for time series. For this, a model based on the component of SIFAR method proposed by Veloz et al. [89] associated with lags relevance evaluation is developed. This model represents a set of local predictors in order to define different sub-domains along the lag space. The process is summarized in Figure 3.3.

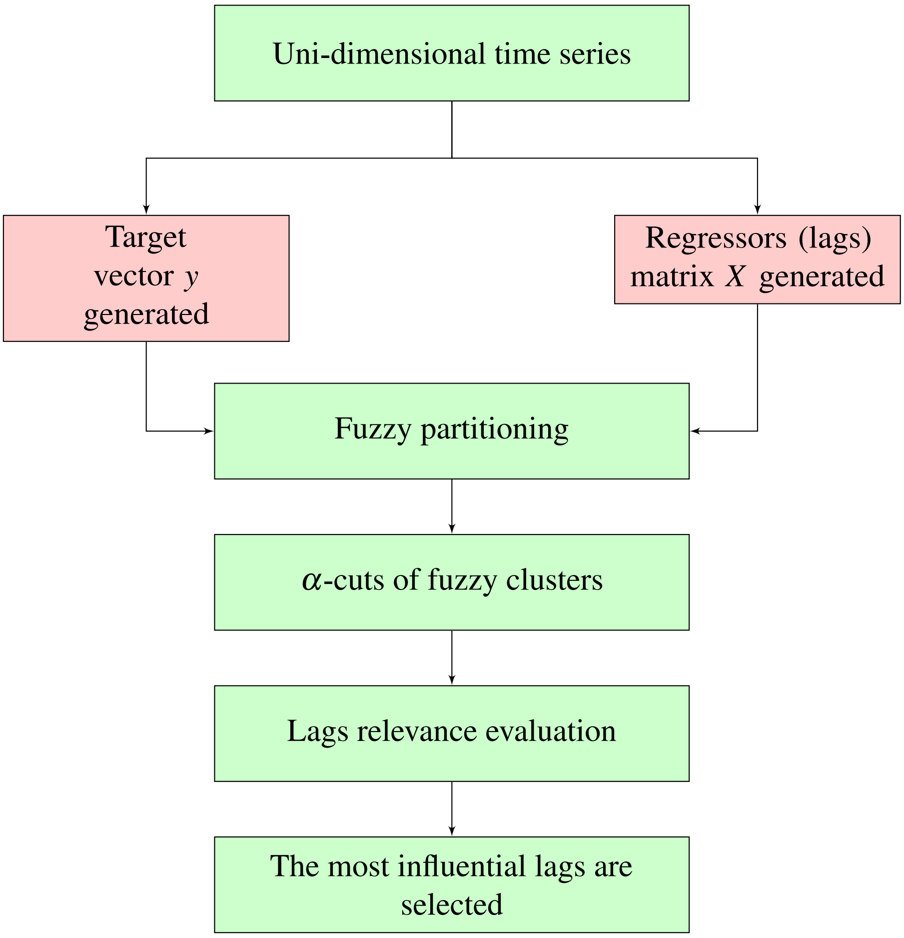


Fig. 3.3 Schematic representation of selecting relevant lags process of time series.

As described above, the time series $x(t)$ is a stochastic process whose samples are observed in regular time intervals ranging from $t = 1, \cdots, T$. For selection of lags, SIFAR model used a nonlinear autoregressive relationship $x(t) = f(x(t-1), \cdots, x(t-d))$ where d is the maximum expected order of the time series model and is user-defined. For notation purposes, a time series is arranged in a regressor matrix $\mathbf{X} = [x_{kj}]_{N\,x\,d}$, with $N = T - d$, and a target vector $\mathbf{Y} = (x(d+1), \cdots, x(T))'$. The j-th column of $\mathbf{X}$ is given by $(x(d+1-$

$j), \cdots, x(T-j))'$. The k-th row of the matrix $\mathbf{X}$ and the k-th element of the vector $\mathbf{y}$ are denoted as $\mathbf{x}_k$ and y_k, respectively.

The first step is to apply fuzzy C-means (FCM) in the target space $\mathbf{Y}$ using the entire dataset, generating a local domain and considering only those values belonging to this region with a high degree of membership $\mathbf{X}$. The number of partitions in both spaces is defined in advance and $\mathbf{V} = \{v_1, ..., v_{n_0}\}$ represents the cluster prototypes, where n_0 is the number of clusters in which the target space is partitioned. After that, the possibility distribution is computed for the h_{th} cluster as follows

$$t_{hk} = \frac{1}{\sum_{p=1}^{N}\left(\frac{|y_k - v_h|}{|y_k - v_p|}\right)^{\frac{2}{\eta-1}}}, \tag{3.1}$$

with $1 \leq h \leq n_o$ and $1 \leq k \leq N$. y_k represents the k-th element of the target vector y, while η is user-defined, usually set to 2.

The third step computes scale parameters $\{\sigma_1, ..., \sigma_{n_0}\}$ using the diagonal elements of the fuzzy covariance matrix

$$\sigma_h^2 = \frac{\sum_{k=1}^{N}(t_{hk})^m (y_k - v_h)^2}{\sum_{k=1}^{N}(t_{hk})^m}. \tag{3.2}$$

Afterwards, for each obtained cluster, an α-cut set $\{O_\alpha^{(1)}, ..., O_\alpha^{(h)}, ..., O_\alpha^{(n_0)}\}$ is computed, where $O_\alpha^{(h)} = \{y_k \in \mathbf{Y} | \mu_h(y_k) \geq \alpha_0\}$ constitutes the most representative and similar data samples. For the h-th α-cut $O_\alpha^{(h)}$ FCM is applied to the dataset $\{\mathbf{x}_{k*}\}$ where, $k* = k | y_k \in O_\alpha^{(h)}$.

The following step consists of applying the initial procedures to each α-cut generated in the regressor space, that is, applying the FCM technique and obtaining the cluster centers, in addition of calculating their possibility distribution and scale parameters. The set of cluster prototypes obtained are denoted by $\mathbf{V}^{(h)} = \{\mathbf{v}_1^{(h)}, ..., \mathbf{v}_{n_{in}}^{(h)}\}$ where $\mathbf{v}_i^{(h)} = [v_{ij}^{(h)}]$ corresponds to the i-th cluster prototype vector associated with the h-th α-cut obtained in the target space and $v_{ij}^{(h)}$ represents its element associated with the j-th lag. Through

$$t_{ik}^{(h)} = \frac{1}{\sum_{p=1}^{N}\left(\frac{||\mathbf{x}_k - \mathbf{v}_i^{(h)}||}{||\mathbf{x}_k - \mathbf{v}_p^{(h)}||}\right)^{\frac{2}{\eta-1}}}\mathbf{I}_{k*}(k) \tag{3.3}$$

the possibility distribution is computed, where $\mathbf{I}(\cdot)$ is the indicator function. Based on it, the scale parameters (σ_{ij}) are computed using the equation

$$\sigma_{ij}^{(h)} = \frac{\sum_{k \in k*}((t_{ik})^{(h)})^m (x_{kj} - (v_{ij})^{(h)})^2}{\sum_{k \in k*}((t_{ik})^{(h)})^m}, \tag{3.4}$$

where x_{kj} corresponds to the j-th lag variable of the k-th element of the process and $v_{ij}^{(h)}$ are the set of cluster centers.

The next step consists in evaluating the contribution to local smooth mapping between the regressor and target spaces of each lag with the following equation:

$$R_h^j = |k_{in}^{(h)}\ k_0^{(h)}|, \tag{3.5}$$

where $k_0^{(h)} = \{k|\ y_k \in O_\alpha^{(h)}\}$, $k_{in}^{(h)} = \{k|\ \mathbf{x}_k \in I_\alpha^{(h)}\}$, $I_\alpha^{(h)} = \bigcup_{i=1}^{n_{in}} \{\mathbf{x}_k \in \mathbf{X}|\ y_k \in O_\alpha^{(h)}; w_i^{(h)}(\mathbf{x}_k) \geq \alpha_{in}\}$, $|\cdots|$ constitutes the cardinality of the resulting set and $w_i^{(h)}$ the strength level for the i-th cluster in the regressor space.

The set of lags is ordered according to R_j, $j = 1,..,d$ results, and the first q elements are taken. This selection could involve redundant partitions which later will be removed based on the similarity of μ_{ij} membership functions. Two clusters will be merged if the result of the supremum minimum composition $s^{(j)}(i_1,i_2) = \sup\min(\mu_{i1j},\mu_{i2j})$, with i_1,$i_2 \in \{1,...,n\}$ and $i_1 \neq i_2$, is greater than or equal to a threshold α_f defined by the user. In this step, and after applying the entire the algorithm for each uni-dimensional time series considered, the relevant and non-redundant lags are selected.

3.1.2 Adaptive Neuro-Fuzzy Inference System

Once the most relevant lags are selected, they constitute of a Neuro-Fuzzy model to predict one day-ahead events of time series. For this, a model based on the Adaptive Neuro-Fuzzy Inference System (ANFIS) proposed in 1993 by Jang [11] is developed. It is a special feedforward neural network and its fundamental axis is to model the phenomenon based on the fuzzy Takagi-Sugeno-Kang rules of the form:

$$R_r : \textbf{If } x_1 \textbf{ is } \mu_{j_1}^{(1)} \text{ and} \cdots \text{and } x_n \textbf{ is } \mu_{j_n}^{(n)} \textbf{ then } y = \beta_o^{(r)} + \beta_1^{(r)} x_1 + \cdots + \beta_n^{(r)} x_n, \tag{3.6}$$

where μ_{j_i} is the degree to which an input x_i satisfies a linguistic quantifier, calculated using a certain membership function that involves a set of parameters η called premises. Depending on this degree of membership, the output y is determined and its parameter vector Θ (containing the parameters β in the linear equation) is called consequent.

The network implies five components (layers): Layer 1 associates each input to a certain fuzzy set through a Gaussian-type membership function. In layer 2, the *AND* T-norm operator is used to obtain the strength of each of the rules specified in layer 1, which will later be normalized in layer 3. The layer 4 computes the contribution of each rule using both consequent parameters and layer 3 output vector. Finally, the layer 5 computes the weighted global output of all the incoming signals as it was showed in Figure 2.3.

Once the base architecture is constructed with a predefined number of nodes for each layers, a hybrid learning is developed. This is, an Ordinary Least Square (OLS) estimation is used to determine the consequent parameters and the back-propagation learning algorithm to estimate the premise parameters.

3.1.3 Self-Identification of fuzzy rules

After the ANFIS hybrid learning is developed, a series of mathematical operators (proposed by Allende-Cid et al. [21]) are iteratively applied in order to automatically identify the number of rules required to model the dataset: *GrowNet*, *Split Membership Functions* and *Vanish Membership Functions* (Figure 3.4). Grow-Net is the first operator to run and it evaluates if the current fuzzy rules (sub-networks) work well according to the user predetermined threshold δ. This comparison is based on the firing strength w_k of all sub-networks in the layer 2 of the ANFIS model for each input $\mathbf{x}$ with dimension d, where the maximum should be greater than threshold δ:

$$\max_{k=1...K} w_k > \delta^d, \tag{3.7}$$

otherwise, the sample $(\mathbf{x}, y)$ is grouped into a set ϑ_k and from this, the operator generates a new sub-network for each dimension $\mu_{A_i^{K+1}}(x_i; \eta_i^{(K+1)}), i = 1, ..., d$ in order to increase the granularity of the partition of the feature space. This new sub-network is created only for each group that has a number of samples more than the user defined N_{grow} and the premise parameters are initialized with the mean and standard deviation of the samples from the group, while the consequent parameters are randomly initialized.

In the case that Grow-Net did not add new rules, Split-Net divides into two new ones those sub-networks with poor performance based on a predetermined threshold ε, which represents the mean square error:

$$E_k = \frac{1}{N_k} \sum_{(\mathbf{x},y)\in\vartheta_k} (y - g(\mathbf{x}; \eta, \Theta))^2, \tag{3.8}$$

where N_k represents the sample size of ϑ_k and $g(\mathbf{x}; \eta, \Theta)$ corresponds to the output of layer 5 of ANFIS model. To split a sub-network, Grow-Net requires a minimum number of samples N_{split} (predefined by user) and a mean square error greater than ε. Then, the premise parameters are divided and the consequent parameters are randomly initialized.

Finally, the VanishNet operator deletes a sub-network that has a poor performance or is modeling not a sufficient amount of data points (according to a threshold λ, user-defined). This makes possible to eliminate networks created in the previous steps whose contribution to the model are not significant. This operators works when variable age_k (an auxiliary variable that increases by one if the sub-network models no data) reached the hyper-parameter λ. The operators are executed iteratively and this iteration stops when new sub-network is not created. After that, the next step adjusts the premise and consequent parameters of the remaining sub-networks. A summary of these operators is showed in Figure 3.4.

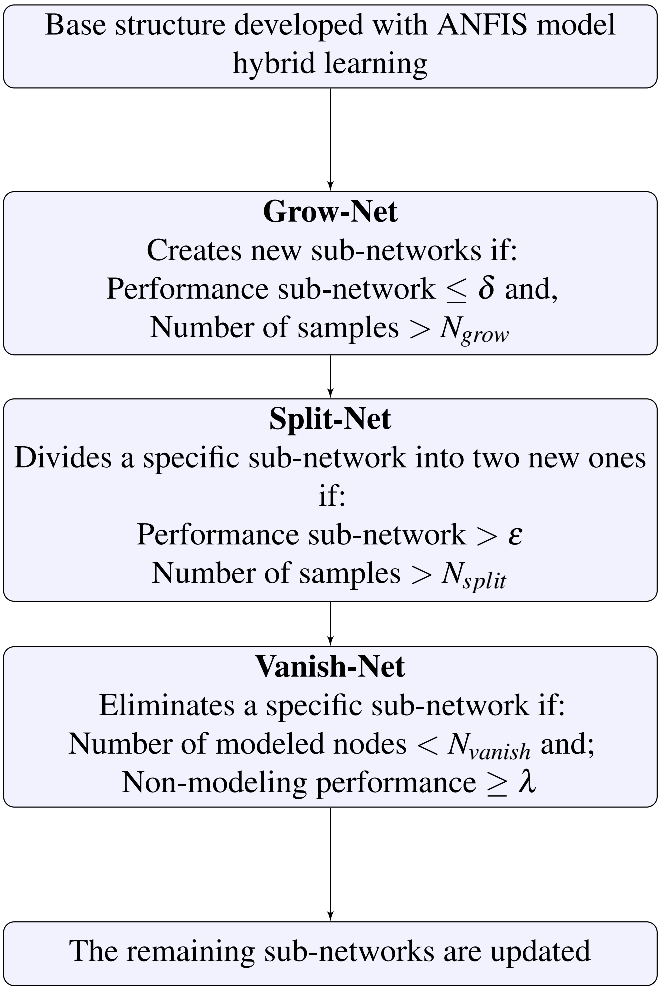


Fig. 3.4 Mathematical operators applied iteratively to identify the number of fuzzy rules required to model the training data-set.

3.2 Stacking of single time series forecasting

Once the single forecasts have been obtained from each data source, they are combined to obtain a final prediction considering the concept of stacked regression (Figure 3.1). For this module a model based on the proposal of Breiman [87] was developed.

Suppose one has available K predictors $v_1(x), \cdots, v_K(x)$ of a numerical outcome variable y in terms of a vector x. Assume these were constructed using the same learning set $D = (y_n, x_n),\ n = 1, \cdots, N$ where each x_n is an input vector. Instead of selecting the single best predictor among the $v_K(x)$, a more accurate predictor can be gotten by combining all of them in a new model (second level). This new level represents a linear combination of different predictors and it can be expressed as follows:

$$v(x) = \sum_{K=1}^{N} \alpha_K v_K(x) \tag{3.9}$$

Given a learning set $D = (y_n, x_n),\ n = 1, \cdots, N$ one possibility is to take the α_K to minimize

$$\sum_n (y_n - \sum_K \alpha_K v_K(x_n))^2 \tag{3.10}$$

under the constraints $\alpha_K \geq 0,\ K = 1, \cdots, N$. This resulting predictor appears to have lower prediction error than the single prediction v_K.

Although several models have been used to estimate the α_K coefficients that minimize the estimation error, such as linear regression or ridge regression, in this work the SONFIS model (presented by Allende-Cid et al. [21]) is proposed as stacking method. This is based on the advantage of using a Neuro-Fuzzy method to model phenomena with a certain degree of uncertainty. Thus, the SONFIS model is used to combined the v_K predictors based on the fuzzy Takagi-Sugeno-Kang rules of the form:

$$R_r : \textbf{If } v_1 \textbf{ is } \mu_{j_1}^{(1)} \text{ and} \cdots \text{and } v_n \textbf{ is } \mu_{j_n}^{(n)} \textbf{ then } y = \alpha_o^{(r)} + \alpha_1^{(r)} v_1 + \cdots + \alpha_n^{(r)} v_n, \tag{3.11}$$

where μ_{j_i} is the degree to which an input v_i satisfies a linguistic quantifier, calculated using a certain membership function. Depending on this degree of membership, the output y and its parameter vector Θ (containing the parameters α in the linear equation) are determined. As in the case of the SINFIM method, the mathematical operators proposed by Allende-Cid et al. [21] are iteratively applied in order to automatically identify the number of rules required to model the dataset. Therefore, With this stacking phase, the final forecast of the time series would be obtained, thus having a lower prediction error than the single forecasts.

Chapter 4

Application of the proposed Stacking Neuro-Fuzzy framework: The case of Runoff forecasting from distributed meteorological stations.

Hydrological models are key tools for water and environment resource management. They represent hydrological processes and evaluate conditions for urban and environmental planning such as land use, flood and water resources management of watersheds [3]. A popular approach is modeling the relationship between rainfall and runoff process, by using a set of equations that explain runoff as a function of rainfall and may include other parameters used to describe watershed characteristics [57]. Although there is a cause and effect relationship between both processes, rainfall-runoff modeling with high accuracy is affected by its non-linear behavior associated with the complex characteristics of the water cycle [59]. In addition to rainfall, runoff depends on many other factors, such as initial soil moisture, land use, basin geomorphology, evaporation, infiltration, distribution and rainfall duration [58].

There are several approaches to model a the rainfall-runoff relationship. Methods like simple and multiple regression models, auto regressive moving average (ARMA), the lumped, process-oriented, deterministic bucket-style models, artificial neural networks and distributed, physics models, are part of them [90, 16]. Each of these approaches has its own advantages and drawbacks, but the reduction of the predictive uncertainty of the model is considered as one the paramount elements to be considered [57, 91]. This uncertainty can be incorporated taking into account measurement of the input variables, such as rainfall and temperature, as well as the selection of the parameter values used in the model, since they depend on the climatic conditions. Even the choice of the structure of the model (establishing governing equations) by itself implies uncertainty [92]. In fact, in a recent work, a large number of researchers looking for greater harmonization on research efforts, identified that the mechanism to disentangle and reduce the structural uncertainty of the model/parameter/input in hydrological prediction as one of main unresolved problems in hydrology [93].

Fuzzy logic and Neuro-Fuzzy models has emerged as useful approach to model phenomena with some degree of uncertainty, including hydrological processes. The main idea is to consider variables in a linguistically uncertain manner rather than numerical precise quantities [94, 76]. In this sense, the Adaptive Network-based Fuzzy Inference System (ANFIS) is one of the most common Neur-Fuzzy models applied to model the rainfall-runoff relationship [16, 95, 15, 96]. However, due to its limitations associated with parameter selection and overfitting, a series of hybrid models have been proposed. Zhou et al. [97] proposed an evolutionary recurrent ANFIS for modelling multi-step-ahead flood forecasts embedded with Genetic Algorithm (GA) and Least Square Estimator (LSE). Bartoletti et al. [43] proposed a combination between Principal Component Analysis (PCA) and ANFIS model with the emphasis on how to set-up an efficient data structure that produces a good output flow estimation. PCA approach provides some degree of freedom by choosing the number of components to keep, allowing a trade-off between accuracy and network complexity. Meanwhile, Nourani et al. [44] proposed to apply a wavelet transformation in the input time series and then introduce different scales in the ANFIS model to obtain the runoff prediction in a structure called WANFIS. The Wavelet transform provides useful decomposition of the main time series (rainfall and runoff in this case) to improve the forecasting. A recent modified Adaptive Neuro-Fuzzy Inference System for runoff estimation was proposed by Nath et al. [45]. This work was focused on reducing the computational complexity of ANFIS by incorporating the Particle Swarm Optimization (PSO) algorithm in a framework called PSO-ANFIS, which was used for estimating the parameters pertaining to ANFIS. Other optimized approaches such as ANFIS-ant colony optimization for continuous domain (ANFIS-ACOR) [98], ANFIS-Grey Wolf Optimizer (GWO) (ANFIS-GWO) [42], or ANFIS-genetic algorithms (ANFIS-GA) [15] have been proposed, with their own advantages and applicability.

The extension of ANFIS model called Self-Organization Neuro-Fuzzy Inference System (SONFIS) proposed by Allende-Cid et al. [21, 22], although has the advantage of automatically organizes its structure to identify the set of Takagi-Sugeno-Kang type rules necessary to model the data set, it has not been used for modeling hydrological phenomena, which serves as motivation for its use in runoff prediction. Since the relationship between rainfall and runoff has a degree of uncertainty, it may decrease if the number of membership functions and the respective rules are not set in advance.

On the other hand, the selection of lags of both time series (rainfall and runoff) is a key pre-processing step in day-ahead forecasting runoff events. The literature suggests, among the most common methods for lag selection methods, the partial autocorrelation function (PACF) [45, 99, 100, 101], cross-correlation analysis (CCA), or MICCA, a combination of these last two [102]. Nevertheless, these functions locate the lags based on their linear dependence with the value to be predicted so that for phenomena with non-linear behaviors, algorithms with a more flexible structure can identify the most optimal lags [44]. Other approaches used in hydrology are: entropy analysis [103], mutual information (MI) [104] and algorithms that might involve dimensionality reduction such as the gamma test [105], PCA or clustering methods [106]. Furthermore, in recent years, methods based on machine learning have been presented as alternatives for the selection of suitable inputs, such as the input variable selection proposed by Taormina and Chau [107] that employs an algorithm which combines Binary-coded discrete Fully Informed Particle Swarm optimization (BFIPS) and Extreme Learning Machines (ELM). More recently, Veloz et al. [89] proposed SIFAR (Self-Identification of Lags of an Autoregressive Takagi-Sugeno-Kang-Based Model) as a method to find the most influential lags of a fuzzy model for time series applications. SIFAR algorithm has not been used in hydrology and provides a local representation, which allows the description of a nonlinear system using a set of simple mathematical functions, aggregated to describe the overall complexity that underlies its dynamic process.

Moreover, considering a single source of rainfall can bias runoff modeling, generating errors that can affect environmental planning, a suitable combination of various estimated runoff values obtained from different single models could provide a more complete and accurate representation of the basin response than each individual model [30]. Considering this idea of integrating the output of different models, several schemes have been applied for hydrological forecasting, using various methods such as simple and weighted average, median, selection of the best initial results, multiple linear regression or ANN, among others [30, 33, 34, 35, 36, 37]. Fuzzy approaches have been used as combination models showing good results to fuse rainfall-runoff forecasts [54, 55]. However, Neuro-Fuzzy models have not been applied as stacking alternatives, only as single predictive runoff methods, even considering its advantages in modeling phenomena associated with degrees of uncertainties [10].

In Chile, conceptual methods have been mainly used to model the relationship between rainfall and runoff, despite their limitations associated with the assumption of environmental standard conditions [60]. Besides, there are no studies aimed at data combination, even though many investigations suggest the use of more than one meteorological station as strategy to reduce the uncertainty in runoff forecasting. Taking the advantages of working with Neuro-Fuzzy models and considering various meteorological stations (and physically distributed), in this work a novel stacking Neuro-Fuzzy framework to forecast the runoff from distributed meteorological stations is proposed. This method is applied in two Chilean watersheds and it represents a main contribution to professionals who work in the management of water resources in the country.

4.1 Study region

Data was collected from two Chilean watersheds: Diguillín river watershed in San Lorenzo (Atacalco), sub-watershed of the Diguillín river, and Colorado river watershed in Maule Region. Diguillín river is located between latitudes 36°48'S, 37°03'S and longitudes 71°19'W, 72°22'W with a drainage area of 208,49[km^2], inserted in Ñuble region and has a total route of 102 [km] to its confluence with the Itata River [108, 109] (Figure 4.1). Colorado River watershed is located inside the Lontué river sub-watershed and is controlled by the Colorado river pluviometric station in Junta con Palos located at 35°16'28" latitude and 71°00'10" longitude, with a drainage area of 878 [km^2] (Figure 4.2) [110, 111].

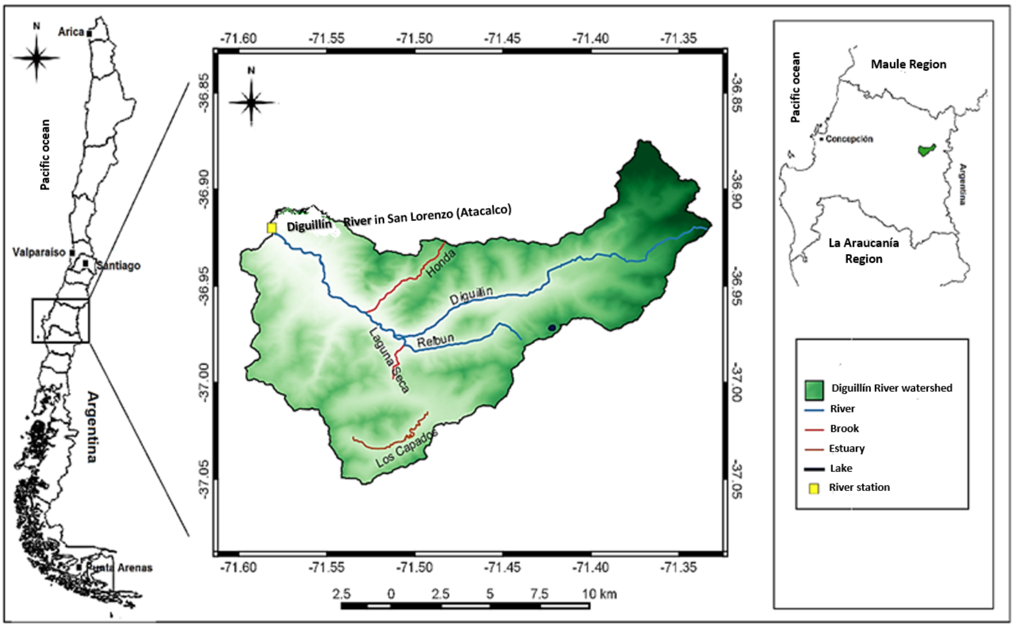


Fig. 4.1 Diguillín river watershed in San Lorenzo (Atacalco), sub-watershed of the Diguillín river, located in Ñuble Region, Chile.

Average daily rainfall recorded by "Fundo Atacalco" and "Las Trancas" stations and average daily runoff from the Diguillín river watershed from years 2000 to 2018 (Figure 4.3a),

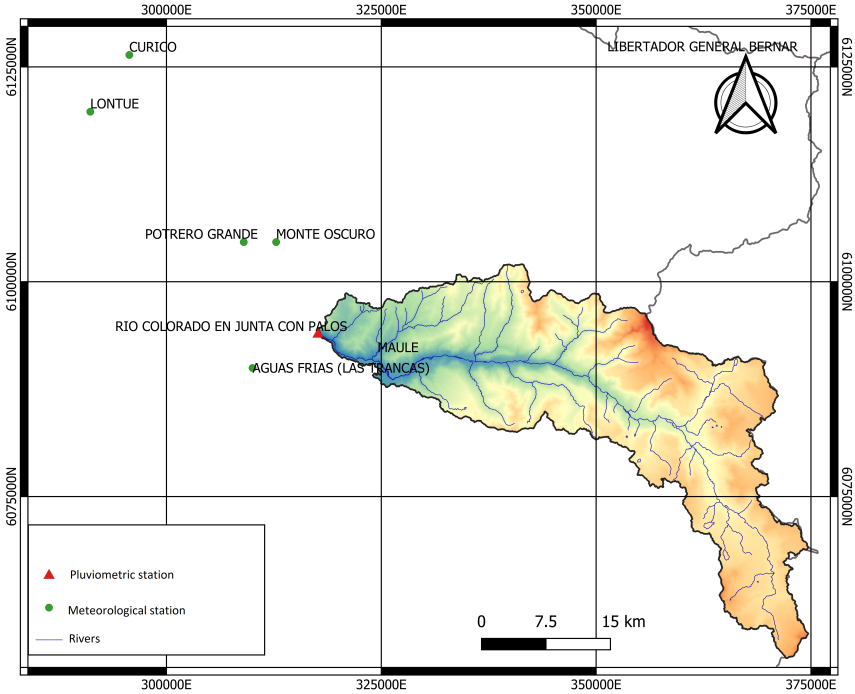
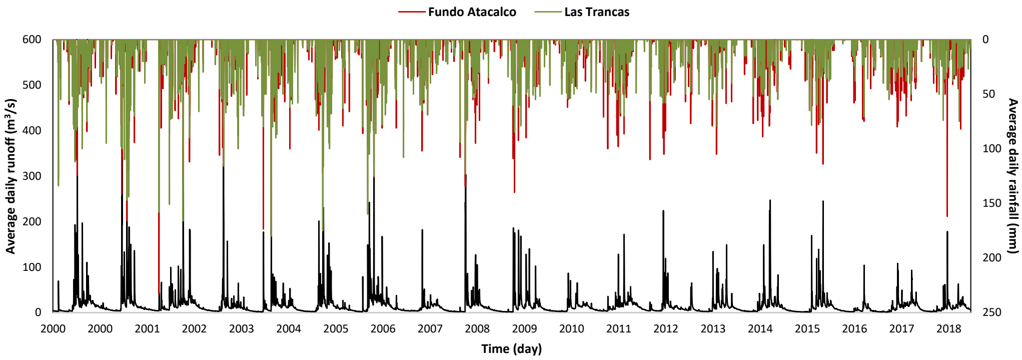


Fig. 4.2 Colorado river watershed located in the Province of Curicó, Maule Region, Chile.

and average daily rainfall recorded by “Agua fría", “Curicó", “Lontué", “Monte oscuro" and “Potrero grande" stations and average daily runoff from the Colorado river watershed from years 1999 to 2017 (Figure 4.3b), were collected from the Chilean directorate of water resources (DGA). Rainfall measurements were made using a Hellman rain gauge, meanwhile runoff measurements were made with a limnimeter, and the geographical location of each metorological station is showed in Table 4.1. In both watersheds, average daily rainfall was higher in the first ten years, having as maximum peak in the year 2002 and minimum peaks after 2010. In fact, it should be taken into account that, since 2010, an uninterrupted sequence of dry years, with annual rainfall deficits ranging from 25 to 45%, has prevailed in central Chile giving an event series called megadrought (MD). This is the longest continuous dry spell in the historical record (1915 onwards) and it coincided with a very warm decade in the interior valleys of central Chile and the subtropical Andes [112]. In contrast, the average daily runoff were as diverse as average daily rainfall. During the first 5 years, the runoff values were higher but with a progressive decrease and with extreme minimum values in 2013 and 2017.

(a) Diguillín river watershed.



(b) Colorado river watershed.

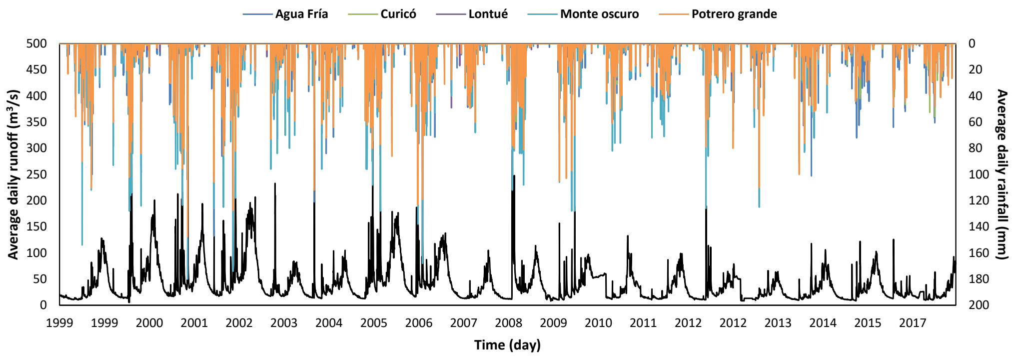


Fig. 4.3 Diguillín (a) and Colorado (b) river watershed data sets. Average daily rainfall shows great variability across all data with a notable decrease due the megadrought (MD) that has taken place in central Chile since 2010.

4.2 Single runoff forecasting using SINFIM method

To carry out Single runoff forecasting, it is necessary to develop the SINFIM method. For this phase, the "Fundo Atacalco" station in the Diguillín river watershed was considered, whose configuration of hyperparameters was later applied to the other metorological stations.

4.2.1 Parameter levels for Self-Identification of rainfall and runoff relevant lags

To develop the first step of the model, it is necessary to set some parameters required in the self-identification of relevant lags. Based on previous studies and knowledge expert we choose the parameters showed in Table 4.2, at the same time that ten repetitions were performed to evaluate the consistency of the lags found.

Table 4.1 Geographical location of the meteorological stations considered in the study.

Watershed	Meteorological station	Latitude	Longitude
Diguillín river	Fundo Atacalco	36°55’03”	71°34’53”
	Las Trancas	36°54’39”	71°28’46”
Colorado river	Agua Fría	35°18’26”	71°05’07”
	Curicó	34°58’30”	71°14’59”
	Lontué	35°02’38”	71°17’22”
	Monte oscuro	35°07’30”	70°58’43”
	Potrero grande	35°10’60”	71°05’52”

Table 4.2 Parameter levels for Self-Identification of rainfall and runoff relevant lags.

	Parameter	Level
c	Number of clusters for which the target and regressor space are partitioned.	3
q	Maximum number of lags that will be considered as input in the Neuro-Fuzzy model.	4
α_{out}	Threshold for which the α-cuts are generated for each partition of the target.	0,3
α_{in}	Threshold for which the α-cuts are generated for each partition of the regression.	0,3
α_f	Threshold to decide when to merge similar clusters.	0,9

Table 4.3 Parameter levels for the self-identification of fuzzy rules.

Parameter	**Level**			
Epoch training	2	3	5	
ε	0,1	0,3	0,5	
N_{split}	15	20	25	30
δ	0,5	0,7	0,9	
N_{grow}	10	15	20	25
λ	2			

The automatic selection of lags, in addition to provide the input required for runoff prediction, is expected to produce a certain degree of multi-collinearity due to both the relationship between precipitation and the generated runoff, as well as the dependence on the daily average runoff levels of the previous days. However, this tends to significantly affect predictions when the predictive method is a linear regression model, but with low impact on non-linear prediction models. An alternative for dealing with multi-collinearity is to use models such as neural networks, since they have a better fit and a lower mean square error [113, 114]. Thus, using a model that combines fuzzy logic with neural networks, in addition to the non-linear method used for selection of lags, it is generally safe to ignore the possible effect of multi-collinearity.

4.2.2 Parameter levels for the self-identification of fuzzy rules

Once the relevant lags of each time series model parameters were selected, two experiments were carried out dividing the time series into two non-overlapping segments (training and validation set). In the first experiment, the initial fifteen years of data was considered as training set and the last 3 years as validation set. In the second, the former seventeen years were used for the training set and the last year was used for the validation set. Such setup was done aiming to evaluate the ability of the SINFIM method for predicting runoff in any season of the year, having January as the beginning of summer and December as the end of spring. Hence, one year has been selected as the validation period (experiment 2). On the other hand, three years were selected as the validation set to evaluate the performance of the model in predicting the runoff values considering more than one period (experiment 1). For the purpose of choose the best parameters combination to develop the self-identification of fuzzy rules for each experiment, a factorial design was conducted carried out with the parameter levels as showed in Table 4.3.

Considering $N_{split} > N_{grow}$ and $N_{vanish} = N_{grow}$, 270 parameter combinations, could be obtained, which were compared using Mean Square Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{4.1}$$

where Y_i is the vector of observed runoff values and $\hat{Y}_i$ id the vector of predicted runoff values.

Due to mega-drought in the last periods, rainfall events represent in many cases extreme values that affect runoff levels. Therefore, the selection of parameters for the Neuro-Fuzzy model was made to make the algorithm sensitive to these extreme values. This differentiates it from the original model proposed by Allende-Cid et al. [21], where the adjustment of the parameters had as one of its objectives the stabilization of the model to increase its robustness.

4.2.3 Comparison with other approaches

In the interest of comparing the proposed SINFIM framework, other approaches were applied using the same relevant lags found:

1. *ANFIS.* This model was developed considering the number of membership functions (also Gaussian type) and number of epochs found in SINFIM method but without the self-identification of fuzzy rules.

2. *Artificial Neural Network (ANN).* A feed-forward neural network was developed considering the number of membership functions found with SINFIM method as hidden neurons size.

3. *Long Short-Term Memory (LSTM).* To develop this deep learning model, a factorial design was previously carried out to evaluate the most suitable set of hyperparameters: hidden layers size=10, 30, 50; mini-batch size=4, 12, 16, 32, 64; epochs=50, 100, 200, 300; optimizer=RMSprop, SGD, ADAM, and a activation function type “linear". The combination that showed the best result was: 50 hidden layers, 100 epochs of training, 32 mini-batch size and an optimizer ADAM.

In addition, the stochastic model ARIMA was developed to forecast runoff using rainfall as exogenous variable, but the results are shown in Appendix A, because the stationarity assumption was not met.

4.3 Stacking of single runoff predictions

In the proposal, a model based on SONFIS was considered as stacking phase, developing, for this, a preliminary experiment (similar to that carried out in the development phase of the

SINFIM method) to select the parameters of the mathematical operators Grow-Net, Split-Net and Vanish-Net. In order to compare with it, other stacking approaches were considered such as ANFIS model (with the same number of membership functions found in SONFIS), ANN (with the number of hidden neurons size based on the rules based on SONFIS), multiple linear regression (MLR) and the average of single runoff predictions.

4.3.1 Performance measures

To compare the proposed framework with the other models, the following performance measures were employed:

- Coefficient of Determination (R^2):

$$R^2 = \left[\frac{\sum_{i=1}^{n} (Y_i - \bar{Y}_i)(\hat{Y}_i - \tilde{Y}_i)}{\sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y}_i)^2} \sqrt{\sum_{i=1}^{n} (\hat{Y}_i - \tilde{Y}_i)^2}} \right]^2, \tag{4.2}$$

  where $\tilde{Y}_i$ corresponds to average of predicted runoff. R^2 values are in the interval $[0,1]$. In general, a value close to 1 indicates a good fit of the data and a high percentage of variability explained by the model.

- Nash-Sutcliffe Efficiency (NSE):

$$NSE = 1 - \frac{\sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y}_i)^2}, \tag{4.3}$$

  where $\bar{Y}_i$ is the average of observed runoff. NSE is a normalized statistic that determines the relative magnitude of the residual variance compared to the measured data variance, hence, $NSE = 1$ corresponds to a perfect match of the model to the observed data. According to Molnar [115], NSE values can be interpreted as follows: $< 0,2$ insufficient model, $0,2 - 0,4$ satisfactory model, $0,4 - 0,6$ good model, $0,6 - 0,8$ very good model and $> 0,8$ as an excellent model.

- Percent Bias (PBIAS):

$$PBIAS = 100 \cdot \frac{\sum_{i=1}^{n} (\hat{Y}_i - Y_i)}{\sum_{i=1}^{n} \hat{Y}_i}. \tag{4.4}$$

  PBIAS measures the average tendency of the simulated values to be larger or smaller than their observed ones. The optimal value of PBIAS is 0, with low-magnitude values indicating accurate model simulation. Positive values indicate overestimation bias, whereas negative values indicate model underestimation bias. Based on Moriasi et al. [116], the model can be analyzed as follows: $PBIAS < |10|$ very good, $|10| \leq PBIAS < |15|$ good, $|15| \leq PBIAS < |25|$ satisfactory, and $PBIAS \geq |25|$ unsatisfactory.

- Kling Gupta efficiency (KGE):

$$KGE = 1 - \sqrt{(r-1)^2 + \left(\frac{\sigma_{pred}}{\sigma_{obs}} - 1\right)^2 + \left(\frac{\tilde{Y}_i}{\bar{Y}_i} - 1\right)^2}, \tag{4.5}$$

where r is the linear correlation between observed and predicted values, while σ_{obs} and σ_{pred} represent the standard deviation of observed and predicted runoff values, respectively. KGE ranges from $-\infty$ to 1, where a value closer to 1 indicates a good accuracy for the model [117].

Once the model parameters were selected, each experiment was executed 10 times and results were presented in tables as mean $\pm$ standard deviation. To show the relationship between observed and predicted values (using the trial with the best result) in both training and validation set, were constructed scatter plots with 95% confidence intervals around the estimated regression line using bootstrap with 1000 resamples.

4.4 Experimental results

4.4.1 Experiment 1: Development of the SINFIM method for runoff forecasting

The first step of the framework gave the relevant lags to predict q_{t+1} through a self-identification model, obtaining $\{x_t$ and $q_t, q_{t-1}, q_{t-2}, q_{t-3}\}$ as rainfall x and runoff q time series relevant lags, respectively.

Once the input set was selected, a previous step was executed to include runoff missing values (254 of 6787 values). This forecasting was made considering the previous values as a training set to predict the missing values. After all of them had been estimated, the experiments were carried out as described in Chapter 3. On the other hand, missing rainfall values were imputed with the average value of observed rainfall in other stations.

With the input set, the parameter levels for the self-identification of fuzzy rules were evaluated, and the combinations with best performance are shown in Table 4.4 for the first experiment and Table 4.5 for the second experiment.

In the first experiment, using the last 3 years as a validation set, the combination number 1 showed the best MSE result, while the minimum MSE value was obtained in combination

Table 4.4 parameter levels for the self-identification of fuzzy rules using last 3 years as data validation set. The combination with the best results of MSE stands out.

N°	**Epochs**	ε	N_{split}	δ	N_{grow}	**MSE**$_{test}$	**N° fuzzy rules**
1	2	0,3	30	0,7	15	***53,168 ± 0,128***	14
2	3	0,1	25	0,7	15	54,657 ± 0,001	10
3	3	0,1	25	0,7	20	54,654 ± 0,006	10
4	3	0,3	20	0,7	15	54,685 ± 0,012	10
5	3	0,3	25	0,7	15	54,663 ± 0,015	10
6	3	0,3	25	0,7	20	54,658 ± 0,038	10
7	3	0,5	25	0,7	15	54,694 ± 0,050	10
8	3	0,5	30	0,7	15	54,654 ± 0,019	10
9	5	0,1	15	0,7	10	59,282 ± 0,203	15
10	5	0,1	20	0,7	10	56,887 ± 5,341	14
11	5	0,1	25	0,7	10	58,521 ± 0,005	15
12	5	0,1	25	0,7	15	57,416 ± 0,179	10
13	5	0,1	25	0,7	20	54,589 ± 4,829	10
14	5	0,1	30	0,7	15	58,360 ± 0,123	10
15	5	0,1	30	0,7	20	57,808 ± 4,913	10
16	5	0,5	15	0,7	10	52,994 ± 6,062	14
17	5	0,5	20	0,7	10	56,192 ± 5,380	15
18	5	0,5	20	0,7	15	58,507 ± 4,661	10
19	5	0,5	25	0,7	15	58,333 ± 0,133	10
20	5	0,5	30	0,7	10	63,981 ± 0,021	15

Table 4.5 Parameter levels for the self-identification of fuzzy rules using last year as data validation set. The combination with the best results of MSE stands out.

N°	**Epochs**	ε	N_{split}	δ	N_{grow}	**MSE**$_{test}$	**N° fuzzy rules**
1	2	0,3	30	0,7	15	37,995 ± 0,014	10
2	3	0,1	25	0,7	15	38,120 ± 0,001	10
3	3	0,1	25	0,7	20	38,113 ± 0,001	10
4	3	0,3	20	0,7	15	38,114 ± 0,014	10
5	3	0,3	25	0,7	15	38,115 ± 0,012	10
6	3	0,3	25	0,7	20	38,112 ± 0,015	10
7	3	0,5	25	0,7	15	38,108 ± 0,003	10
8	3	0,5	30	0,7	15	38,113 ± 0,023	10
9	5	0,1	15	0,7	10	41,101 ± 0,060	14
10	5	0,1	20	0,7	10	41,024 ± 0,532	14
11	5	0,1	25	0,7	10	41,699 ± 3,697	14
12	5	0,1	25	0,7	15	43,419 ± 6,149	10
13	5	0,1	25	0,7	20	43,446 ± 0,031	10
14	5	0,1	30	0,7	15	43,406 ± 5,956	11
15	5	0,1	30	0,7	20	43,372 ± 0,045	10
16	5	0,5	15	0,7	10	41,432 ± 0,010	14
17	5	0,5	20	0,7	10	40,943 ± 0,784	14
18	5	0,5	20	0,7	15	43,431 ± 0,039	10
19	5	0,5	25	0,7	15	39,629 ± 0,072	10
20	5	0,5	30	0,7	10	***33,951 ± 0,013***	14

20, when only the last year was used as validation set in the second experiment. As a result, both combinations represent, within the options considered, the best set of parameters to model the Rainfall-Runoff relationship with the dataset used.

4.4.2 Experiment 2: Single runoff forecasting- comparison of the SINFIM method with other models

In this section, the performance of SINFIM to model rainfall-runoff relationship of Diguillín river watershed using “Fundo Atacalco" station is shown and compared with the other approaches. The results using the last three years as a validation set (experiment one) are showed in Table 4.6. SINFIM showed the best performance in almost all measures, highlighting its minimum value of MSE and the highest R^2, NSE and KGE results are highlighted; followed by ANN and ANFIS models. LSTM showed the worst performance, with the highest MSE and lowest R^2 and NSE values.

Despite having a good behavior compared to the other algorithms, the SINFIM framework presents a 71% accuracy, leaving still 29% of total variance that is not explained by it. In fact, Figure 4.4 shows that, although in the majority of the validation set there is a good estimation when using the last 3 years as a validation set, there are certain points where the model fails to reach the flow peaks, while in others it has to overestimate the values. This lack of adjustment can be observed in Figure 4.5. In this figure, it can be seen, that although most of the points are within the confidence bands, there are still many scattered values that are not predicted correctly. However, this overestimation is not large, since the average PBIAS obtained was less than 10 and its NSE described it as good model.

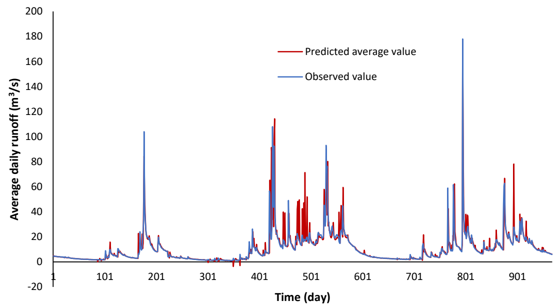
Using last year as validation set showed the same pattern observed in experiment one: SINFIM had the best performance in all measures, followed by ANN and ANFIS. Nevertheless, the measure results were better, reducing the MSE value and increasing R^2 (around 84%), NSE and KGE values, indicating that SINFIM is an excellent model with a good accuracy to determine the relationship between Rainfall and Runoff. In Figure 4.6 a very good performance is observed thorough all data with the exception of two points where an overestimation followed by an underestimation of runoff occurs, respectively. But, it is only two out of 364 runoff values from the validation set. Therefore, the model has proven to be a good alternative for predicting of runoff at any time of the year. This good fit can be seen in the Figure 4.7, backing up the NSE result that asserts SINFIM as a remarkable alternative as a model. The figure shows that only less than 10 points fall out of the confidence interval, indicating a good adjustment and subsequently, a satisfactory prediction of the runoff values.

Table 4.6 First experiment results: using last three years as validation set. In training and validation sets, the best performance in all measures was achieved using SINFIM as method to predict one day-ahead runoff.

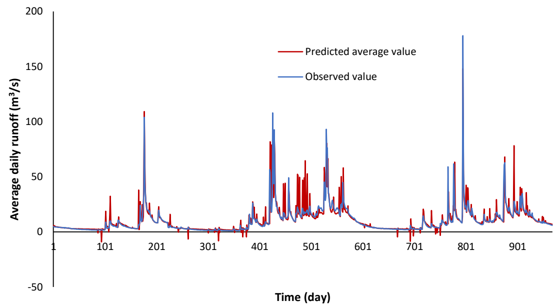
Training					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
SINFIM	**193,44±0,3**	**0,74±0,00**	**0,64±0,001**	**≈ 0,000**	**0,80±0,00**
ANFIS	429,33±190,53	0,42±0,26	-0,51±1,02	9,93±8,22	0,5±0,25
ANN	208,85±24,76	0,72±0,02	0,65±0,02	8,57±0,28	0,75±0,09
LSTM	587,59±14,74	0,19±0,02	-2,79±0,22	66,62±0,19	-0,02±0,01
Validation					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
SINFIM	**53,45±0,05**	**0,71±0,01**	**0,70±0,00**	-4,74±0,08	**0,83±0,00**
ANFIS	62,34±4,68	0,67±0,03	0,65±0,04	-4,92±1,10	0,80±0,01
ANN	63,12±19,88	0,70±0,03	0,68±0,02	7,19±1,42	0,71±0,22
LSTM	84,68±1,17	0,58±0,02	0,56±0,03	**-1,37±3,76**	0,74±0,00

Table 4.7 Second experiment results: using the last year as a validation set. In training and validation sets, the best performance in all measures was achieved using SINFIM as method to predict one day-ahead runoff.

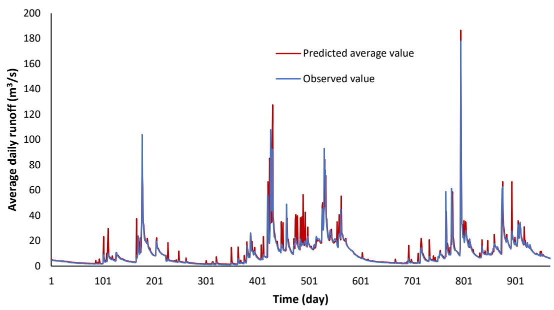
Training					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
SINFIM	**171,31±0,19**	**0,75±0,00**	**0,66±0,00**	**≈ 0,000**	**0,81±0,00**
ANFIS	209,37±43,32	0,69±0,06	0,55±0,13	1,41±1,59	0,76±0,05
ANN	180,83±13,40	0,73±0,02	0,66±0,04	-0,70±4,87	0,81±0,03
LSTM	536,34±0,79	0,21±0,02	-2,69±0,14	66,13±1,17	-0,01±0,00
Validation					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
SINFIM	**30,54±0,04**	**0,84±0,01**	**0,83±0,01**	**-1,26±0,05**	**0,91±0,00**
ANFIS	49,69±9,49	0,73±0,06	0,69±0,09	-1,88±0,87	0,84±0,05
ANN	41,11±7,21	0,83±0,01	0,82±0,02	-6,31±4,92	0,81±0,07
LSTM	62,74±8,07	0,69±0,02	0,67±0,01	2,61±0,92	0,82±0,02



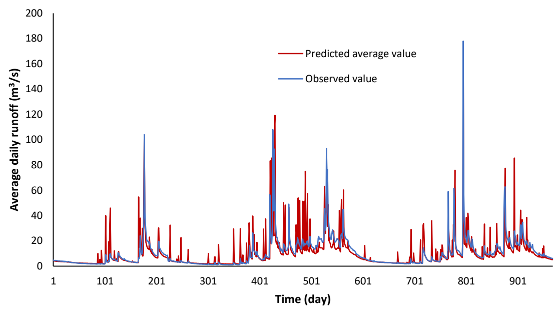
(a) SINFIM



(b) ANFIS

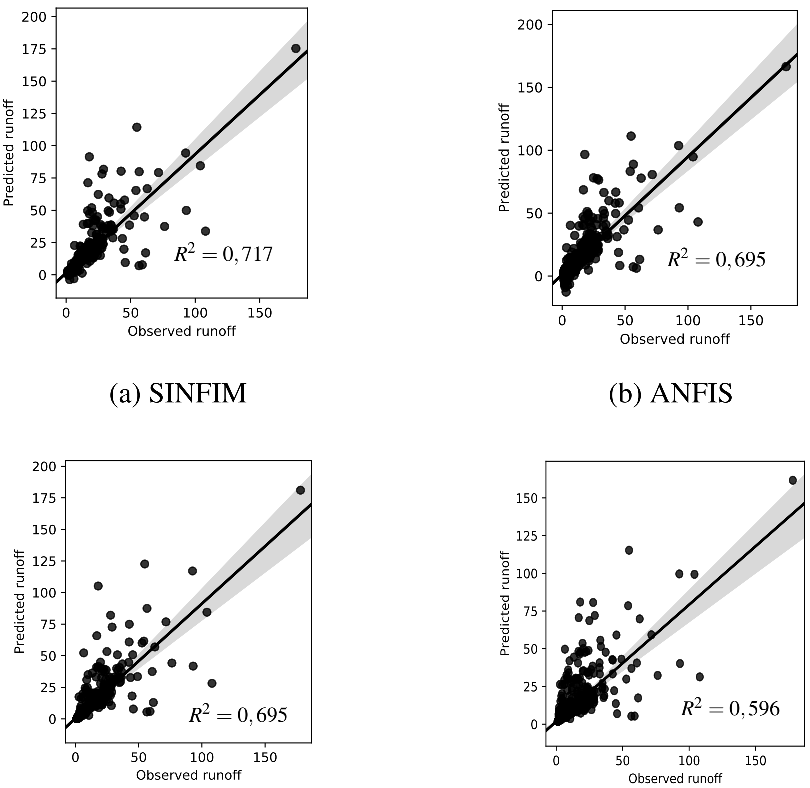


(c) ANN



(d) LSTM

Fig. 4.4 Predicted runoff vs observed runoff using the last 3 years as validation set. Although the best performance is observed with the SINFIM method, there are certain points where the model fails to reach the flow peaks, while in others it has to overestimate the values.



(a) SINFIM (b) ANFIS

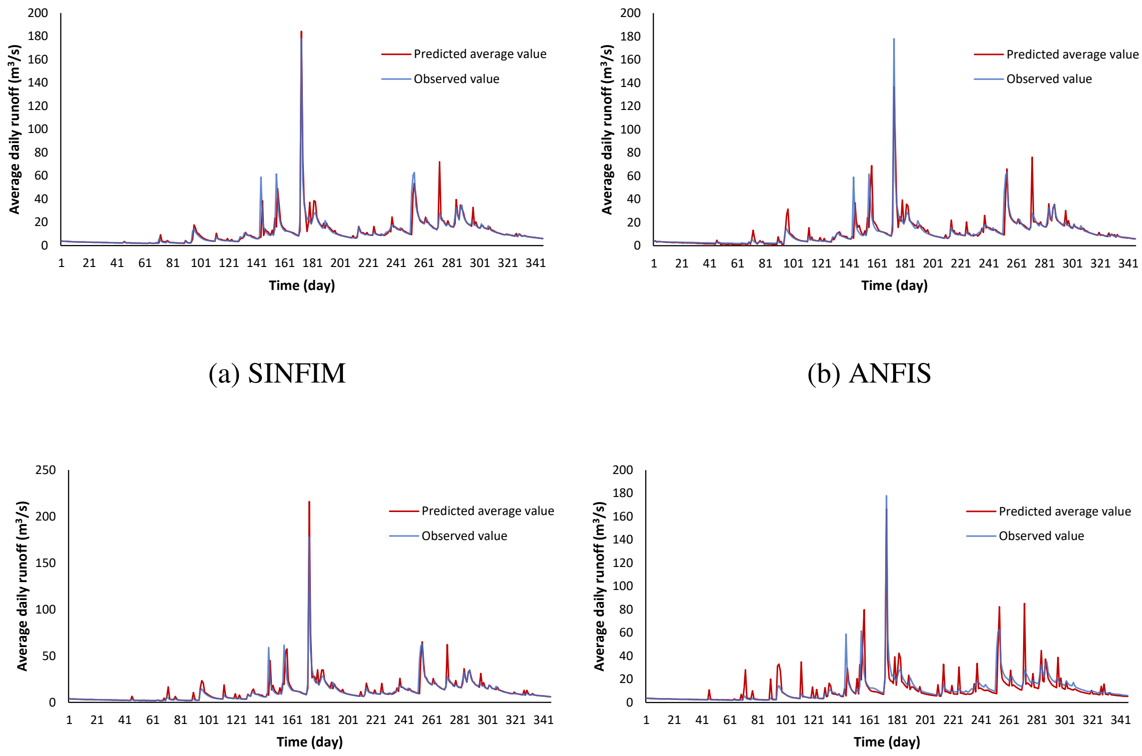
(c) ANN (d) LSTM

Fig. 4.5 Predicted runoff vs observed runoff using the last 3 years as a validation set. The trial with the best result in each method were selected as observed runoff. The best performance was observed with SINFIM method but, although most of the points are within the confidence bands, there are still many scattered values (more than 25) that are not predicted correctly.

4.4.3 Experiment 3: Application of SINFIM to all the meteorological stations considered

As well as it was obtained in the development of the SINFIM method using the "Fundo Atacalco" station in previous section, for the other meteorological stations in both watersheds considered, $\{x_t$ and $q_t, q_{t-1}, q_{t-2}, q_{t-3}\}$ as rainfall x and runoff q time series relevant lags, respectively, were obtained. In turn, as experiment two (using the first 17 years as a training set and the last year as a validation set) was the one with the best results, this configuration (both in training and validation periods and the parameters used in the predictive module) was used to obtain the runoff forecasting.

Results obtained in Diguillín river watershed are shown in Table 4.8. Single runoff forecasting using SINFIM model showed better performance with "Fundo Atacalco" station than "Las Trancas" station both training and validation set. In the latter, the performance was



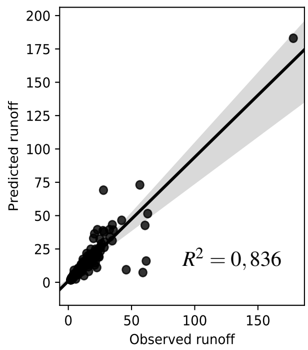
(a) SINFIM (b) ANFIS

(c) ANN (d) LSTM

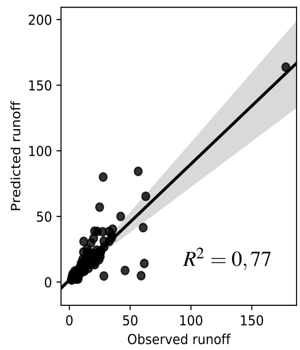
Fig. 4.6 Predicted runoff vs observed runoff using the last year as a validation set. The best performance is observed in all data using the SINFIM method.

wider, showing an MSE three times greater in the “Las Trancas" station, and a percentage of improvement in terms of R^2 of 40% in the case of the “Fundo Atacalco" station. Also, “Fundo Atacalco" showed a better efficiency indexes NSE and KGE, highlighting its ability to predict one day-ahead runoff. This can be seen in Figure 4.8 where both stations had good performance in modeling low and medium values of runoff but with rainfall data recorded in “Las Trancas" stations peak runoff values could not be predicted. Besides, a better adjustment was observed, also, in “Fundo Atacalco" station (Figure 4.9).

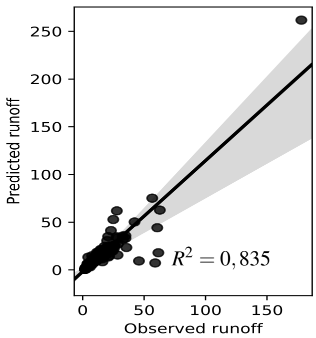
On the other hand, results obtained in Colorado river watershed are shown in Table 4.9. In contrast with observed in Diguillín river watershed, all meteorological stations considered showed good performance in all measures both training and validation set. R^2 always was higher than 86%, and NSE and KGE showed that SINFIM model in all cases was a good model to predict runoff. Besides, PBIAS index indicates an efficient forecasting of SINFIM



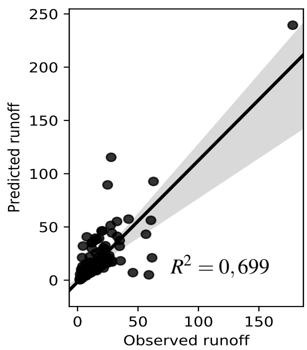
(a) SINFIM



(b) ANFIS



(c) ANN



(d) LSTM

Fig. 4.7 Predicted runoff vs observed runoff using the last year as a validation set. The trial with the best result were selected as observed runoff. The best performance was observed with SINFIM method followed by ANN. Less than 10 points fall out of the confidence interval, indicating a good adjustment and subsequently, a satisfactory prediction of the runoff values.

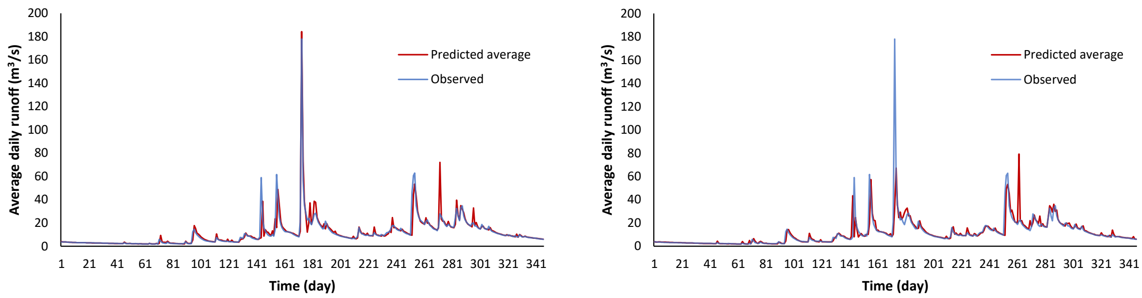
in all stations. However, “Agua Fría", “Monte Oscuro" and “Potrero Grande" have the best results (Figure 4.10), showing a good adjustemnt forecasting both low and peak runoff values, being more accurate in “Monte Oscuro" station (Figure 4.11).

4.4.4 Experiment 4: Stacking of Single runoff predictions

Once the single runoff predictions were obtained, they were combined to estimate the final one day-ahead forecasting. In order to identify the best combination of parameter levels to develop the method based on SONFIS model, a factorial design was carried out and, in case of Diguillín river watershed, the best results are shown in Table 4.10. Combination three has the lowest MSE in validation set, therefore this levels were used in the stacking step.

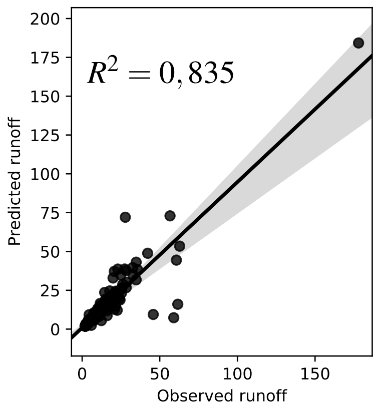
Table 4.8 Runoff one day-ahead forecasting in Diguillín river watershed using SINFIM method. The best results were obtained in “Fundo Atacalco" station.

Training					
Meteorological station	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Fundo Atacalco	**171,3±0,2**	**0,747±0,00**	**0,66±0,00**	≈ 0,000	**0,81±0,00**
Las Trancas	194,4±2,6	0,713±0,004	0,60±0,00	≈ 0,000	0,78±0,00
Validation					
Meteorological station	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Fundo Atacalco	**30,5±0,0**	**0,835±0,00**	**0,83±0,00**	-1,3±0,1	**0,91±0,00**
Las Trancas	97,0±2,1	0,453±0,009	0,07±0,01	≈ 0,000	0,60±0,00

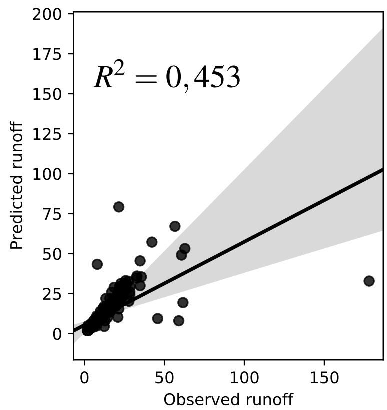


(a) “Fundo Atacalco" station (b) “Las Trancas" station

Fig. 4.8 Runoff one day-ahead forecasting in Diguillín river watershed. In both stations, low and medium values of runoff could be predicted, but peak values only was estimated in “Fundo Atacalco" station.



(a) “Fundo Atacalco" station

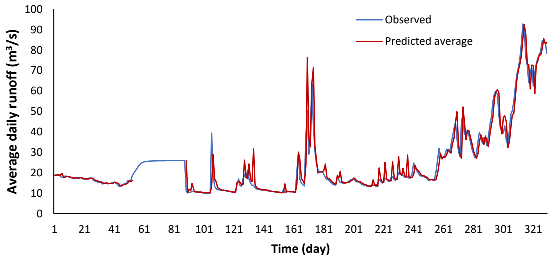


(b) “Las Trancas" station

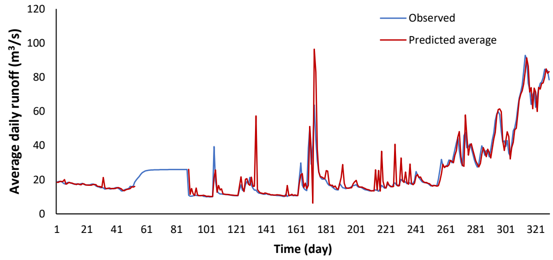
Fig. 4.9 Predicted runoff vs observed runoff in single models of Diguillín river watershed. A better adjustment was observed in “Fundo Atacalco" station.

Table 4.9 Runoff one day-ahead forecasting in Colorado river watershed. In all the stations considered, a very good performance was obtained, however, the best results were from “Monte Oscuro" station.

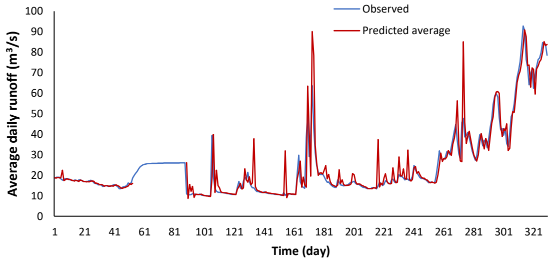
Training					
Meteorological station	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Agua Fría	272,2±18,4	0,826±0,00	0,78±0,02	0,4±0,0	0,85±0,01
Curicó	296,8±6,2	0,810±0,00	0,76±0,01	0,4±0,0	0,85±0,00
Lontué	299,4±6,4	0,808±0,00	0,75±0,01	0,4±0,0	0,85±0,00
Monte Oscuro	**257,0±2,0**	**0,836±0,00**	**0,79±0,00**	0,4±0,0	**0,86±0,00**
Potrero Grande	288,3±7,7	0,815±0,01	0,76±0,01	0,4±0,0	0,85±0,00
Validation					
Meteorological station	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Agua Fría	22,0±2,3	0,935±0,01	0,93±0,00	-1,5±0,3	0,96±0,00
Curicó	39,9±1,5	0,886±0,00	0,89±0,00	-2,9±0,6	0,93±0,01
Lontué	38,1±5,1	0,893±0,01	0,89±0,01	-2,6±0,4	0,93±0,01
Monte Oscuro	**21,0±2,1**	**0,937±0,01**	**0,94±0,01**	-1,4±0,3	**0,97±0,00**
Potrero Grande	21,2±1,6	0,936±0,00	0,93±0,01	-0,9±0,4	0,96±0,00



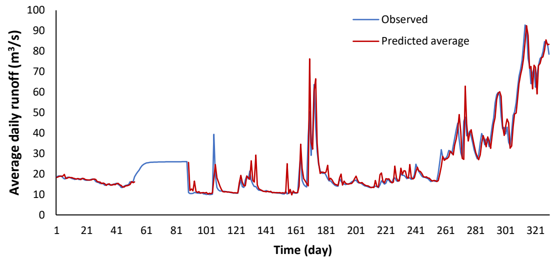
(a) “Agua fría" station



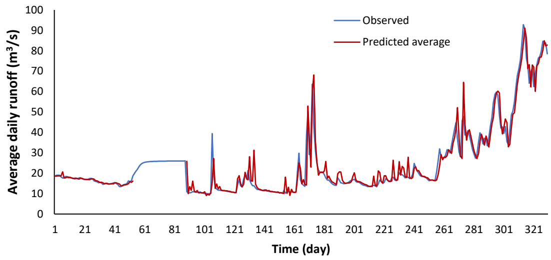
(b) “Curicó" station



(c) “Lontué" station

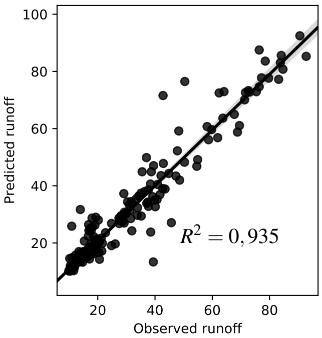


(d) “Monte oscuro" station

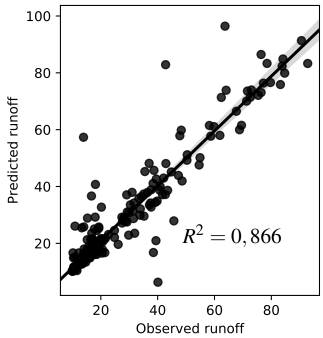


(e) “Potrero grande" station

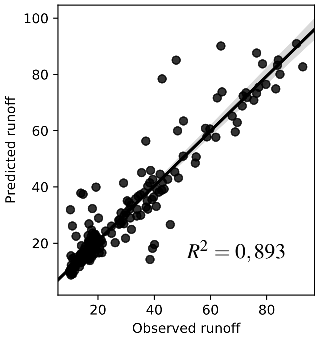
Fig. 4.10 Runoff one day-ahead forecasting in Colorado river watershed. “Agua Fría", “Monte Oscuro" and “Potrero Grande" stations have the best results, although all models showed good performance.



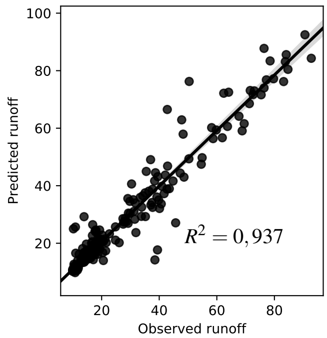
(a) “Agua Fría" station



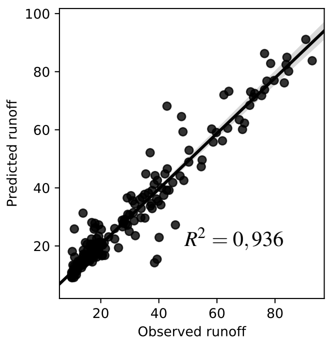
(b) “Curicó" station



(c) “Lontué" station



(d) “Monte Oscuro" station



(e) “Potrero Grande" station

Fig. 4.11 Predicted runoff vs observed runoff in single models of Colorado river watershed. In all the individual cases most of the points are within the confidence interval on the trend line, being “Monte Oscuro" station with the best adjustment.

Table 4.10 Parameter levels for SONFIS model to combine single one day-ahead runoff predictions in Diguillín River. The combination with the best results of MSE stands out.

N°	**Epochs**	ε	N_{split}	δ	N_{grow}	**MSE**$_{test}$	**N° fuzzy rules**
1	2	0,3	30	0,7	15	29,515 ± 0,016	45
2	2	0,3	25	0,9	20	30,593 ± 3,660	21
3	**2**	**0,3**	**20**	**0,9**	**15**	**29,283 ± 0,063**	**17**
4	2	0,1	30	0,7	15	29,484 ± 0,064	45
5	3	0,3	20	0,7	15	32,163 ± 0,013	33
6	3	0,5	30	0,7	15	30,114 ± 0,032	44
7	5	0,1	20	0,7	10	30,083 ± 2,415	34
8	5	0,1	25	0,7	10	29,651 ± 0,234	46
9	5	0,5	20	0,7	10	30,397 ± 3,942	32
10	5	0,5	30	0,7	10	30,386 ± 6,309	31

Table 4.11 Runoff one day-ahead forecasting in Diguillín river watershed using stacking approaches. ANFIS and SONFIS model showed better performance in training and validation set, respectively.

Training					
Method	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Average	161,5±1,5	0,76±0,00	0,658±0,004	≈ 0,0	0,792±0,002
MLR	159,1±1,2	0,77±0,00	0,692±0,003	≈ 0,0	0,823±0,001
ANFIS	**137,0±0,1**	**0,80±0,00**	**0,746±0,000**	≈ 0,0	**0,849±0,000**
ANN	153,9±13,4	0,77±0,02	0,716±0,029	0,6±4,9	0,834±0,021
SONFIS	146,3±0,0	0,78±0,00	0,724±0,000	≈ 0,000	0,838±0,000
Validation					
Method	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Average	40,8±0,7	0,768±0,005	0,661±0,012	-0,7±0,6	0,791±0,11
MLR	34,1±1,6	0,805±0,008	0,762±0,012	0,5±0,8	0,860±0,011
ANFIS	37,5±1,0	0,787±0,006	0,692±0,013	0,3±0,1	0,801±0,007
ANN	61,3±44,3	0,812±0,025	0,755±0,057	-2,0±7,2	0,687±0,211
SONFIS	**29,3±0,1**	**0,842±0,000**	**0,837±0,001**	-2,0±0,0	**0,914±0,000**

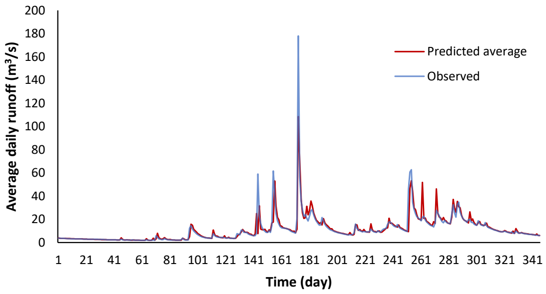
Table 4.12 Parameter levels for SONFIS model to combine single one day-ahead runoff predictions in Colorado River. The combination with the best results of MSE stands out.

N°	**Epochs**	ε	N_{split}	δ	N_{grow}	**MSE**$_{test}$	**N° fuzzy rules**
1	2	0,3	25	0,9	20	21,706 ± 0,337	15
2	2	0,3	20	0,9	15	21,870 ± 0,341	15
3	2	0,1	25	0,7	20	24,384 ± 0,040	28
4	**3**	**0,1**	**25**	**0,5**	**10**	**20,472 ± 0,015**	**60**
5	3	0,3	20	0,7	15	21,385 ± 0,021	15
6	5	0,1	15	0,7	10	21,660 ± 0,012	16
7	5	0,1	20	0,7	10	21,662 ± 0,021	16
8	5	0,5	15	0,7	10	21,658 ± 0,048	16
9	5	0,5	20	0,7	10	21,671 ± 0,003	16
10	5	0,5	20	0,7	15	21,318 ± 0,027	15

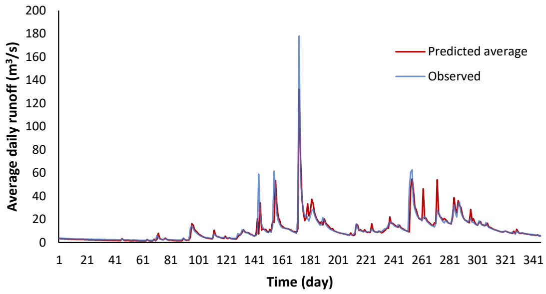
As it was observed in Table 4.11 and Figure 4.12, fusion approaches showed better performance than single forecasting in training set, with ANFIS model with the best measures. Nevertheless, SONFIS model had the best result in all measures in validation set, improving MSE, R^2, NSE and KGE values, being higher than those obtained in “Fundo Atacalco" station as best single forecasting. This adjustment is observed in Figure 4.13, with values closer to trend line.

In the same way as Diguillín river watershed, the best parameter level to combine single runoff forecasting with SONFIS model were evaluated in Colorado river watershed. In this case, the combination four showed the lowest MSE, and it was used in the stacking step (Table 4.12).

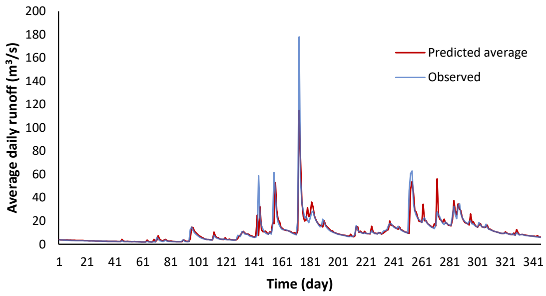
In the same way that Diguillín river watershed and as it was observed in Table 4.13 and Figure 4.14, fusion schemes showed better performance, but in this watershed, SONFIS approach had the best results both training and validation set. With the exception of KGE, SONFIS method improved the single runoff forecasting, increasing the R^2 and NSE values, and decreasing the MSE. Since the individual predictions showed a very high efficiency with the KGE index, the contributions provided by the fusion are minimal. This better adjustment can be seen in Figure 4.15. Although, in all the stacking cases, most of the points are within the confidence interval on the trend line, the lower dispersion was observed with SONFIS model.



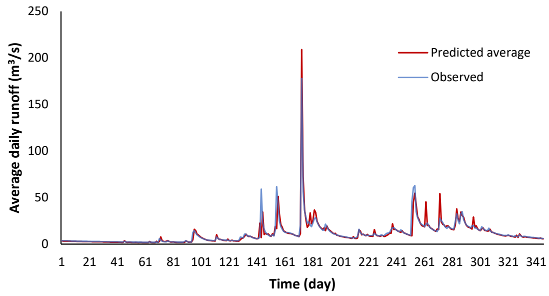
(a) Average



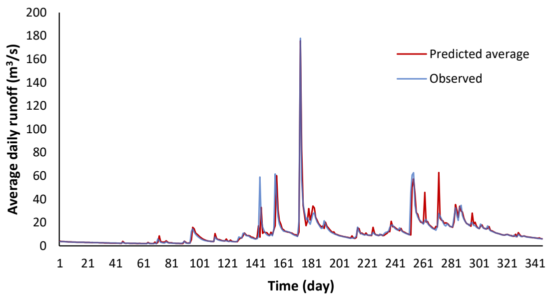
(b) Multiple linear regression



(c) ANFIS

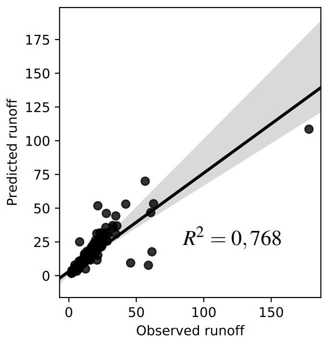


(d) Artifical Neural Network

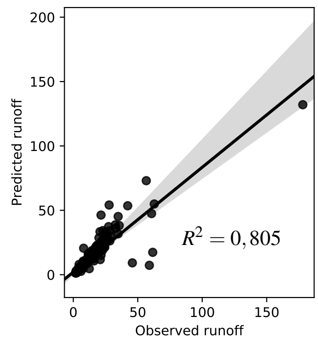


(e) SONFIS

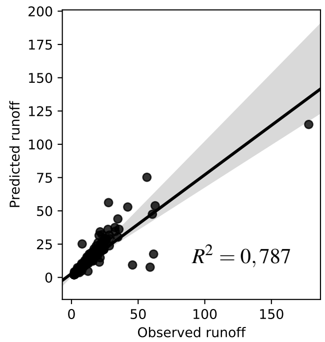
Fig. 4.12 Runoff one day-ahead forecasting in Diguillín river watershed using stacking approaches. The best performance was observed with SONFIS model with good forecasting of low and peak runoff values.



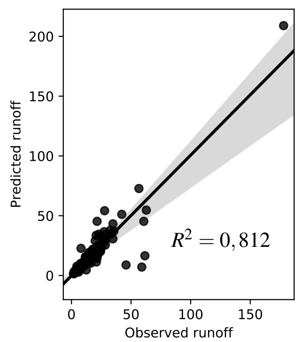
(a) Average



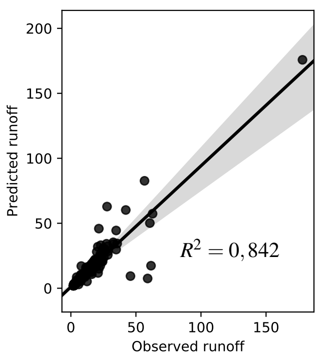
(b) Multiple linear regression



(c) ANFIS

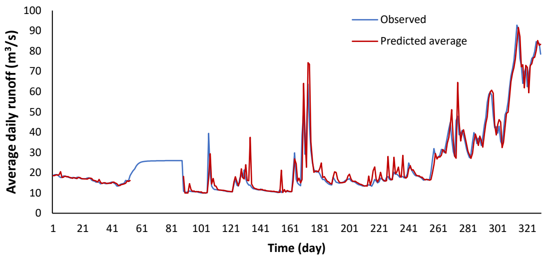


(d) Artificial Neural Network

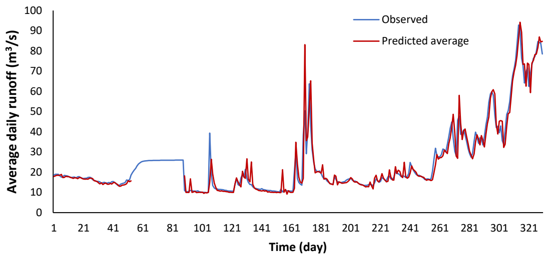


(e) SONFIS

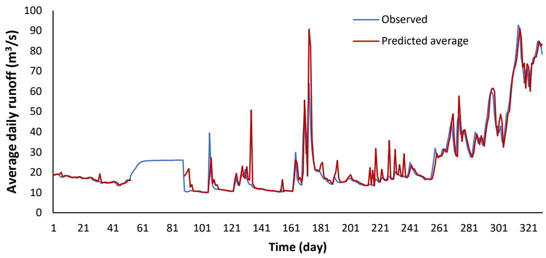
Fig. 4.13 Predicted runoff vs observed runoff of Diguillín river watershed using stacking approaches. Although, in all cases, most of the points are within the confidence interval on the trend line, the lower dispersion was observed with SONFIS model.



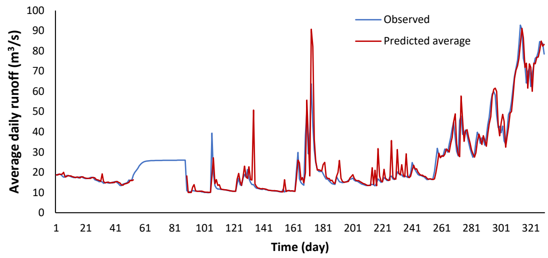
(a) Average



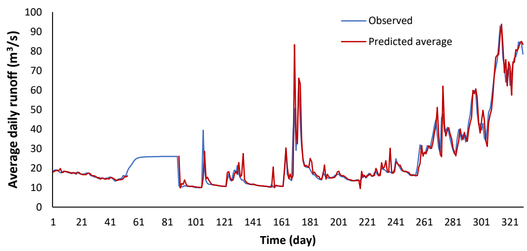
(b) Multiple linear regression



(c) ANFIS

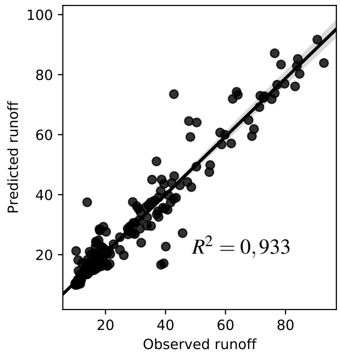


(d) Artifical Neural Network

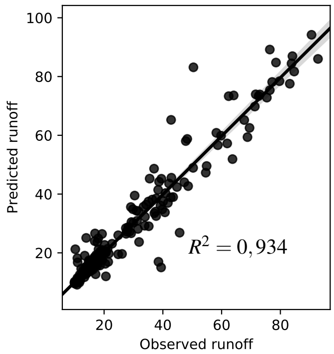


(e) SONFIS

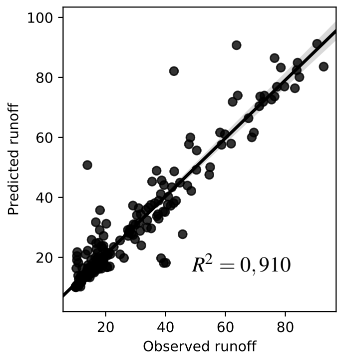
Fig. 4.14 Runoff one day-ahead forecasting in Colorado river watershed using stacking approaches. Although all models showed good performance, SONFIS had the best result predicting low and peak runoff values.



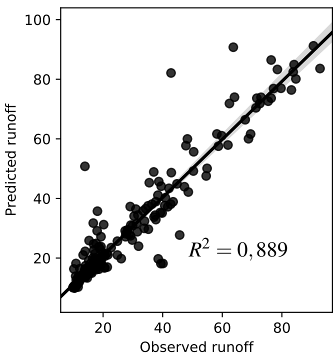
(a) Average



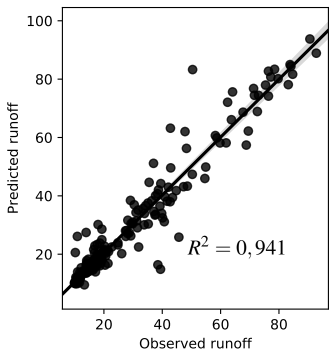
(b) Multiple linear regression



(c) ANFIS



(d) Artificial Neural Network



(e) SONFIS

Fig. 4.15 Predicted runoff vs observed runoff in Colorado river watershed using stacking approaches. Although, in all cases most of the points are within the confidence interval on the trend line, the best adjustment was observed with SONFIS model.

Table 4.13 Runoff one day-ahead forecasting in Colorado river watershed using stacking approaches. SONFIS model had the best performance in both training and validation set.

Training					
Method	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Average	267,4±5,2	0,829±0,003	0,778±0,005	0,4±0,0	0,850±0,002
MLR	247,4±12,1	0,841±0,008	0,811±0,011	$\approx$ 0,0	0,833±0,006
ANFIS	289,9±2,3	0,815±0,002	0,757±0,002	0,4±0,0	0,842±0,003
ANN	262,9±89,5	0,841±0,045	0,808±0,062	-3,8±5,0	0,867±0,066
SONFIS	**180,0±1,5**	**0,885±0,001**	**0,869±0,001**	$\approx$ 0,000	**0,916±0,001**
Validation					
Method	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
Average	22,6±1,1	0,933±0,003	0,932±0,003	-1,8±0,1	0,961±0,002
MLR	22,9±4,9	0,934±0,013	0,934±0,013	0,2±0,6	0,956±0,010
ANFIS	31,2±3,5	0,910±0,009	0,909±0,010	-2,9±0,3	0,942±0,006
ANN	37,6±12,4	0,889±0,037	0,880±0,044	-10,5±9,8	0,866±0,085
SONFIS	**20,5±0,2**	**0,941±0,000**	**0,941±0,000**	-1,2±0,5	0,959±0,002

Chapter 5

Discussion

In this work a new Stacking Neuro-Fuzzy model for time series forecasting, considering the case of one day-ahead runoff from distributed meteorological stations is proposed, considering an hybrid framework with SINFIM method as single runoff forecasting and SONFIS model as stacking approach.

First, in both developed experiments, the proposed SINFIM method for single runoff forecasting achieved a superior performance to the other evaluated predictive models such as ANFIS, ANN, LSTM and the stochastic model ARIMAX, obtaining the best results when using only one year as a validation set. This may be mainly since the variability in the observations of rainfall and runoff decreases as the period to be evaluated is reduced, considering also that the 3 years used as the validation set in experiment 1 are different from each other. With the exception of "Las Trancas" station in Diguillín River, all results of single meteorological stations achieved an accuracy greater than 80%, being higher in the stations of the Colorado river watershed with 93,7% of adjustment in "Monte Oscuro" station. These results were better than those reported by [118] where a comparative study of different wavelet based neural network models for rainfall–runoff modeling was presented. Although the authors highlighted the advantage of pre-processing the time series by extracting features with the wavelet transformation as other studies [53], in addition to the choice of the mother wavelet to improve performance, SINFIM model shows a better NSE values and a higher R^2 in all cases. Similarly, our results indicate better performance metrics than those reported by Behmanesh and Ayashm [119], who assessed the rainfall-runoff relationship in a basin in Mississippi, applying ANN and ANFIS as predictive models. The proposed method outperforms the results by Kumar et al. [120], who applied a traditional ANN to forecast runoff. This results might suggest that the ANN and ANFIS model by themselves do not generate a good prediction of runoff, given the complexity of its relationship with rainfall, in addition to having other influential factors such as temperature or soil saturation.

Another relevant result was the low performance obtained with the stochastic ARIMAX model in runoff forecasting. This has been reported in the literature, mainly due to the linear nature of the ARIMAX model, which fails to capture the behavior of a non-linear phenomenon with high uncertainty such as the rainfall-runoff modeling [3]. For this reason, proposals for hybrid models have arisen where stochastic models are combined with machine learning methods, mainly neural networks, thus improving runoff forecasting [121, 122].

Researches such as Nath et al. [45], Nourani et al. [44] or Anusree and Varghese [16], presented an optimized algorithm using different input combinations. One of the main advantages of SINFIM is that it includes a module for identifying relevant lags so that an a priori selection of the input elements to the predictive model is not required. Although there are other methods for the selection of the input elements of the rain time series and runoff used in Neuro-Fuzzy models, such as partial autocorrelation function (PACF) [45, 99], cross-correlation analysis (CCA) [102], entropy analysis [103] or mutual information (MI) [104]; we consider that SINFIM method is more advantageous, because its structure is developed with a focus on identify lags in a nonlinear environment such as in fuzzy systems [89].

Although Deep Learning (DL) approaches have shown great performances in modeling the relationship between rainfall and runoff in different watersheds and synthetic data [123, 124, 125, 126], according to our results, the Long Short-Term Memory (LSTM) network developed showed the lowest performance as a forecasting models. Despite the fact that among the advantages of the LSTM is its ability to learn long-term dependencies between the provided input and output of the network [127], the high variability of the observations in our case study might have influenced the performance of the model. This could be explained because the data includes daily observations from all seasons of the year, as well as a great period with little rain associated with the mega-drought phenomenon that has been going through Chile since 2010 [112]. In fact, data heterogeneity, in terms of quantities measured and scales, have already been mentioned as an difficult aspect to directly implement existing results obtained on hydrological deterministic experiments [128]. Furthermore, SINFIM method has shown comparable performance with works such as Jothiprakash et al. [95], Morales et al. [129], showed several models with good performances, through varying the number of membership functions and combinations in the time series. Although, in some of these models, the R^2 performance was higher than that obtained in our study, they have the drawback that the user has to set the number of membership functions, while SINFIM does it automatically. Similarly, our performance results in Diguillín river watershed are lesser than the second of the architectures proposed by Nourani et al. [3], who presented a fusion between the wavelet transformation with ANFIS model (with a R^2

of 90%). This could be mainly due to the characteristics of the watersheds under study. Nourani et al. [3] used data from watersheds located in Iran, whose daily maximum runoff values do not exceed $90m^3/s$ while rainfall is less than 40mm. In our work, the hydrological conditions are more variable, having precipitation values that reach 233mm and runoff rates up to $544m^3/s$. Therefore, the ability of SINFIM to model extreme runoff values given a rainfall event is highlighted, a fact that can be seen in Figures 4.4 and 4.6, where the model successfully predicts runoff peaks greater than $150m^3/s$. In addition, SINFIM stands out for not requiring the pre-processing of the time series to obtain a good performance, which facilitates its application and that would allow a more direct understanding of the predicted runoff values.

However, in this work better results were obtained using a fusion scheme. When it comes to stacking methods, SONFIS model achieved the best performance in validation set in both watersheds studied, higher than the other combination schemes used and single predictions. In contrast to results described by [30, 50, 34, 35], ANN did not have improved results than single models, even though the two watersheds considered had different behavior of runoff levels and variations in the number of stations. This supports considerations made by [51, 52] who suggest that ANN, in some cases, could not model more accurate non-stationary data in Stacking schemes. However, SONFIS model has the advantages of not include a pre-processing step as is proposed by Shoaib et al. [52] who used a wavelet approach for combining the outputs of different rainfall–runoff models. Similarly, statistical approaches such as average and multiple linear regression only improved the performance of some meteorological stations such as “Las Trancas" in Diguillín River or “Curicó" and “Lontué" in Colorado River but always lower than the best single results. This differs from the work of Abrahart and See [30] and Song et al. [48] where these statistical approaches showed good results. Nevertheless, the lower performance agrees with the study of Exbrayat et al. [37] who evaluate as integration scheme statistical methods such as mean, median and multiple linear regression, not being able to find an optimal stacking alternative.

Comparing with ANFIS, in both watersheds, SONFIS showed better performance in the validation set, but not in the Diguillín river training set. Even when ANFIS has been widely applied in rainfall-runoff modeling [16, 95, 15, 96], and showing good results, there are few studies that used this model as stacking scheme. One of its disadvantages is the overfitting when data are overtrained [17]. This may be the reason why a good performance was observed in Diguillín river training set but without achieving an improvement in the performance of the individual models in the validation set. Another aspect to be considered is the number of premises in the fuzzy system has to be pre-defined by user and, this could affect the forecasting. For this, SONFIS model is highlighted because the method developed itself identifies the number of rules necessary to model rainfall-runoff phenomenon [21], reducing the error associated in the parameter selection.

On the other hand, SONFIS improved single predictions of meteorological stations, in both watersheds. This differs from the work of Abrahart and See [30] who evaluated six stacking schemes with different results between watersheds and suggesting that although an integration approach gives better results than single models, they depend of the characteristic of watersheds and their meteorological stations. This represents a great advantage for SONFIS, since, although it has to be evaluated in other watersheds in the region, it can be scalable and be a useful tool in runoff prediction. Nevertheless, the obtained improvement in each catchment is in the same proportion that the model can explain the variability of the phenomenon studied. That is, in the case of the Colorado River, all the stations provided rainfall information, which distribution permit to forecast runoff accurately. Therefore, the improvements observed with the stacking are favored by the good performance obtained in the individual models. However, in the case of the Diguillín River, the best individual prediction had an adjustment of around 83,5%, thus there is an approximate 16,5% unexplained variability. In this case, the stacking reduced this variability a little, but it is clear that, for this watershed, the rainfall information is not enough to model the runoff and it may be necessary to incorporate other variables such as temperature, height or environmental humidity, among others.

Despite the good performance observed, the great variability of the observations presented in the training set affects the flow prediction in the validation set, mainly in terms of low and medium flows. The stacking manages to improve and predict the flow peaks given by some event of precipitation or thaw, but the estimation at low flows tends to be different between the two basins. For the Diguillín river basin, low flows tend to be more regular than in the case of the Colorado river basin, therefore, in the latter, the prediction is more difficult. That is why it is suggested for this basin to consider some variables associated with soil saturation in order to improve the prediction in periods of drought. However, it is necessary to highlight two important aspects. The first of them is that in this work, unlike other investigations such as those of Nourani et al. [44], Asadi et al. [130] Talei et al. [131], Ghose et al. [15] or Panchal et al. [96], which work only with rainy months or with specific precipitation events, we worked with eighteen continuous years and considering all the seasons, so the model is trained considering many variations in rainfall and runoff. Secondly, it must be considered that the years considered in the basins under study have a period affected by the mega-drought that affects the country since approximately 2010 [112]. Therefore, prior to this year, there is a greater amount of precipitation observed and different flow behavior. The presence of this phenomenon therefore affects the prediction in the year of validation.

An aspect to consider corresponds to the selection of the hyperparameters in ANN, ANFIS, and LSTM models. It is well known that the selection of parameters prior to training can affect performance if they are not suitable for modeling with the data set under study [132, 133, 134]. In fact, there are optimization tools such as Harris hawks optimizer [135], particle swarm optimization [136, 135], genetic algorithm [137], or grey wolf optimization [42], among others, that allow the optimal selection of these values to achieve the best performance. However, in this work the selection of hyperparameters was made based on the experience of the researchers and the number of fuzzy rules obtained in SINFIM method, thus it is possible to further improve the performance of these models. Nevertheless, evaluating optimization algorithms for each of the models used and comparing them with the proposed method could be include in a future work.

However, by making a global evaluation, stacking manages to capture the flow behavior and estimate the peaks or continuous increases such as those observed in the Colorado river basin. One of the main advantages of the proposed model is to capture the distribution of rainfall in the basins since it uses information from different meteorological stations. In this regard, it shares similarities with distributed hydrological models, which use the integration of data from various sources. This acquires greater importance in basins whose geospatial distribution does not allow obtaining homogeneous rainfall data, so integrating the information would reduce variability [138, 139] and a better representation of the hydrological cycle phenomena [140]. Therefore, with the proposed stacking framework, not only an improvement in the runoff forecasting is obtained, and a better representation of the rainfall-runoff relationship is also generated, thus avoiding the bias of using a single meteorological station.

Chapter 6

Concluding remarks and future works

Time series data are found in many different areas such as industry, health, agronomy, or climate change. They have a dynamic behavior and their prediction in most cases has high complexity. Such is the case of modeling the relationship between rainfall and runoff, which associated uncertainty is one of the aspects that affect the forecasting. Given this uncertain environment, Neuro-Fuzzy models have emerged as a useful approach, since they take advantage of their ability of neural networks to optimize parameters in a fuzzy system. In this work, a new framework called Self-Identification Neuro-Fuzzy Inference Model (SINFIM) for predicting time series, specifically one day-ahead runoff in a Chilean watershed, was proposed. Specific findings include:

1. SINFIM method has shown better performance than other methods such as ANFIS, ANN, LSTM and the stochastic model ARIMAX for predicting one day-ahead runoff, achieving an adjustment higher than 80% in the watershed under study, and higher that 90% in the watershed validation. In addition, Nash-Sutcliffe efficiency and percent BIAS indicate that SINFIM is a promising model.

2. The proposal, by automatically identifying the lags and the fuzzy rules required in a Neuro-Fuzzy System, allows the reduction of the error and uncertainty associated with user setting parameters.

3. SINFIM method can predict any runoff value regardless of the season of the year, managing to estimate low, medium and peak runoff values. Also, time series datasets do not need a pre-processing step to be included in the model as input.

However, despite the good results of the SINFIM method, the proposed stacking model showed the best performance of predicting one day-ahead runoff compared with single predictions and other stacking approaches. Therefore, this model can improve the prediction by combining the single runoff estimations, regarding the season of the year, the catchment evaluated, and the number of meteorological stations. Thus, an additional advantage is obtained from those already provided by the SINFIM method. Finally, a runoff prediction framework based on the SINFIM model as a single runoff prediction and SONFIS as a stacking method represents a good approach for environmental planning and hydrological processes.

Although the proposed model had a good level of performance in terms of predicting the time series considered, it has certain aspects to be considered for future research:

1. Although the proposed method has good performance indicators in the prediction of flow, it cannot fully explain the variability associated with the phenomenon under study, being mostly observed in the Diguillín river watershed, which showed a maximum of 84% adjustment. However, this adjustment was greater in the basin used in the validation of the method. Thus, the characteristics of the basin affect the prediction of runoff, requiring in some cases to include other covariates that would contribute to improving performance.

2. The main limitation of this study is that no optimization algorithm was applied to obtain the hyperparameters in the models used to compare the SINFIM, and this could affect their performances. Only in the LSTM a factorial design with a grid of options was made. That is why the evaluation of models that allow obtaining the optimal hyperparameters of the methods used will be considered in future research. However, even using these models as runoff prediction approach, the uncertainty in the selection of lags was addressed considering the automatic selection carried out with the SIFAR model.

3. The Neuro-Fuzzy model, although it has a scheme that allows an interpretation based on the linear regression equations obtained in the section of the consequent parameters, the latter generates little interpretable values when the model is adjusted to predict the extreme values. For this reason, it is interesting to evaluate an optimal stabilization mechanism of the model that in turn allows the prediction of an outlier in the time series. On the other hand, for the case study, the model had a better performance in predicting high flow values, with a tendency to overestimate the low and medium values. In this case, it is advisable to evaluate other covariates that could improve the adjustment of the model for these values.

4. Finally, although this study worked with two watersheds (the Diguillín river for the development of the method and the Colorado river for its validation), it is necessary to

apply both developed methods (SINFIM and the proposed stacking) in other basins with different characteristics both in Chile and in other countries to evaluate their performance.

5. Furthermore, it is interesting to evaluate the method in the multi-step ahead runoff prediction scheme, in addition to integrating the model developed with geospatial methods and conceptual hydrological models.

Thus, the challenges that still remain to be solved can be summarized as follows:

1. In the case study, include other covariates that would contribute to improving performance.

2. To develop optimization algorithms that allows to find the most suitable hyperparameters for ANFIS, ANN and LSTM for their comparison with our proposal.

3. To apply and evaluate the proposed framework in other watersheds both in Chile and other countries.

4. To evaluate the proposed framework in the multi-step ahead forecasting.

5. To integrate the proposed framework with geospatial methods and conceptual hydrologycal models.

However, with the results obtained in this study, we can conclude that the proposed method represents a good alternative in runoff one day-ahead prediction, thus constituting a useful tool for professionals involved in environmental planning.

References

[1] Muhammad Faheem Mushtaq, Urooj Akram, Muhammad Aamir, Haseeb Ali, and Muhammad Zulqarnain. Neural network techniques for time series prediction: A review. *JOIV: International Journal on Informatics Visualization*, 3(3):314–320, 2019.

[2] Jose A Guajardo, Richard Weber, and Jaime Miranda. A model updating strategy for predicting time series with seasonal patterns. *Applied Soft Computing*, 10(1):276–283, 2010.

[3] Vahid Nourani, Aida Hosseini Baghanam, Jan Adamowski, and Mekonnen Gebremichael. Using self-organizing maps and wavelet transforms for space–time pre-processing of satellite precipitation and runoff data in neural network based rainfall–runoff modeling. *Journal of hydrology*, 476:228–243, 2013.

[4] Li Xiong and Yue Lu. Hybrid ARIMA-BPNN model for time series prediction of the chinese stock market. In *2017 3rd International Conference on Information Management (ICIM)*, pages 93–97. IEEE, 2017.

[5] KP Agrawal, Sanjay Garg, Shashikant Sharma, Pinkal Patel, and Ayush Bhatnagar. Fusion of statistical and machine learning approaches for time series prediction using earth observation data. *International Journal of Computational Science and Engineering*, 14(3):255–266, 2017.

[6] Muhammad Faheem Mushtaq, Urooj Akram, Adeel Tariq, KHAN Irfan, and Muhammad Zulqarnain. An innovative cognitive architecture for humanoid robot. *International Journal of Natural and Engineering Sciences (IJNES) E-ISSN: 2146-0086*, 8 (2):36–43, 2017.

[7] Seiyed Mossa Hosseini and Najmeh Mahjouri. Integrating support vector regression and a geomorphologic artificial neural network for daily rainfall-runoff modeling. *Applied Soft Computing*, 38:329–345, 2016.

[8] Fatemeh Zahedi and Zahra Zahedi. A review of neuro-fuzzy systems based on intelligent control. *arXiv preprint arXiv:1805.03138*, 2018.

[9] Nitin Kumari and Smita Sunita. Comparison of anns, fuzzy logic and neuro-fuzzy integrated approach for diagnosis of coronary heart disease: a survey. *Int J Comput Sci Mobile Comput*, 2(6):216–224, 2013.

[10] Detlef D Nauck and Andreas Nürnberger. Neuro-fuzzy systems: A short historical review. In *Computational Intelligence in Intelligent Data Analysis*, pages 91–109. Springer, 2013.

[11] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[12] Fi-John Chang and Ya-Ting Chang. Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. *Advances in water resources*, 29(1):1–10, 2006.

[13] Mehmet Tektaş. Weather forecasting using anfis and arima models. *Environmental Research, Engineering and Management*, 51(1):5–10, 2010.

[14] Rajnish K Yadav and Manoj Balakrishnan. Comparative evaluation of arima and anfis for modeling of wireless network traffic time series. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):15, 2014.

[15] DK Ghose, SS Panda, and PC Swain. Prediction and optimization of runoff via anfis and ga. *Alexandria Engineering Journal*, 52(2):209–220, 2013.

[16] K Anusree and KO Varghese. Streamflow prediction of karuvannur river basin using anfis, ann and mnlr models. *Procedia Technology*, 24:101–108, 2016.

[17] Ahmed Al-Hmouz, Jun Shen, Rami Al-Hmouz, and Jun Yan. Modeling and simulation of an adaptive neuro-fuzzy inference system (anfis) for mobile learning. *IEEE Transactions on Learning Technologies*, 5(3):226–237, 2011.

[18] Fang-Mei Tseng and Gwo-Hshiung Tzeng. A fuzzy seasonal arima model for forecasting. *Fuzzy Sets and Systems*, 126(3):367–376, 2002.

[19] Zhongfu Tan, Jinliang Zhang, Jianhui Wang, and Jun Xu. Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models. *Applied Energy*, 87(11):3606–3610, 2010.

[20] AK Parida, Ranjeeta Bisoi, PK Dash, and S Mishra. Times series forecasting using chebyshev functions based locally recurrent neuro-fuzzy information system. *International Journal of Computational Intelligence Systems*, 10(1):375–393, 2017.

[21] Héctor Allende-Cid, Rodrigo Salas, Alejandro Veloz, Claudio Moraga, and Héctor Allende. Sonfis: Structure identification and modeling with a self-organizing neuro-fuzzy inference system. *International Journal of Computational Intelligence Systems*, 9(3):416–432, 2016.

[22] Héctor Allende-Cid, Alejandro Veloz, Rodrigo Salas, Steren Chabert, and Héctor Allende. Self-organizing neuro-fuzzy inference system. In *Iberoamerican Congress on Pattern Recognition*, pages 429–436. Springer, 2008.

[23] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[24] Federico Castanedo. A review of data fusion techniques. *The Scientific World Journal*, 2013, 2013.

[25] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[26] Chen Diao, Bin Wang, and Ning Cai. Data fusion of multivariate time series: Application to noisy 12-lead ecg signals. *Applied Sciences*, 9(1):105, 2019.

[27] Zhiqiang Guo, Huaiqing Wang, Quan Liu, and Jie Yang. A feature fusion based forecasting model for financial time series. *PloS one*, 9(6):e101113, 2014.

[28] Ko Ming Hsiao, Geoff West, Svetha Venkatesh, and Mohan Kumar. Temporal data fusion in multisensor systems using dynamic time warping. In *Sensor Fusion 2005: Workshop on Information Fusion and Dissemination in Wireless Sensor Networks*, pages 1–9. IEEE, 2005.

[29] Li-Wei Ko, Yi-Chen Lu, Humberto Bustince, Yu-Cheng Chang, Yang Chang, Javier Ferandez, Yu-Kai Wang, Jose Antonio Sanz, Gracaliz Pereira Dimuro, and Chin-Teng Lin. Multimodal fuzzy fusion for enhancing the motor-imagery-based brain computer interface. *IEEE Computational Intelligence Magazine*, 14(1):96–106, 2019.

[30] Robert J Abrahart and Linda See. Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. 2002.

[31] Hristos Tyralis, Georgia Papacharalampous, Apostolos Burnetas, and Andreas Langousis. Hydrological post-processing using stacked generalization of quantile regression algorithms: Large-scale application over conus. *Journal of Hydrology*, 577: 123957, 2019.

[32] Ossama Abdel-Hamid, Li Deng, and Dong Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech*, volume 2013, pages 1173–5, 2013.

[33] Mahboubeh Ghazali, Tooraj Honar, and Mohammad Reza Nikoo. A fusion-based neural network methodology for monthly reservoir inflow prediction using modis products. *Hydrological Sciences Journal*, 63(15-16):2076–2096, 2018.

[34] Fereshteh Modaresi, Shahab Araghinejad, and Kumars Ebrahimi. Selected model fusion: an approach for improving the accuracy of monthly streamflow forecasting. *Journal of Hydroinformatics*, 20(4):917–933, 2018.

[35] Yaser Tahmasebi Biragani, Farhad Yazdandoost, and Hossein Ghalkhani. Flood forecasting using artificial neural networks: an application of multi-model data fusion technique. *Journal of Hydraulic Structures*, 2(2):62–73, 2016.

[36] Phanida Phukoetphim, Asaad Y Shamseldin, and Keith Adams. Multimodel approach using neural networks and symbolic regression to combine the estimated discharges of rainfall-runoff models. *Journal of Hydrologic Engineering*, 21(8):04016022, 2016.

[37] J-F Exbrayat, NR Viney, Jan Seibert, H-G Frede, and L Breuer. Multi-model data fusion as a tool for pub: example in a swedish mesoscale catchment. *Advances in Geosciences*, 29:43–50, 2011.

[38] Lotfy A Zadeh. Calculus of fuzzy restrictions. In *Fuzzy sets and their applications to cognitive and decision processes*, pages 1–39. Elsevier, 1975.

[39] Ratnadip Adhikari and Ramesh K Agrawal. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*, 2013.

[40] Burcin Seyda Corba, Erol Egrioglu, and Ali Zafer Dalar. Ar–arch type artificial neural network for forecasting. *Neural Processing Letters*, 51(1):819–836, 2020.

[41] Romina Torres, Rodrigo Salas, et al. Self-adaptive fuzzy qos-driven web service discovery. In *2011 IEEE International Conference on Services Computing*, pages 64–71. IEEE, 2011.

[42] Majid Dehghani, Akram Seifi, and Hossien Riahi-Madvar. Novel forecasting models for immediate-short-term to long-term influent flow prediction by combining ANFIS and grey wolf optimization. *Journal of Hydrology*, 576:698–725, 2019.

[43] N Bartoletti, F Casagli, Stefano Marsili-Libelli, A Nardi, and L Palandri. Data-driven rainfall/runoff modelling based on a neuro-fuzzy inference system. *Environmental modelling & software*, 106:35–47, 2018.

[44] Vahid Nourani, Özgür Kisi, and Mehdi Komasi. Two hybrid artificial intelligence approaches for modeling rainfall–runoff process. *Journal of Hydrology*, 402(1-2): 41–59, 2011.

[45] Amitabha Nath, Fisokuhle Mthethwa, and Goutam Saha. Runoff estimation using modified adaptive neuro-fuzzy inference system. *Environmental Engineering Research*, 2019.

[46] H Allende, C Moraga, R Ñanculef, and R Salas. Ensembles methods for machine learning pattern recognition and machine vision. *Series Information Sciences & Tecnology. In honor and memory of Prof. KS. Fu*, pages 247–261, 2010.

[47] Asaad Y Shamseldin, Kieran M O'Connor, and Ahmed E Nasr. A comparative study of three neural network forecast combination methods for simulated river flows of different rainfall—runoff models. *Hydrological Sciences Journal*, 52(5):896–916, 2007.

[48] Peibing Song, Weifeng Liu, Jiahui Sun, Chao Wang, Lingzhong Kong, Zhenxue Nong, Xiaohui Lei, and Hao Wang. Annual runoff forecasting based on multi-model information fusion and residual error correction in the ganjiang river basin. *Water*, 12 (8):2086, 2020.

[49] Xiao-yun Chen, Kwok-wing Chau, and Wen-chuan Wang. A novel hybrid neural network based on continuity equation and fuzzy pattern-recognition for downstream daily river discharge forecasting. *Journal of hydroinformatics*, 17(5):733–744, 2015.

[50] Yun Bai, Jingjing Xie, Xiaoxue Wang, and Chuan Li. Model fusion approach for monthly reservoir inflow forecasting. *Journal of Hydroinformatics*, 18(4):634–650, 2016.

[51] Barbara Cannas, Alessandra Fanni, Linda See, and Giuliana Sias. Data preprocessing for river flow forecasting using neural networks: wavelet transforms and data partitioning. *Physics and Chemistry of the Earth, Parts A/B/C*, 31(18):1164–1171, 2006.

[52] Muhammad Shoaib, Asaad Y Shamseldin, Sher Khan, Mudasser Muneer Khan, Zahid Mahmood Khan, and Bruce W Melville. A wavelet based approach for combining the outputs of different rainfall–runoff models. *Stochastic environmental research and risk assessment*, 32(1):155–168, 2018.

[53] Eliana Vivas, Héctor Allende-Cid, Rodrigo Salas, and Lelys Bravo. Polynomial and wavelet-type transfer function models to improve Fisheries' landing forecasting with exogenous variables. *Entropy*, 21(11):1082, 2019.

[54] Lihua Xiong, Asaad Y Shamseldin, and Kieran M O'connor. A non-linear combination of the forecasts of rainfall-runoff models by the first-order takagi–sugeno fuzzy system. *Journal of hydrology*, 245(1-4):196–217, 2001.

[55] PC Nayak, KP Sudheer, and SK Jain. Rainfall-runoff modeling through hybrid intelligent system. *Water Resources Research*, 43(7), 2007.

[56] Sengwee Toh, Robert Wellman, R Yates Coley, Casie Horgan, Jessica Sturtevant, Erick Moyneur, Cheri Janning, Roy Pardee, Karen J Coleman, David Arterburn, et al. Combining distributed regression and propensity scores: a doubly privacy-protecting analytic method for multicenter research. *Clinical epidemiology*, 10:1773, 2018.

[57] Daniele Masseroni, Alessio Cislaghi, Stefania Camici, Christian Massari, and Luca Brocca. A reliable rainfall–runoff model for flood forecasting: review and application to a semi-urbanized watershed at high flood risk in italy. *Hydrology Research*, 48(3): 726–740, 2017.

[58] ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. Artificial neural networks in hydrology. ii: Hydrologic applications. *Journal of Hydrologic Engineering*, 5(2):124–137, 2000.

[59] Rana Muhammad Adnan, Zhongmin Liang, Slavisa Trajkovic, Mohammad Zounemat-Kermani, Binquan Li, and Ozgur Kisi. Daily streamflow prediction using optimally pruned extreme learning machine. *Journal of Hydrology*, 577:123981, 2019.

[60] Víctor Parra, Patricio Fuentes-Aguilera, and Enrique Muñoz. Identifying advantages and drawbacks of two hydrological models based on a sensitivity analysis: a study in two chilean watersheds. *Hydrological sciences journal*, 63(12):1831–1843, 2018.

[61] William WS Wei. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. 2006.

[62] Kun Il Park and Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018.

[63] Jan Grandell. Time series analysis lecture notes. *KTH Stockholm University*, 2015.

[64] Rajesh Wadhvani et al. Review on various models for time series forecasting. In *2017 International Conference on Inventive Computing and Informatics (ICICI)*, pages 405–410. IEEE, 2017.

[65] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

[66] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[67] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[68] Arash Azadegan, Lejla Porobic, Sepehr Ghazinoory, Parvaneh Samouei, and Amir Saman Kheirkhah. Fuzzy logic in manufacturing: A review of literature and a specialized application. *International Journal of Production Economics*, 132(2): 258–270, 2011.

[69] Gilberto Arenas-Díaz and Edgar René Ramírez Lamus. Medidas difusas e integrales difusas. *Universitas Scientiarum*, 18(1):7–32, 2013.

[70] Michio Sugeno. Theory of fuzzy integrals and its applications. *Doct. Thesis, Tokyo Institute of technology*, 1974.

[71] Ronald R Yager. Using fuzzy measures for modeling human perception of uncertainty in artificial intelligence. *Engineering Applications of Artificial Intelligence*, 87:103228, 2020.

[72] Ronald R Yager. Uncertainty modeling using fuzzy measures. *Knowledge-Based Systems*, 92:1–8, 2016.

[73] ZX Guo and WK Wong. Fundamentals of artificial intelligence techniques for apparel management applications. *Optimizing Decision Making in the Apparel Supply Chain Using Artificial Intelligence (AI): from Production to Retail*, pages 13–40, 2013.

[74] Romina Torres, Rodrigo Salas, and Hernan Astudillo. Time-based hesitant fuzzy information aggregation approach for decision-making problems. *International Journal of Intelligent Systems*, 29(6):579–595, 2014.

[75] Tomohiro Takagi and Michio Sugeno. Derivation of fuzzy control rules from human operator's control actions. *IFAC Proceedings Volumes*, 16(13):55–60, 1983.

[76] Claudio Moraga and Rodrigo Salas. A new aspect for the optimization of fuzzy if-then rules. In *35th International Symposium on Multiple-Valued Logic (ISMVL'05)*, pages 160–165. IEEE, 2005.

[77] Alejandro Veloz, Héctor AllendeCid, Héctor Allende, Claudio Moraga, and Rodrigo Salas. A flexible neuro-fuzzy autoregressive technique for non-linear time series forecasting. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 22–29. Springer Berlin Heidelberg, 2009.

[78] Saman K Halgamuge and Manfred Glesner. A fuzzy-neural approach for pattern classification with the generation of rules based on supervised learning. In *Les réseaux neuro-mimétiques et leurs applications. Journées internationales*, pages 165–173, 1992.

[79] Detlef Nauck and Rudolf Kruse. Nefclass-a neuro-fuzzy approach for the classification of data. *Deaton, Dave Oppenheim, and Jim Hightower, editors, Applied Computing*, pages 26–28, 1995.

[80] Nadine Tschichold-Gürman. Generation and improvement of fuzzy classifiers with incremental learning using fuzzy rulenet. In *Proceedings of the 1995 ACM symposium on Applied computing*, pages 466–470. ACM, 1995.

[81] C-T Lin and C. S. George Lee. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on computers*, (12):1320–1336, 1991.

[82] Detlef Nauck and Rudolf Kruse. Neuro-fuzzy systems for function approximation. *Fuzzy sets and systems*, 101(2):261–271, 1999.

[83] Sandra M Sulzberger, N Tschichold-Gurman, and Sjur J Vestli. Fun: Optimization of fuzzy rule based systems using neural networks. In *IEEE International Conference on Neural Networks*, pages 312–316. IEEE, 1993.

[84] Andreas D Lattner, Alexander Grimme, and Ingo J Timm. An evaluation of meta learning and distribution strategies in distributed machine learning. In *Proceedings of the European Conference on Data Mining*, 2010.

[85] Héctor Allende-Cid, Héctor Allende, Raúl Monge, and Claudio Moraga. Discrete neighborhood representations and modified stacked generalization methods for distributed regression. *J. UCS*, 21(6):842–855, 2015.

[86] Zheng-Chu Guo, Lei Shi, and Qiang Wu. Learning theory of distributed regression with bias corrected regularization kernel network. *The Journal of Machine Learning Research*, 18(1):4237–4261, 2017.

[87] Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

[88] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[89] Alejandro Veloz, Rodrigo Salas, Héctor Allende-Cid, Héctor Allende, and Claudio Moraga. Identification of lags in nonlinear autoregressive time series using a flexible fuzzy model. *Neural Processing Letters*, 43(3):641–666, 2016.

[90] Keith J Beven. *Rainfall-runoff modelling: the primer*. John Wiley & Sons, 2011.

[91] Saritha Padiyedath Gopalan, Akira Kawamura, Hideo Amaguchi, Tadakatsu Takasaki, and Gubash Azhikodan. A bootstrap approach for the parameter uncertainty of an urban-specific rainfall-runoff model. *Journal of Hydrology*, 579:124195, 2019.

[92] Wouter JM Knoben, Jim E Freer, Keirnan JA Fowler, Murray C Peel, and Ross A Woods. Modular assessment of rainfall–runoff models toolbox (marrmot) v1. 2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations. *Geoscientific Model Development*, 12(6):2463–2480, 2019.

[93] Günter Blöschl, Marc FP Bierkens, Antonio Chambel, Christophe Cudennec, Georgia Destouni, Aldo Fiori, James W Kirchner, Jeffrey J McDonnell, Hubert HG Savenije, Murugesu Sivapalan, et al. Twenty-three unsolved problems in hydrology (uph)–a community perspective. *Hydrological Sciences Journal*, 64(10):1141–1158, 2019.

[94] Zekai Sen and Abdusselam Altunkaynak. Fuzzy awakening in rainfall-runoff modeling. *Nordic Hydrology*, 35(1):31–43, 2004.

[95] V Jothiprakash, RB Magar, and Sunil Kalkutki. Rainfall–runoff models using adaptive neuro–fuzzy inference system (anfis) for an intermittent river. *Int J Artif Intell*, 3:1–23, 2009.

[96] Ratansharan Panchal, TMV Suryanarayana, and FP Parekh. Adaptive neuro-fuzzy inference system for rainfall-runoff modeling. *International Journal of Engineering Research and Applications*, 4:202–206, 2014.

[97] Yanlai Zhou, Shenglian Guo, and Fi-John Chang. Explore an evolutionary recurrent anfis for modelling multi-step-ahead flood forecasts. *Journal of Hydrology*, 570: 343–355, 2019.

[98] Armin Azad, Saeed Farzin, Hamed Kashi, Hadi Sanikhani, Hojat Karami, and Ozgur Kisi. Prediction of river flow using hybrid neuro-fuzzy models. *Arabian Journal of Geosciences*, 11(22):718, 2018.

[99] Tuo Xie, Gang Zhang, Jinwang Hou, Jiancang Xie, Meng Lv, and Fuchao Liu. Hybrid forecasting model for non-stationary daily runoff series: A case study in the Han river basin, China. *Journal of Hydrology*, 577:123915, 2019.

[100] Thi TT Tran, Ngo NH Giang, Hieu N Duong, Hien T Nguyen, Tran Van Hoai, and Vu Van Nghi. A comprehensive study on predicting river runoff. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pages 251–256. IEEE, 2017.

[101] Hossein Mojaddadi Rizeei, Biswajeet Pradhan, and Maryam Adel Saharkhiz. Surface runoff prediction regarding LULC and climate dynamics using coupled LSTM, optimized ARIMA, and GIS-based SCS-CN models in tropical region. *Arabian Journal of Geosciences*, 11(3):53, 2018.

[102] Tak Kwin Chang, Amin Talei, Sina Alaghmand, and Melanie Po-Leen Ooi. Choice of rainfall inputs for event-based rainfall-runoff modeling in a catchment with multiple rainfall stations using data-driven techniques. *Journal of Hydrology*, 545:100–108, 2017.

[103] V Nourani, TR Khanghah, and A Hossieni Baghanam. Application of entropy concept for input selection of wavelet-ANN based rainfall-runoff modeling. *Journal of Environmental Informatics*, 26(1):52–70, 2015.

[104] Robert J May, Holger R Maier, Graeme C Dandy, and TMK Gayani Fernando. Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modelling & Software*, 23(10-11):1312–1326, 2008.

[105] Anita Singh, Anurag Malik, Anil Kumar, and Ozgur Kisi. Rainfall-runoff modeling in hilly watershed using heuristic approaches with gamma test. *Arabian Journal of Geosciences*, 11(11):261, 2018.

[106] R Remesan, Michaela Bray, and J Mathew. Application of PCA and clustering methods in input selection of hybrid runoff models. *Journal of Environmental Informatics*, 31 (2), 2018.

[107] Riccardo Taormina and Kwok-Wing Chau. Data-driven input variable selection for rainfall–runoff modeling using binary-coded particle swarm optimization and extreme learning machines. *Journal of hydrology*, 529:1617–1632, 2015.

[108] Esteban Figueroa-Rodrigo Gómez-Fernando Sánchez and José Luis. Plan de manejo cuenca río diguillín. *Diagnostico*, 40:41.

[109] Ian Stemberga. Evaluación del uso de información satelital de precipitaciones para diferentes métodos de infiltración en la cuenca del río diguillín en san lorenzo, región del biobío, chile, 2017.

[110] H Niemeyer Fernández. Hoyas hidrográficas de Chile, Sexta Región. *Ministerio de Obras Públicas, Dirección general de Aguas*, pages 350–372, 1980.

[111] Cade-Idepe Consultores en Ingeniería. Diagnóstico y clasificación de los cursos y cuerpos de agua según objetivos de calidad-cuenca río Mataquito, 2004.

[112] René D Garreaud, Camila Alvarez-Garreton, Jonathan Barichivich, Juan Pablo Boisier, Duncan Christie, Mauricio Galleguillos, Carlos LeQuesne, James McPhee, and Mauricio Zambrano-Bigiarini. The 2010-2015 megadrought in central chile: impacts on regional hydroclimate and vegetation. 2017.

[113] Akhil Garg and Kang Tai. Comparison of statistical and machine learning methods in modelling of data with multicollinearity. *International Journal of Modelling, Identification and Control*, 18(4):295–312, 2013.

[114] CP Obite, NP Olewuezi, GU Ugwuanyim, and DC Bartholomew. Multicollinearity effect in regression analysis: A feed forward artificial neural network approach. *Asian Journal of Probability and Statistics*, pages 22–33, 2020.

[115] P Molnar. Calibration. watershed modelling, ss 2011. *Institute of Environmental Engineering, Chair of Hydrology and Water Resources Management, ETH Zürich, Switzerland*, 2011.

[116] Daniel N Moriasi, Jeffrey G Arnold, Michael W Van Liew, Ronald L Bingner, R Daren Harmel, and Tamie L Veith. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900, 2007.

[117] Wouter JM Knoben, Jim E Freer, and Ross A Woods. Inherent benchmark or not? comparing Nash–Sutcliffe and Kling–Gupta efficiency scores. *Hydrology and Earth System Sciences*, 23(10):4323–4331, 2019.

[118] Muhammad Shoaib, Asaad Y Shamseldin, and Bruce W Melville. Comparative study of different wavelet based neural network models for rainfall–runoff modeling. *Journal of Hydrology*, 515:47–58, 2014.

[119] J Behmanesh and S Ayashm. Rainfall-runoff modeling in the turkey river using numerical and regression methods. *Journal of Fundamental and Applied Sciences*, 7 (1):91–102, 2015.

[120] P Sundara Kumar, TV Praveen, and M Anjanaya Prasad. Artificial neural network model for rainfall-runoff-a case study. *International Journal of Hybrid Information Technology*, 9(3):263–272, 2016.

[121] Ali Osman Pektaş and H Kerem Cigizoglu. Ann hybrid model versus arima and arimax models of runoff coefficient. *Journal of hydrology*, 500:21–36, 2013.

[122] Vahid Nourani, Masoud Mehrvand, and Aida Hosseini Baghanam. Implication of som-ann based clustering for multistation rainfall-runoff modeling. *Journal of Urban and Environmental Engineering*, 8(2):198–210, 2014.

[123] Caihong Hu, Qiang Wu, Hui Li, Shengqi Jian, Nan Li, and Zhengzheng Lou. Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, 10(11):1543, 2018.

[124] Frederik Kratzert, Daniel Klotz, Claire Brenner, Karsten Schulz, and Mathew Herrnegger. Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018.

[125] Xiaoli Li, Zhenlong Du, and Guomei Song. A method of rainfall runoff forecasting based on deep convolution neural networks. In *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, pages 304–310. IEEE, 2018.

[126] Song Pham Van, Hoang Minh Le, Dat Vi Thanh, Thanh Duc Dang, Ho Huu Loc, and Duong Tran Anh. Deep learning convolutional neural network in rainfall–runoff modelling. *Journal of Hydroinformatics*, 22(3):541–561, 2020.

[127] I-Feng Kao, Yanlai Zhou, Li-Chiu Chang, and Fi-John Chang. Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, page 124631, 2020.

[128] Jean Marçais and Jean-Raynald de Dreuzy. Prospective interest of deep learning for hydrological inference. *Groundwater*, 55:699–692, 2017.

[129] Yerel Morales, Marvin Querales, Harvey Rosas, Héctor Allende-Cid, and Rodrigo Salas. A self-identification neuro-fuzzy inference framework for modeling rainfall-runoff in a chilean watershed. *Journal of Hydrology*, page In press, 2020.

[130] Shahrokh Asadi, Jamal Shahrabi, Peyman Abbaszadeh, and Shabnam Tabanmehr. A new hybrid artificial neural networks for rainfall–runoff process modeling. *Neurocomputing*, 121:470–480, 2013.

[131] Amin Talei, Lloyd Hock Chye Chua, and Chai Quek. A novel application of a neuro-fuzzy computational technique in event-based rainfall–runoff modeling. *Expert Systems with Applications*, 37(12):7456–7468, 2010.

[132] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, 20(53):1–32, 2019.

[133] Jan N Van Rijn and Frank Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2367–2376, 2018.

[134] Romina Torres, Rodrigo Salas, Héctor Allende, and Claudio Moraga. Robust expectation maximization learning algorithm for mixture of experts. In *International Work-Conference on Artificial Neural Networks*, pages 238–245. Springer, 2003.

[135] Yazid Tikhamarine, Doudja Souag-Gamane, Ali Najah Ahmed, Saad Sh Sammen, Ozgur Kisi, Yuk Feng Huang, and Ahmed El-Shafie. Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs. particle swarm optimization. *Journal of Hydrology*, page 125133, 2020.

[136] Zhong-kai Feng, Wen-jing Niu, Zheng-yang Tang, Zhi-qiang Jiang, Yang Xu, Yi Liu, and Hai-rong Zhang. Monthly runoff time series prediction by variational mode decomposition and support vector machine based on quantum-behaved particle swarm optimization. *Journal of Hydrology*, 583:124627, 2020.

[137] Esmaeel Dodangeh, Mahdi Panahi, Fatemeh Rezaie, Saro Lee, Dieu Tien Bui, Chang-Wook Lee, and Biswajeet Pradhan. Novel hybrid intelligence models for flood-susceptibility prediction: Meta optimization of the GMDH and SVR models with the genetic algorithm and harmony search. *Journal of Hydrology*, 590:125423, 2020.

[138] Juraj Bartok, Ondrej Habala, Peter Bednar, Martin Gazak, and Ladislav Hluchỳ. Data mining and integration for predicting significant meteorological phenomena. *Procedia Computer Science*, 1(1):37–46, 2010.

[139] Baxter E Vieux. Distributed hydrologic modeling using gis. In *Distributed Hydrologic Modeling Using GIS*, pages 1–17. Springer, 2001.

[140] Sara Patricia Ibarra-Zavaleta, Rosario Landgrave, Rabindranarth Romero-López, Annie Poulin, and Raúl Arango-Miranda. Distributed hydrological modeling: Determination of theoretical hydraulic potential & streamflow simulation of extreme hydrometeorological events. *Water*, 9(8):602, 2017.

Appendix A

Single runoff forecasting using the ARIMAX model

This appendix shows the development of the estimation of the ARIMAX model for runoff forecasting. As the proposal was developed, two experiments were carried out: considering three and one year as validation set. The results are shown as follows.

Experiment 1: using last three years as validation set

To apply the ARIMAX model to forecast one day-ahead runoff, first, the stationarity of the time series was evaluated using the autocorrelation function (ACF), Partial ACF (PACF) and the Augmented Dickey-Fuller test. Figure A.1 shows that the ACF values are higher than the confidence interval, confirming that there is a possible linear association between observations separated by larger lags, indicating a possible non-stationarity, which was confirmed with the PACF and the Dickey-Fuller test, obtaining a p-value =0.01 and a Dickey-Fuller= -24.2968.

An ARIMA(2,0,2) was estimated as the best model. The estimated coefficients were: $\phi_1 = 1.54 \pm 0.03$, $\phi_2 = -0.54 \pm 0.04$, $\theta_1 = -0.9 \pm 0.03$, $\theta_2 = -0.04 \pm 0.02$, $constant = 15.3 \pm 1.8$ and regressor coefficient = 0.1 ± 0.02. With the exception of θ_2 (p-value=0.07621), all coefficients were significant at 95% confidence. The σ^2 estimated was $358.9(m_3/s)^2$, $AIC = 50.956.61$ and $BIC = 51003.32$. As can be observed in Table A.1, the estimated model has low performance in the prediction of runoff, and fails in the estimation of the peak values, making an estimate approximately constant at low values instead (Figure A.2).

Experiment 2: using last year as validation set

As well as Experiment 1, the stationarity of the time series was evaluated using the autocorrelation function (ACF), Partial ACF (PACF) and the Augmented Dickey-Fuller

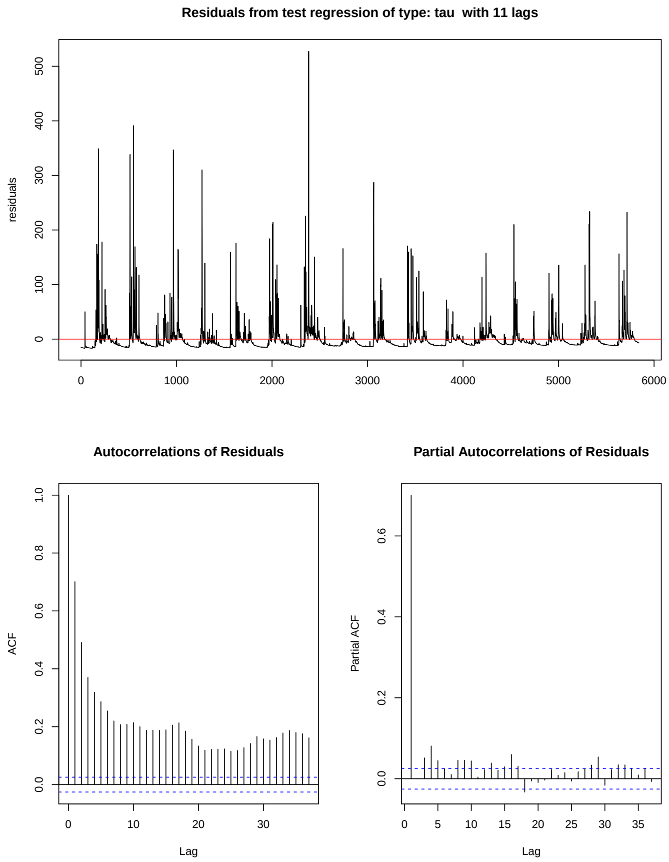


Fig. A.1 Autocorrelation analysis of runoff time series from Diguillín river watershed considering the first fifteen years. There is a linear association between observations separated by larger lags, indicating a possible non-stationarity

test. Figure A.3 shows, that ACF values are higher than confidence bounds, with a possible linear association between observations separated by larger lags, and indicating a possible non-stationarity, which was confirmed with the PACF and the Dickey-Fuller test, obtaining a p-value =0.01 and a Dickey-Fuller= -25.3194.

An ARIMA(1,1,2) was estimated as the best model. The estimated coefficients were: $\phi_1 = 0.5733 \pm 0.02$, $\theta_1 = -0.9 \pm 0.02$, $\theta_2 = -0.03 \pm 0.02$, and regressor coefficient = 0.1 ± 0.02. With the exception of θ_2 (p-value=0.07621), all coefficients were significant at 95% confidence. The σ^2 estimated was $331.9(m_3/s)^2, AIC = 55838.45$ and $BIC = 55872.31$. As observed in experiment 1, the model has low performance (Table A.2) and makes low runoff predictions with a constant trend, failing to capture the variability of the studied phenomenon (Figure A.4).

Table A.1 Stochastic model results using the last three year as a validation set. In training and validation sets, a low performance to predict one day-ahead runoff was observed.

Training					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
ARIMAX(2,0,2)	358.5405	0.5038	-0.0372	-0.0281	0.5767
Validation					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
ARIMAX(2,0,2)	174.0327	0.0091	-5.0506	-48.86	-0.2178

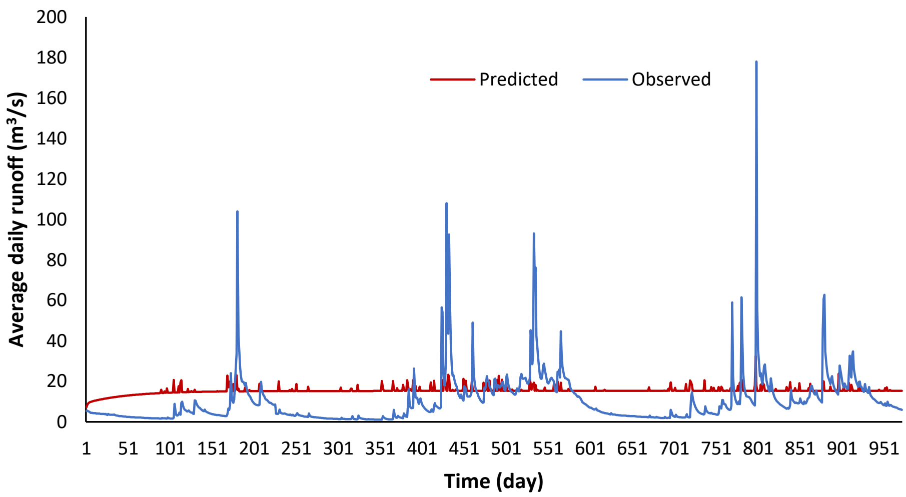


Fig. A.2 Runoff forecasting through ARIMAX model using last three years as validation set. The model fails to capture the variability of the phenomenon studied, maintaining an approximately constant flow prediction.

Table A.2 Stochastic model results using the last year as a validation set. In training and validation sets, a low performance to predict one day-ahead runoff was observed.

Training					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
ARIMAX(1,1,2)	331.6033	0.5097	0.0721	0.0617	0.6046
Validation					
Model	**MSE**	R^2	**NSE**	**PBIAS**	**KGE**
ARIMAX(1,1,2)	167.07	0.0287	-58.2	9.48	-0.1963

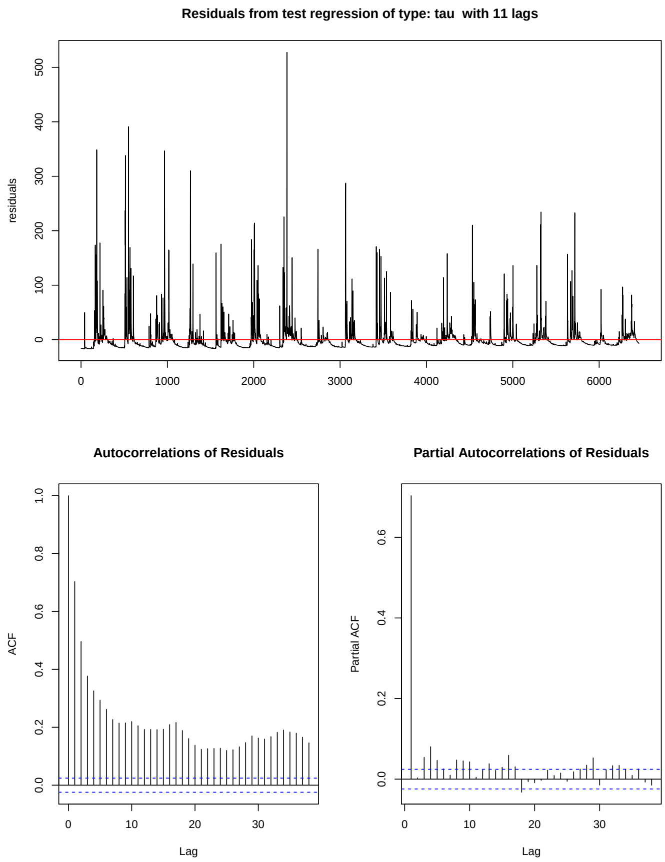


Fig. A.3 Autocorrelation analysis of runoff time series from Diguillín river watershed considering the first seventeen years. There is a possible linear association between observations separated by larger lags, indicating a possible non-stationarity

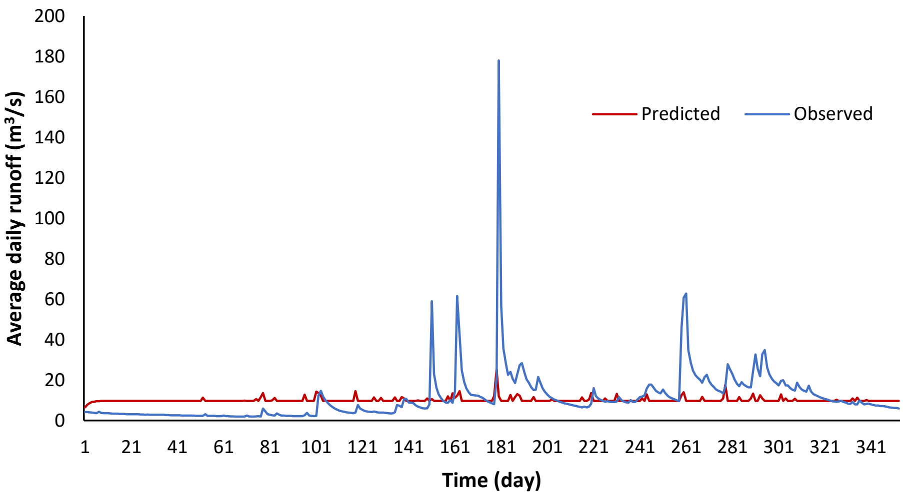


Fig. A.4 Runoff forecasting through ARIMAX model using last year as validation set. The model fails to capture the variability of the phenomenon studied.