ADONiS—Adaptive Online Nonsingleton Fuzzy Logic Systems

Direnc Pekaslan, Student Member, IEEE, Christian Wagner, Senior Member, IEEE, and Jonathan M. Garibaldi, Senior Member, IEEE

Abstract—Nonsingleton fuzzy logic systems (NSFLSs) have the potential to capture and handle input noise within the design of input fuzzy sets (FSs). In this article, we propose an online learning method that utilizes a sequence of observations to continuously update the input FSs of an NSFLS, thus providing an improved capacity to deal with variations in the level of input-affecting noise, common in real-world applications. The method removes the requirement for both a priori knowledge of noise levels and relying on offline training procedures to define input FS parameters. To the best of our knowledge, the proposed ADaptive, Online Nonsingleton (ADONiS) fuzzy logic system (FLS) framework represents the first end-to-end framework to adaptively configure nonsingleton input FSs. The latter is achieved through online uncertainty detection applied to a sliding window of observations. Since real-world environments are influenced by a broad range of noise sources, which can vary greatly in magnitude over time, the proposed technique for combining online determination of noise levels with associated adaptation of input FSs provides an efficient and effective solution which elegantly models input uncertainty in the FLS’s input FSs, without requiring changes in any other part (e.g., antecedents, rules or consequents) of the FLS. In this article, two common chaotic time series (Mackey–Glass, Lorenz) are used to perform prediction experiments to demonstrate and evaluate the proposed framework. Results indicate that the proposed adaptive NSFLS framework provides significant advantages, particularly in environments that include high variation in noise levels, which are common in real-world applications.

Index Terms—Adaptive input, firing strength, nonsingleton fuzzy logic system (NSFLS), noise estimation, online learning.

I. INTRODUCTION

NOISE (uncertainty) can be described as unwanted disruptive influences in data and a broad range of noise sources are common in real-world environments. These sources can vary greatly in magnitude, with each source may contribute vastly different levels of noise. Gaining insight into the level of noise is a critical part of many applications. Thus, noise studies are attracting considerable interest, particularly in terms of accurate estimation and/or removal of noise from data. However, because of the variation in the noise levels, accurate estimation becomes a challenging task, and, even when estimation is accurate, noise removal algorithms may still not achieve the optimal performance [1]. In light of this, methods have become established to handle noise within the system that may provide performance benefits in decision-making applications.

Fuzzy set (FS) theory was first introduced by Zadeh [2] and is designed to involve intermediate degrees by means of membership functions (MFs), which provided the basis for fuzzy logic systems (FLSs). Thus, FLSs are considered as robust systems in decision making under uncertainty [3] and generally comprise three essential steps; fuzzification, inferencing, and defuzzification.

The fuzzification step maps (commonly) crisp input values singleton or nonsingleton FSs. When input data are corrupted by noise, nonsingleton FSs are useful to model this noise and, generally, have the potential to deliver better results compared to (otherwise identical) singleton fuzzy logic systems (SFLSs). [3]–[18]. In nonsingleton fuzzy logic system (NSFLS) design, the input FS models are commonly constructed based on the level of uncertainty or noise affecting the system inputs. In the literature, this is achieved through two key approaches.

1) A priori knowledge about the uncertainty level, for example based on expert insight, is utilized to define the parameters of NSFLSs offline. While efficient, this approach is invariably dependent on the availability of this priori knowledge, and does not allow the modeling of levels of uncertainty that vary over time.

2) Training procedures are implemented to optimize the fuzzification stage’s parameters in an offline setting using existing data. While such methods avoid the need for a priori insight, they are dependent on the availability of a training dataset that accurately reflects all potential real-world operating conditions, and, are generally limited to the selection of one type of the FS model to capture the input uncertainty. Both of these aspects tend to prevent NSFLSs dependent on offline optimization from being able to adapt to the breadth of changing uncertainty levels inherent to real-world applications.

Overall, due to the varying circumstances in real-world applications, determining the parameters in an offline manners is not an efficient approach to capture input uncertainty in

Manuscript received March 13, 2019; revised June 25, 2019; accepted August 2, 2019. Date of publication August 7, 2019; date of current version October 6, 2020. This work was supported by the School of Computer Science Studentship from the University of Nottingham. (Corresponding author: Direnc Pekaslan.)

The authors are with the Intelligent Modelling and Analysis Group and Lab for Uncertainty in Data and Decision Making, School of Computer Science, University of Nottingham, NG7 2RD Nottingham, U.K. (e-mail: direnc.pekaslan@nottingham.ac.uk; christian.wagner@nottingham.ac.uk; jon.garibaldi@nottingham.ac.uk).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2019.2933787
different, nonstationary settings. Recently, Pekaslan et al. [19] introduced a noise estimation strategy that provides a pathway to parameterize input FSs in respect to the level of noise detected over a specified period of time (a window). However, analysis and time-series prediction experiments in [19] were limited to designing input FSs retrospectively, while the level of noise injected into the given time series was kept stable for all experiments. While the latter facilitates the experiments, it does not reflect real-world conditions where noise and uncertainty vary, at times substantially, over time.

Other approaches have also been developed to deal with variation in uncertainty affecting systems, such as evolving fuzzy systems [20]–[25]. Here, FLS parameters, from antecedent and consequent FSs, to rule bases are adjusted through some procedures. While such approaches are effective, they are not designed to specifically reflect changes in input uncertainty in the inputs FS with potentially negative effects on FLS interpretability (e.g., changes in input uncertainty result in changes to the antecedents). The approach put forward in this article is specifically designed to constrain parameter changes in the (NS)FLSs as a result of changes in the uncertainty affecting a system’s inputs—to the input FSs.

In this article, we propose a complete ADaptive, ONline nonsingleton (ADONiS) framework that incorporates online uncertainty detection and associated parameterization of the nonsingleton input FSs. The proposed approach avoids both the need for a priori knowledge of the uncertainty levels experienced at runtime and the need for offline training, while providing the means for systems to continuously adapt to changing levels of uncertainty. Specifically, in the proposed approach, input FS parameters are continuously adapted based on information gained from an uncertainty level estimation process that iteratively estimates uncertainty levels over a sequence of recent observations.

As time-series forecasting provides an ideal test bed for the systematic evaluation (offering the potential to accurately control the levels of uncertainty/noise affecting system inputs at any given time) of techniques designed to deal with input uncertainty, we demonstrate/explore the proposed framework in the context of time-series prediction. In the experiments, the injection of noise into two well-known chaotic time series [Mackey–Glass (MG) and Lorenz] is used in order to mimic real-world circumstances, where uncertainty levels vary over time.

The structure of this article is as follows. Section II provides background information on noise generation/estimation, FSs and related literature. In Section III, the methodology of this article is provided. Section IV explains the experimental environment. This is followed by the results and discussion in Section V. Finally, conclusions and a reflection on valuable future work are provided in Section VII.

II. BACKGROUND

This section gives a brief overview of key concepts in respect to the noise estimation technique used in this article. Also, singleton and nonsingleton FSs and approaches for their comparison are introduced in respect to the literature.

A. Noise Generation

In the literature, noise is generally measured by the signal-to-noise ratio (SNR) that is often expressed in decibels (dB). While lower noise levels are represented by high SNR values, a high level of noise is captured by low SNR values. The SNR calculation is implemented by

$$\text{SNR} = 10 \times \log\left(\frac{\sigma_{n, f}^2}{\sigma_n^2}\right) \tag{1}$$

where $\sigma_{n, f}$ is the standard deviation of the noise-free dataset and $\sigma_n$ is the noise level.

Often, it is assumed that the values SNR and $\sigma_{n, f}$ are known a priori. Thus, when $\sigma_n$ is sought, (1) is rearranged as follows:

$$\sigma_n = \frac{\sigma_{n, f}}{10^{(\frac{\text{SNR}}{10})}}. \tag{2}$$

Gaussian noise represents a generally appropriate approximation of noise in real-world scenarios and it is thus one of the common variants for noise adding procedures [1]. Here, the mean value is set to zero and the noise level ($\sigma_n$) is used to drive the standard deviation of the Gaussian distribution. Then, a random noise value is generated from the created distribution and is added to the noise-free set ($x_t$) as follows:

$$x'_t = x_t + \mathcal{N}(0, \sigma_n^2), \quad t = 1, 2, 3, \ldots, N \tag{3}$$

where $x'_t$ represents the noisy value and $\mathcal{N}(0, \sigma_n^2)$ is the Gaussian distribution to gather random noise values and $N$ is the number of values in the dataset. Thereby, a noisy set can be generated for each $t$ in the dataset.

In addition to Gaussian noise, in this article, we also use uniform noise as an alternative to provide a more comprehensive analysis of the proposed framework behavior. For details, please see the supplementary materials Section S-III.

B. Noise Estimation

In the proposed framework, four main steps are proposed to complete the process of nonsingleton input FS generation. In the second step, which is based on noise estimation, various noise estimation techniques can be utilized. Numerous noise estimation techniques have been developed over the years [1], [26]–[29], which utilize different models and techniques such as maximum likelihood, deep auto encoders, or principal component analysis. In this article, for the sake of simplicity, one of the early noise estimation approaches, initially proposed for images [30] is used. Here, first, a difference operator is implemented over image patches and the standard deviation of the result in each patch is calculated. Thereafter, a histogram is evaluated in order to generate a noise level estimate ($\hat{\sigma}_n$) of the whole image.

In this article, a sequence of observations is used as a (sliding) window. The difference operator applied across this window, following [30]

$$y = \frac{1}{\sqrt{2}}(x_{i+1} - x_i) \quad \forall i \in W_t \tag{4}$$

where $W_t$ is the defined window as $W_t = \{i \mid i \in [1, p - 1]; \ p \in \mathbb{N}\}$ at the time $t$, $p$ is the number of values in the defined window, and $y$ is the difference vector.
Then, the standard deviation of the difference vector is formulated as follows:

$$\tilde{\sigma}_n = \sqrt{\frac{1}{p - 1} \sum_{i=1}^{p-1} (y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the difference vector and $\tilde{\sigma}_n$ is the estimated uncertainty that can be utilized as the standard deviation of the Gaussian nonsingleton input value ($x_i$). This estimate subsequently provides the basis for the proposed approach to handle varying uncertainty levels affecting a system’s inputs—without a priori knowledge or optimization.

C. Singleton and Nonsingleton FLSs

Generally, FLSs are completed in three steps: fuzzification, inferencing, and defuzzification, where the output set is mapped to crisp value. In this section, we focus on the fuzzification step in the singleton/nonsingleton case, briefly reviewing both the traditional and recently explored composition approaches to model the interaction between input FS and antecedent FS as part of the firing strength generation in each rule.

1) Singleton and Nonsingleton FSs: In the fuzzification step of FLSs, a given crisp input is transformed into an input FS. Assume that $I$ is an FS in the universe of discourse $X$ where an FS $\mu_I(x)$ takes values in the interval $[0, 1]$, formulated as

$$I = \{ x, \mu_I(x) \mid \forall x \in X \} .$$

Singleton sets (see Fig. 1) are characterized by a single value $x$ with a membership of 1 as follows:

$$\mu_I(x) = \begin{cases} 1, & \text{if } x \in I \\ 0, & \text{if } x \notin I \end{cases} .$$

Generally, input data are distorted by noise in real-world applications. Hence, capturing the noise becomes critical, and this can be done by associating input data ($x$) with nonsingleton FSs (see Fig. 1). Conceptually, it is assumed that the incoming input $x$ is the value that is likely to be correct but because of existing uncertainty, neighboring values of $x$ have also potential to be correct. As we go away from the input $x$ value, the possibility of being correct value is decreasing [6]. Nonsingleton FSs can thus capture input uncertainty in an efficient manner, without requiring changes to other (unrelated) parts of the FLSs, such as antecedents or consequents. As an example, a nonsingleton Gaussian input FS associated with a given crisp input $x$ is formulated as follows:

$$\mu_I(x) = \exp \left[ -\frac{1}{2} \left( \frac{x - x_i}{\sigma} \right)^2 \right]$$

where $\sigma$ is the width or standard deviation of the FS, which can be a given ($\sigma_n$) or estimated ($\tilde{\sigma}_n$) to capture the uncertainty level inputs and $x_i$ are the neighboring values of the mean that is located at $x$.

In practice when it is assumed that a system’s inputs are subject to low levels of uncertainty, the width or $\sigma$ values of the associated Gaussian NSFLSs is intuitively small, while large $\sigma$ values are used to model larger levels of uncertainty. It is important to note that while Gaussian NSFLSs provide an easy and efficient means to model uncertainty levels affecting system inputs, they of course represent a very basic approximation of the actual uncertainty affecting a system’s inputs.

2) Determining the Firing Strength in NSFLSs: In the inference step of FLSs, inputs and antecedent MFs are processed for all the system’s rules. Throughout this processing, the firing strength of each rule is generated based on the interaction/composition of antecedents and input FSs.

Let us assume there are two given FSs, $A$ (antecedent) and $I$ (input), on a universe of $X$, i.e.,

$$A = (x, \mu_A(x)|x \in X) \quad I = (x, \mu_I(x)|x \in X) .$$

In the singleton and nonsingleton cases, the standard inference approach [3] is for the firing strength to be equal to the maximum membership degree of the intersection between input ($I$) and antecedent sets ($A$).

While this approach is standard for singleton inference, a broad body of work has explored the use of other approaches to establishing what is effectively the compatibility between FSs through various measures of similarity [31]–[35]. More recently, in the context of input uncertainty handling in NSFLSs, a variety of methods have been explored to faithfully capture the interaction between input and antecedent MFs [15]–[18], [36]. Here, these approaches have been shown to provide superior modeling of uncertainty in comparison to the standard composition approach. More detailed explanation of these approaches can be found in the supplementary materials Section S-I. In this article, we, therefore, contrast both the traditional composition approach, as well as alternatives, including the similarity-based (sim-NS), centroid-based (cen-NS), and subshod-based (sub-NS) approaches, and provide brief background for each as follows.

1) Similarity-based NSFLSs: Turksen et al. [37], [38] proposed the approximate analogical reasoning schema (AARS) for firing strength generation. In these studies, different measures are proposed to be used between the FSs and the similarity ($S_{AARS}$) is calculated. Thereafter, Raha et al. [39], [40] extend the work by proposing a new similarity measure. Chen et al. [41], [42] employ similarity measures and a predefined threshold value ($\tau_0$). If the calculated similarity is greater than $\tau_0$, the given rule is fired. Wagner et al. [16] explored the use of similarity measures to generate the firing strength in NSFLSs in order...
to better capture interaction between input and antecedent FSs. In [17], a recently introduced similarity measure [43] designed to minimize aliasing (where different FSs result in the same degree of similarity) is also explored.

2) Centroid-based NSFLSs: The centroid-based inferencing approach or cen-NS focuses on the area of intersection between input and antecedent MFs. In [15] and [36], the centroid of the intersection between input and antecedent MFs is calculated. Then, the membership degree in the antecedent FS corresponding to the centroid on the intersection is used as the firing strength.

3) Subsethood-based NSFLSs: In [44], a subsethood measure is utilized in conjunction with a threshold, similarly to [41] and [42]. In [18], the authors explored the use of subsethood in NSFLSs (without using a threshold) to drive the rules’ firing strengths.

In recent years, cen-NS and sim-NS have been applied to real-world unmanned aerial vehicle control applications [45]–[47] showing performance increases under varying noise conditions in respect to SFLSs and NSFLSs employing traditional inference.

In the NSFLS literature, several studies focus on the fuzzification step of NSFLSs to tune parameters. Initial work was carried out to change NSFLS parameters dynamically by implementing a training process [4], [5]. Later on, Menhaj et al. [48] proposed to use FLSSs to expedite the convergence speed of the NSFLSs training. A recent study on fault classification in electromechanical switch systems proposed to use set-membership to achieve higher convergence speed in the training of NSFLSs [49] as well. While these studies achieved promising results, they mainly rely on offline training procedures for better performance. However, as noted previously, real-world applications are often subject to changing conditions affecting a system’s inputs and associated uncertainty at run-time. Apart from offline training studies, Pourabdollah et al. [50] suggested using a weighted moving average to update the input $x$ value. After gathering updated input values, input FSs are constructed over this new input. However, similarly to the offline case, it is assumed that the synthetic and stable noise levels are known a priori.

III. ADONiS FLS Framework

A. General Framework Structure

We propose a general framework to configure NSFLSs suitable for real-world applications where noise/uncertainty levels affecting system inputs vary in magnitude over time, which is commonplace in areas such as signal processing, robotics, and forecasting. The framework has the following four steps.

1) Define the size of the frame: The size of the frame over which to collect a sequence of observations is defined. The frame size can be dynamically set or be stable. For example, when using sensors, such as in a robotics context, the size of the frame may be selected in respect to the sampling rate of the sensors or based on a fixed time frame. Another approach, such as in a time-series context, a search algorithm can be used to determine an optimal frame size.

2) Estimate the uncertainty level: After collecting a sequence of observations, estimate the uncertainty level across the gathered values. Note that, in this step, based on design choice, different uncertainty estimation techniques can be used. According to the input structure of the NSFLS, the uncertainty detection can be implemented by considering different features such as actual input variance versus expected variance, over time, for each input individually or together. In this article, in order to detect noise levels, the algorithm in Section II-B is used as an example.

3) Construct Input FSs: Employ the estimated uncertainty level to construct the nonsingleton input FS for the subsequent (next) iteration of the FLS. For example, Gaussian FSs can be used and the detected uncertainty can be used as the $\sigma$ value of the Gaussian input FSs.

4) Advance the defined frame: After constructing the input FS(s) for the current iteration of the FLS, the FLS output is computed and the time frame is advanced.

The four-step process enables the proposed framework to adapt to variation in the levels of uncertainty affecting a system’s inputs. At each time step, inputs are associated with a given non-singleton FS, for which the parameters are determined directly by the levels of uncertainty detected within the preceding time frame. This flowchart of this generic approach can be seen in Fig. 2. Employing a uncertainty detection technique to construct input FSs provides two main advantages to the NSFLS framework: first, it removes the requirements for a priori knowledge of uncertainty levels (and their variation); and, second, it provides the capacity for adapting to changes in the levels of uncertainty affecting a system (e.g., in respect to varying environmental circumstances).

In this article, we proceed to develop one specific instance of the general framework for time-series analysis as shown in the following section.

B. Specific Framework Instance for Time-Series Prediction

In this article, we focus on applying the proposed ADONiS framework to the context of time-series prediction as an initial area of the application enabling efficient evaluation and demonstration. As such, we provide an instance of the following four generic ADONiS framework steps tailored to time series prediction.

1) Define the size of the frame: The size of a frame—effectively a sliding window—is defined, and a sequence...
of observations collected, as illustrated using red dashed lines in Fig. 3. In this article, the window size is referred as $p$, with discrete values within the window being referred to as $[x_{t-p+1}, x_t]$, where $t$ is the last observed value at time $t$.

2) Estimate the uncertainty: After capturing the input data within the sliding window, the noise level over the gathered values is estimated. In this article, we employ the noise estimation approach detailed in Section II-B.

3) Construction of Input FSs: The uncertainty estimate from the previous step is leveraged to inform the nonsingleton FSs that is associated with the last input in the frame for the next iteration of the FLS.

4) Advance the defined frame: The NSFLS input FS is generated for the current window. Thereafter, the defined window is advanced by one step and the same procedure is repeated.

Algorithm 1 presents the pseudo-code of the proposed adaptive input FSs fuzzification in the context of time-series analysis.

Fig. 3 visualizes how each nonsingleton input FS is constructed according to the estimated noise levels by using the proposed sliding window procedure. Fig 3(a) illustrates a sample instance where a low level of noise leads to a narrow width/s.d. in the input FS. In the second illustration [see Fig. 3(b)], the noise level is higher than the previous sample, which results in wider input FSs, in turn providing the capacity to handle the level of noise. An animated illustration of the complete process can be seen in the media materials available online.

The following section focuses on the evaluation and demonstration of the proposed framework through a series of time-series prediction experiments and analyses.

IV. EXPERIMENTS

As part of the experiments, two commonly used chaotic (MG and Lorenz) time series are used to implement time-series forecasting. In order to generate the respective datasets, initially 2000 samples (from $t = -999$ to $t = 1000$) are generated and, in order to avoid fluctuations in the initial part of the time series, only the last 1000 (from $t = 1$ to $t = 1000$) points are preserved for use in the experiments.

Specifically, the MG time series is generated by using the nonlinear time delay differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t)$$  \hspace{1cm} (10)

where $a$, $b$, and $n$ are constant real numbers, $t$ is the current, and $\tau$ is the delay time. For $\tau > 17$, (10) is known to exhibit chaotic behavior. In this article, $\tau = 30$, $a = 0.2$, and $b = 0.1$.

The Lorenz time series [51] is derived from a model of the earth’s atmospheric convection flow heated from below and cooled from above. It is described using nonlinear differential equations as follows:

$$\dot{x} = \sigma(y - x) \quad \dot{y} = x(p - z) - y \quad \dot{z} = xy - bz$$  \hspace{1cm} (11)
where the dots denote the next values for the three variables $x, y, z$ in the time series. $\sigma$, $b$, and $p$ are, respectively, set to 10, $\frac{8}{9}$, and 28, as suggested in [51].

In order to make a comprehensive and fair comparison between the proposed adaptive framework and nonadaptive NSFLSs, we consider four different key aspects, from training to testing, fuzzification, and prediction, which are captured as part of two experiments (each split into two substages) in Fig. 6. We proceed by providing additional detail for each of the key design aspects.

A. Training

One of the most common training techniques for FLS rule generation—the one-pass Wang–Mendel method is implemented on the first 70% of the time-series samples, using a commonly adopted FLS architecture [52] with seven antecedents based on evenly spaced MFs. The prediction is implemented by using nine past points and the tenth value is predicted. The details of the model training can be found in the supplementary materials Section S-II.

Two key training approaches are explored in this article to generate system rules: Noise-Free Time Series-based training and Noisy Time Series-based training, where the latter is based on the same time-series sample that has been subjected to the injection of a predefined level of noise as detailed in the given experiments.

B. Test Set Generation

In order to test the performance of the proposed adaptive NSFLSs, the remaining 30% of the time-series dataset is used in three different scenarios: using a stable 10 dB level of noise [see Fig. 4(a)]; a mixed scenario where the noise varies from a very low and stable 20 dB to a period of high, but stable 0 dB, before returning to 20 dB [see Fig. 4(b)]; the low and high noise levels from the previous testing series are perturbed randomly in magnitude (by 10 dB) itself, which is illustrated in Fig. 4(c).

In the experiments, two different noise types (Gaussian and uniform noise) are used to analyze the proposed framework. Due to the page limitation, we only show the experiments with Gaussian noise results in the main article, while the results for the uniform noise experiments are presented in the supplementary materials Section S-III.

C. Fuzzification

In this part, both the proposed adaptive input FS generation approach, and, a simpler a priori input FS definition approach are used.

The proposed Adaptive technique follows the ADONiS framework as outlined in Section III. Since the forecasting is done with nine past points, we define the window size ($p$) to be nine and the noise estimation calculation (5) is implemented on the past nine points $[x_{t-9+1}, x_t]$. Then, the estimated noise ($\hat{\sigma}_n$) is used as the standard deviation parameter of the $x_t$ singleton Gaussian input FSs. Thus, fuzzification is implemented in an online learning manner and the input FSs are adapted without requiring any a priori knowledge about the noise levels. In Fig. 5, the adaptive framework is shown as implemented on the Mixed Stable Noise [see Fig. 4(b)] and the generated input FSs (as generated step-by-step across the time-series prediction process) are visualized.

As a point of reference, the inputs FSs are designed manually giving rise to four different comparative systems: a singleton FLS using singleton input FSs, and three distinct NSFLSs, using input FSs configured for 20 dB ($\sigma_{20}$), 10 dB ($\sigma_{10}$), and 0 dB ($\sigma_0$) noise levels, respectively. In the experiments, the comparison of each variant is provided to ensure that the proposed adaptive fuzzification technique is challenged against different nonadapative approaches.

D. NSFLS Prediction

After generating the adequate input FSs to be associated with the most recent crisp input to the (NS)FLSs, the last nine input FSs are processed to predict the next crisp point in the time series (and thus the criss input for the next iteration). An overall view of the generated input FSs over time can be seen in Fig. 5. In the used (NS)FLS models, the min and max operators are used for the $t$-norm and $t$-conorm, respectively. Then, the centre of gravity technique is utilized in the defuzzification step. The details of the NSFLS configuration can be found in the supplementary materials Section S-II. The performance of each FLS is assessed using the symmetric mean absolute percent error (sMAPE) [53], which provides a result between 0% and 200%. Also, as one the most common error measures, mean square error (MSE)-based results are included in the supplementary materials Sections S-IV and S-V. Because of the randomization in the noise adding procedures, each time series is generated $30 \times$ and
at the end of each experiment set the average sMAPE values are provided in the main manuscript, while the MSE results are provided (together with further detail on the NSFLS models) in the supplementary materials Section S-IV due to the space limitation.

As shown in Fig. 6, two main experiments are conducted. In Experiment 1, the proposed adaptive framework is compared to the standard, nonadaptive NSFLS approach with predefined input FS standard deviations (for predefined noise levels). In Experiment 2, four different inference architectures (standard, cen-NS, sim-NS, and sub-NS, see Section II-C) are used in the proposed ADONiS framework and the results are compared to explore the performance of the proposed adaptive framework in respect to these different architectures.

V. RESULTS

Results are discussed separately for both Experiments 1 and 2 summarized in Fig. 6. Also, as mentioned in Section IV, in order to analyze the proposed framework behavior under different noise types, experiments are conducted using both Gaussian and uniform noise. The results for the latter are included in the Supplementary Materials Section S-III.

A. Experiment 1—Adaptive and Nonadaptive Comparison

In Experiment 1, prediction results for the ADONiS framework are compared to different trials of nonadaptive NSFLSs.

1) Experiment 1.1 Noise-Free Training: Experiment 1.1 is conducted separately for both MG and Lorenz time series. The models are trained by using noise-free datasets for each time series and then the three test datasets (see Fig. 4) are used for individual forecasting experiments.

First, the Stable Noise test series [see Fig. 4(a)] is used to compare the adaptive and nonadaptive approaches. The prediction results are reported at the left-hand side of Fig. 7 for MG and Fig. 8 for Lorenz time series.

As can be seen, the adaptive fuzzification prediction results are quite similar to the $\sigma_{20}$ and $\sigma_{10}$ fuzzification value results and provide generally low error (which is intuitive as 10 dB noise was injected into this time series), while the singleton approach and the $\sigma_0$ fuzzification lead to higher error levels.

Second, the Mixed Stable Noise test series [see Fig. 4(b)] is used. The sMAPE result comparisons are provided in the middle of Figs. 7 and 8, which show that the adaptive technique has a clear performance benefit over each case of nonadaptive
systems. We note that the reason for the poor performance of the singleton FLS is most likely due to the high “drop-out” of rules arising from the adopted rule learning method; however, this is beyond the scope of this article.

Third, the more challenging test series [Variable Noise as shown in Fig. 4(c)] is used in testing with results shown in part of Figs. 7 and 8. Here, again, the adaptive technique outperforms each of the nonadaptive systems.

2) Experiment 1.2 Noisy Training: After completing three sets of experiments with the noise-free trained models, a more challenging context is established training the models with noisy datasets. Three training datasets, which follow a similar structure as the corresponding test datasets, are generated. For example, if the model will be tested with the Mixed Stable noise from Fig. 4(b), the training dataset is generated based on the same structure and noise levels. The sMAPE results are shown in Figs. 9 and 10. We note that the adaptive and the two nonadaptive (\(\sigma_{20}\) and \(\sigma_{10}\)) fuzzification techniques produce similar sMAPE results.

For the second scenario, another training dataset is generated to correspond to the Mixed Stable Noise time series structure [see Fig. 4(b)]. This training series is used to generate system rules, and the resulting systems are tested using the Mixed Stable Noise. The results for both adaptive and nonadaptive systems are shown in Figs. 9 and 10. Although a variety of prefixed \(\sigma\) values are evaluated as bases of comparison for the adaptive technique, the proposed adaptive technique has the lowest sMAPE values.

Furthermore, for the third case, a similar dataset to the Variable Noise [see Fig. 4(c)] dataset is used for training. As the results in Figs. 9 and 10 show, the adaptive approach produces the lowest or close to the lowest sMAPE values.

B. Experiment 2—ADONiS Using Advanced NSFLS Inference

In Experiment 2, advanced NSFLS cen-NS, sim-NS, and sub-NS and the standard NSFLS (see Section II-C) architectures are compared when used in conjunction with the ADONiS framework. In each variant of the experiments, first, the standard technique is implemented and then the same experiments are repeated by using cen-NS, sim-NS, and sub-NS composition methods within the NSFLSs. As shown in Fig. 6, individual experiments are conducted for noise-free and noisy training, respectively, for both the MG and Lorenz time series.

1) Experiment 2.1 Noise-Free Training: The models are trained by using noise-free training for both MG and Lorenz time series and subsequently evaluated for all three noise scenarios captured in Fig. 4. The results for all three scenarios and both time series are shown in Figs. 11 and 12.

As can be seen in the figures, when the sub-NS approach is used with the adaptive strategy, it produces slightly better
there are no substantial differences between the adaptive and some nonadaptive systems for the MG and Lorenz experiments. As the test series contains 10 dB noise level, manually adjusting input FSs to the 10 dB $\sigma_{10}$ values can be argued to provide an unfair or at least unreasonable advantage to the nonadaptive technique in the sense that in the real world, of course, the actual level of noise/uncertainty cannot commonly be known in advance. In contrast, the proposed adaptive framework does not require any a priori information about the noise levels. We note that the proposed adaptive framework is shown to be best able to handle unseen noise levels in comparison to the nonadaptive variants.

In Experiment 1.2, we explore the scenario where predefined noise levels are employed during training (see Figs. 9 and 10). The latter provides a more realistic replication of a real-world context in comparison to noise-free training, nevertheless it still does not account for the fact that noise levels commonly vary (rather than staying constant) in the real world. Even though we test the NSFLSs with a series of noise levels that include the actual noise levels for which the nonadaptive NSFLSs were training, the proposed adaptive technique produces either the lowest or close to the lowest sMAPE values in all scenarios.

In addition the Gaussian type noise injection, Experiments 1.1 and 1.2 are repeated under uniform noise and the results are provided in the supplementary materials Section S-III. As can be seen in those results, the proposed framework shows the similar prediction performance as Gaussian noise type and it generally outperforms manually designed NSFLS counterparts.

As a core part of the rationale underpinning this article is the strong focus on appropriate managing uncertainty affecting system inputs, we have explored (see Experiment 2) a series of recently proposed advanced NSFLS architectures designed to effectively exploit the interaction between input and antecedent MFs. As shown in Figs. 11 and 12, the sub-NS architecture can achieve slightly better sMAPE values against to any other advanced NSFLSs under noise-free training. Figs. 13 and 14 show that in the noisy training conditions both sub-NS and sim-NS have a performance benefit.

In order to assess the statistical reliability of these results, a series of paired sample $t$-tests are conducted. Divergence of results are found depending upon type of error measure used (sMAPE versus MSE), and according to noise levels both in testing and training. However, in general sub-NS is found to be the best performing technique. This is particularly true within the noise-free training conditions, in which it performs significantly better than all other measures in 10 out of 12 conditions. In noisy training conditions, sub-NS is found to be the significantly better in only 3 out of 12 conditions, while sim-NS is found to be best in 8. The detailed results can be found in the supplementary materials Section S-V.

Based on these experiments, which show that the standard NSFLS composition approach is always outperformed, there is strong potential for combining the proposed ADONiS framework with the advanced NSFLS inference architectures in real-world application with varying noise levels.
Overall, the experiments show that while in some specific (constant and low-noise) cases, offline (predefined) fuzzification for a noise level of $\sigma_{10}$ or $\sigma_{10}$ is superior, for cases where the uncertainty levels are unstable, the proposed ADONIS framework delivers superior results across the extensive variety of scenarios explored. The proposed ADONIS framework technique adjusts input FSs dynamically, in an online manner, which directly benefits applications subject to unstable and/or unknown noise levels as are common in real-world circumstances.

VII. Conclusion

In this article, for the first time, a complete framework (ADONIS) was proposed to dynamically configure NSFLS input FSs, in an online manner, by applying an uncertainty detection technique to a sequence of recent observations. Acknowledging the fact that in the real world, sources of (varying levels of) uncertainty are pervasive, a variety of different training/testing scenarios were explored to systematically evaluate the proposed framework. The results from the comparison of the proposed adaptive and nonadaptive techniques suggest that the proposed approach of dynamically changing input FSs is a suitable approach for handling input uncertainty in real-world applications, particularly where input noise levels may vary. Furthermore, four recently introduced advanced variants of NSFLSs, designed to capture the interaction of input and antecedent FSs with high fidelity, were evaluated in conjunction with ADONIS. Here, the results show that combining ADONIS with advanced NSFLS inference mechanisms can deliver better uncertainty handling and results in comparison to the traditional NSFLS inference.

In the future, due to the increased modeling capabilities of type-2 fuzzy logic in handling uncertainty, an extended design with the capacity not only to model changing uncertainty levels, but also the rate of change of these levels, will be explored. Furthermore, we will seek to expand the experimental evaluation of the proposed and similar frameworks to different noise types, going beyond the Gaussian and uniform noise models used in this article. Also, evolving fuzzy systems and parameter optimization algorithms such as grid search will be investigated to determine the appropriate window size to be used in different application contexts of ADONIS, including more complex applications, such as multivariable time series with different types of noise or robotic sensor applications such as [47].

REFERENCES

[1] X. Liu, M. Tanaka, and M. Ohtani, “Single-image noise level estimation for blind denoising,” *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5226–5237, Dec. 2013.
[2] L. Zadeh, “Fuzzy sets,” *Inf. Control.*, vol. 8, no. 3, pp. 338–353, 1965.
[3] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[4] G. C. Mouzouris and J. M. Mendel, “Dynamic non-singleton fuzzy logic systems for nonlinear modeling,” *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 199–208, May 1997.
[5] G. C. Mouzouris and J. M. Mendel, “Nonlinear time-series analysis with non-singleton fuzzy logic systems,” in *Proc. IEEE/IFAF Comp. Intell. Financial Eng.*, 1995, pp. 47–56.
[6] G. C. Mouzouris and J. M. Mendel, “Nonsingleton fuzzy logic systems: Theory and application,” *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 1, pp. 56–71, Feb. 1997.
[7] J. M. Mendel, “Uncertainty, fuzzy logic, and signal processing,” *Signal Process.*, vol. 30, no. 1, pp. 915–933, 2000.
[8] Q. Liang and J. M. Mendel, “Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters,” *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 551–563, Oct. 2000.
[9] S. Jafarianzadah and M. Shahbaziab, “System identification of a non-linear continuous stirred tank heater based on type-2 fuzzy system,” in *Proc. IEEE/IFAF Comp. Intell. Control Decis.*, 2011, pp. 1869–1874.
[10] Y. Chen, D. Wang, and S. Tong, “Forecasting studies by designing mam- dani interval type-2 fuzzy logic systems: With the combination of BP algorithms and KM algorithms,” *Neurocomputing*, vol. 174, pp. 1133–1146, 2016.
[11] A. B. Cara, I. Rojas, H. Pomares, C. Wagner, and H. Hagras, “On com- paring non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system,” in *Proc. IEEE Symp. Adv. Type-2 Fuzzy Logic Syst.*, Apr. 2011, pp. 126–133.
[12] A. B. Cara, C. Wagner, H. Hagras, H. Pomares, and I. Rojas, “Multiojective optimization and comparison of non-singleton type-1 and singleton interval type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 459–476, Jun. 2013.
[13] Z. H. Akpolat, “Non-singleton fuzzy logic control of a dc motor,” *J. Appl. Sci.*, vol. 5, no. 5, pp. 887–891, 2005.
[14] T. W. Chua and W. W. Tan, “Non-singleton genetic fuzzy logic system for arrhythmias classification,” *Eng. Appl. Artif. Intell.*, vol. 24, no. 2, pp. 251–259, 2011.
[15] A. Pourabdollah, C. Wagner, and J. Aladi, “Changes under the hood—A new type of non-singleton fuzzy logic system,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Nov. 2015, pp. 1–8.
[16] C. Wagner, A. Pourabdollah, J. McCulloch, R. John, and J. M. Garibaldi, “A similarity-based inference engine for non-singleton fuzzy logic sys- tems,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 316–323.
[17] D. Pekaslan, S. Kabir, J. M. Garibaldi, and C. Wagner, “Determining firing strengths through a novel similarity measure to enhance uncertainty handling in non-singleton fuzzy logic systems,” in *Proc. Int. Joint Conf. Comput. Intell.*, 2017, pp. 83–90.
[18] D. Pekaslan, J. M. Garibaldi, and C. Wagner, “Exploring subsetting to determine firing strength in non-singleton fuzzy logic systems,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2018, pp. 1–8.
[19] D. Pekaslan, J. M. Garibaldi, and C. Wagner, “Noise parameter estimation for non-singleton fuzzy logic systems,” in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Man, Cybern., Oct. 2018, pp. 2960–2965.
[20] G. Andonovski, G. Mušič, S. Blažič, and I. Škrjanc, “Evolving process iden- tification for process monitoring and prediction of non-linear systems,” *Eng. Appl. Artif. Intell.*, vol. 68, pp. 214–221, 2018.
[21] P. Angelov and R. Yager, “Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density,” in *Proc. IEEE Workshop Evolving Adaptive Intell. Syst.*, Apr. 2011, pp. 62–69.
[22] P. Angelov, *Autonomous Learning Systems*. Hoboken, NJ, USA: Wiley, 2013.
[23] I. Škrjanc, S. Blažič, and P. Angelov, “Robust evolving cloud-based PID control adjusted by gradient learning method,” in *Proc. IEEE Conf. Evolving Adaptive Intell. Syst.*, Jun. 2014, pp. 1–8.
[24] W. Wang and J. Vrbaneck, Jr., “An evolving fuzzy predictor for industrial applications,” *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1439–1449, Dec. 2008.
[25] E. Lughofer, “Evolving fuzzy systemsfundamentals, reliability, interpretability, useability, applications,” in *Handbook on Computational Intelligence: Volume 1: Fuzzy Logic, Systems, Artificial Neural Net- works, and Learning Systems*. Switzerland: World Scientific, 2016, pp. 67–135.
[26] R. B. Kerr, “On signal and noise level estimation in a coherent PCM channel,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-2, no. 4, pp. 450–454, 1966.
[27] I. Cohen, “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging,” *IEEE Trans. Speech Audio Pro- cess.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.
[28] T. Sase, J. P. Ramrez, K. Kitajo, K. Aihara, and Y. Hirata, “Estimating the level of dynamical noise in time series by using fractal dimensions,” *Phys. Lett. A.*, vol. 380, no. 11, pp. 1151–1163, 2016.
[29] M. Sun, X. Zhang, H. V. hamme, and T. F. Zheng, “Unseen noise estimation using separable deep auto encoder for speech enhancement,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 1, pp. 93–104, Jan. 2016.
IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 28, NO. 10, OCTOBER 2020

Direnc Pekaslan (S’19) received the B.Sc. degree in computer science from Azerbaijan Technical University, Baku, Azerbaijan, in 2010, and the M.Sc. degree with distinction in advanced computer science from the University of Sheffield, Sheffield, U.K., in 2015. He is currently working toward the Ph.D. degree to leverage input uncertainty handling with adaptive non-singleton models with the School of Computer Science, University of Nottingham, Nottingham, U.K.

He is currently a member of the Lab for Uncertainty in Data and Decision Making (LUCID) and the Intelligent Modelling and Analysis (IMA) research group. His main research interest includes uncertainty handling with fuzzy logic systems, particularly, focuses on providing adaptive behaviors in nonsingleton fuzzy models that can handle unexpected circumstances in real-world applications.

Christian Wagner (SM’14) received the B.Sc. degree in robotics and embedded systems, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2004, 2005, and 2009, respectively.

He is currently a Professor of computer science with the University of Nottingham, Nottingham, U.K. His research interests include modeling and handling of uncertain data arising from heterogeneous data sources in the context of AI-driven decision support.

Dr. Wagner is an Associate Editor for the IEEE Transactions on Fuzzy Systems journal, the Chair of the IEEE CIS Technical Committee on Fuzzy Systems and Task Force on Cyber Security, as well as an elected member-at-large of the IEEE Computational Intelligence Society Administrative Committee.

Jonathan M. Garibaldi (SM’19) received the B.Sc. (Hons.) degree in physics from Bristol University, Bristol, U.K., in 1984, and the M.Sc. degree in intelligent systems and the Ph.D. degree in uncertainty handling in immediate neonatal assessment from the University of Plymouth, Plymouth, U.K., in 1990 and 1997, respectively.

He is currently the Head of the School of Computer Science, University of Nottingham, Nottingham, U.K., and leads the Intelligent Modelling and Analysis (IMA) research group. The IMA research group undertakes research into intelligent modeling, utilizing data analysis and transformation techniques to enable deeper and clearer understanding of complex problems. He has made many theoretical and practical contributions in fuzzy sets and systems, and in a wide range of generic machine learning techniques in real-world applications. He has authored/coauthored more than 300 papers on fuzzy systems and intelligent data analysis. His main research interests include modeling uncertainty and variation in human reasoning, and in modeling and interpreting complex data to enable better decision making, particularly in medical domains.

Prof. Garibaldi has been the Editor-in-Chief for the IEEE TRANSACTIONS ON FUZZY SYSTEMS, since 2017. He has served regularly in the organizing committees and program committees of a range of leading international conferences and workshops, such as IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, European Conference, and Parallel Problem Solving from Nature.