ADONiS - Adaptive Online Non-Singleton Fuzzy Logic Systems

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

***Abstract*—Non-Singleton Fuzzy Logic Systems (NSFLSs) have the potential to capture and handle input noise within the design of input fuzzy sets. In this paper, we propose an online learning method which utilises a sequence of observations to continuously update the input Fuzzy Sets (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 or relying on offline training procedures to define input FS parameters. To the best of our knowledge, the proposed ADaptive, ONline Non-Singleton (ADONiS) Fuzzy Logic System (FLS) framework represents the first end-to-end framework to adaptively configure non-singleton 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 paper, 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*—Non-Singleton Fuzzy Logic System, Adaptive Input, Noise Estimation, Firing Strength, 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 optimal performance [1]. In light of this, methods have become established to handle noise within the system which 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 non-singleton FSs. When input data are corrupted by noise, Non-Singleton FSs are useful to model this noise and, generally, have the potential to deliver better results compared to (otherwise identical) singleton fuzzy logic system (SFLSs). [3]–[18]. In 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:

- A priori knowledge about the uncertainty level, for example based on expert insight, is utilised 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 modelling of levels of uncertainty which vary over time.
- Training procedures are implemented to optimise 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 data set which accurately reflects all potential real-world operating conditions, and, are generally limited to the selection of one type of FS model to capture the input uncertainty. Both of these aspects tend to prevent NSFLSs dependent on offline optimisation 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 different, non-stationary settings. Recently, [19] introduced a noise estimation strategy which 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

Direnc Pekaslan, Christian Wagner and Jonathan M. Garibaldi are with Intelligent Modelling and Analysis (IMA) Group and Lab for Uncertainty in Data and Decision Making (LUCID), School of Computer Science, University of Nottingham, Nottingham, United Kingdom. (e-mail:direnc.pekaslan@nottingham.ac.uk; christian.wagner@nottingham.ac.uk; jon.garibaldi@nottingham.ac.uk)

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 paper 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 paper, we propose a complete ADaptive, ONline Non-Singleton (ADONiS) framework which incorporates online uncertainty detection and associated parameterization of the Non-Singleton 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 which 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 and Lorenz) is used in order to mimic real world circumstances, where uncertainty levels vary over time.

The structure of this paper is as follows. Section II provides background information on noise generation/estimation, fuzzy sets and related literature. In Section III, the methodology of the present work is provided and Section IV explains the experimental environment. This is followed by the results and discussion in Section V. Lastly, 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 paper. Also, Singleton and Non-Singleton 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) which 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:

$$SNR = 10 * log\Big(\frac{\sigma_{nf}^2}{\sigma_n^2}\Big) \ , \tag{1}$$

where σ_{nf} is the standard deviation of the noise-free dataset and σ_n is the noise level.

Often, it is assumed that the values SNR and σ_{nf} are known a priori. Thus, when the σ_n is sought, (1) is re-arranged as follows:

$$\sigma_n = \frac{\sigma_{nf}}{10^{(\frac{SNR}{20})}}. \tag{2}$$

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 (σ_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:

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

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 paper, we also use uniform noise as an alternative to provide a more comprehensive analysis of the proposed framework behaviour. 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 non-singleton input FS generation. In the second step, which is based on noise estimation, various noise estimation techniques can be utilised. Numerous noise estimation techniques have been developed over the years [1,26]–[29] which utilise different models and techniques such as maximum likelihood, deep auto encoders or principal component analysis. In this paper, for the sake of simplicity, one of the early noise estimation approaches, initially proposed for images [30] is used. Here, firstly, 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 ($\widehat{\sigma}_n$) of the whole image.

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

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

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 calculated as follows:

$$\widehat{\sigma}_n = \sqrt{\frac{1}{p-1}\sum_{i=1}^{p-1}(y_i - \bar{y})^2}, \tag{5}$$

where $\bar{y}$ is the mean of the difference vector and $\widehat{\sigma}_n$ is the estimated uncertainty which can be utilised as the standard deviation of the Gaussian non-singleton input value (x_t).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 optimisation.

C. Singleton and Non-Singleton Fuzzy Logic Systems

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/non-singleton 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 Non-singleton Fuzzy Sets: In the fuzzification step of FLSs, a given crisp input is transformed into an input fuzzy set. Assume that I is a fuzzy set in the universe of discourse X where a FS $\mu_I(x)$ takes values in the interval [0,1], formulated as:

$$I = \{x, \mu_I(x) \mid \forall x \in X\} \ . \tag{6}$$

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

$$\mu_I(x) = \begin{cases} 1 & if \ x \in I \\ 0 & if \ x \notin I \ . \end{cases} \tag{7}$$

Generally, input data is distorted by noise in real-world applications. Hence, capturing the noise becomes critical, and this can be done by associating input data (x) with non-singleton FSs (Fig. 1). Conceptually, it is assumed that the incoming input x is the value which is likely to be correct but because of existing uncertainty, neighbouring 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]. Non-singleton 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 a example, a non-singleton Gaussian input FS associated with a given crisp input x is formulated as follows:

$$\mu_I(x) = \exp\left[-\frac{1}{2}\left(\frac{x - x_i}{\sigma}\right)^2\right] \ , \tag{8}$$

where σ is the width or standard deviation of the FS which can be a given (σ_n) or estimated ($\widehat{\sigma}_n$) to capture the uncertainty level inputs and x_i are the neighbouring values of the mean which 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 σ values of the associated Gaussian NSFLSs is intuitively small, while largeσ 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.

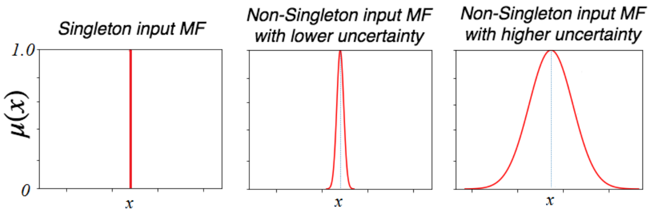


Fig. 1. Singleton and Non-Singleton Gaussian FSs

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

$$\begin{aligned} A &= (x, \mu_A(x) | x \in X) \\ I &= (x, \mu_I(x) | x \in X) \ . \end{aligned} \tag{9}$$

In the singleton and non-singleton 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 NS-FLSs, 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 modelling 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 paper, we therefore contrast both the traditional composition approach, as well as alternatives, including the similarity based (*sim*-NS), centroid based (*cen*-NS) and subsethood based (*sub*-NS) approaches, and provide brief background for each below:

- **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 tpre-defined hreshold value (τ_0). If the calculated similarity is greater than τ_0, the given rule is fired. [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 minimise aliasing (where different FSs result in the same degree of similarity) is also explored.
- **Centroid based NSFLSs:** The centroid-based inferencing approach or *cen*-NS, focuses on the area of intersection between input and antecedent MFs. In [15,36], 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.
- **Subsethood based NSFLSs:** In [44], a subsethood measure is utilised in conjunction with a threshold, similarly

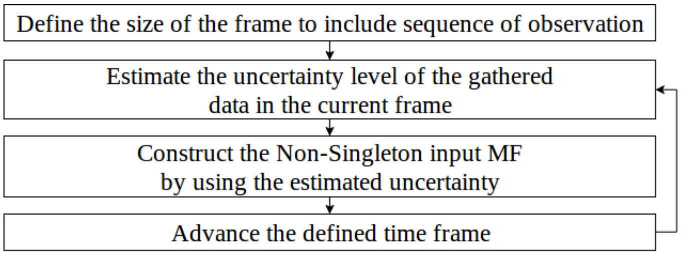

Define the size of the frame to include sequence of observation
Estimate the uncertainty level of the gathered data in the current frame
Construct the Non-Singleton input MF by using the estimated uncertainty
Advance the defined time frame

Fig. 2. The main steps of implemented the sliding window noise estimation framework

to [41,42]. 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 (UAV) 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, [48] proposed to use FLSs 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 above, 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, [50] suggested using a weighted moving average (WMA) 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. ADAPTIVE ONLINE NON-SINGLETON (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, such as is commonplace in areas such as signal processing, robotics and forecasting. The framework has four steps:

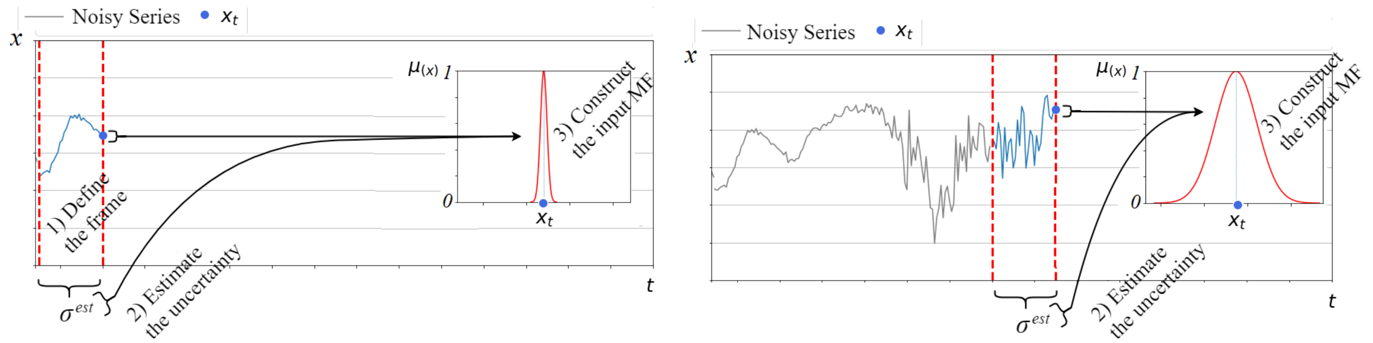
1) *Define the size of the frame:* The size of the frame over which to collect a sequence of observations, is defined. The frame size can be dynamically set or be stable. For example, when using sensors, such as in a robotics context, the size of the frame may be selected in respect to the sampling rate 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:* 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 vs expected variance, over time, for each input individually or together. In this paper, 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 non-singleton 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 σ 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 (See Figs. 3 and 5).The 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: (i) it removes the requirements for a priori knowledge of uncertainty levels (and their variation); and (ii) 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 paper, we proceed to develop one specific instance of the general framework for time series analysis as shown in the next section.

Algorithm 1: Constructing adaptive input FSs by means of sliding window noise estimation and (4) and (5) [30].

```
Input  : the input value $x_t$ at the most recent time step
Output: the corresponding input FS $I_t$ for the input $x_t$
Function noiseEstimation($w_t$):
    $index \leftarrow 1$;
    $y \leftarrow []$        ▷ $y$ is the list for the difference list;
    repeat
        $y[index] \leftarrow \frac{1}{\sqrt{2}}\,(w_t[index+1] - w_t[index])$;
        $index = index + 1$;
    until $index = p - 1$;
    $\widehat{\sigma}_n \leftarrow s.d(y)$        ▷ Standard deviation of the $y$;
    return $\widehat{\sigma}_n$ ;
Function fuzzifyInput($x_t, \widehat{\sigma}_n$)):
    $I_t \leftarrow$ Gaussian FS with mean $x_t$ and s.d. $\widehat{\sigma}_n$;
    return $I_t$ ;
$t \leftarrow 0$;
repeat
    $p \leftarrow frameSize$        ▷ Defining the frame size;
    $w_t \leftarrow [x_{t-p+1}, x_t]$;        ▷ $w_t$ is the current frame;
    $\widehat{\sigma}_n \leftarrow noiseEstimation(w_t)$ ;
    $I_t \leftarrow fuzzifyInput(x_t, \widehat{\sigma}_n)$ ;
    $t \leftarrow t + 1$ ;
until $x_t = end$;
```



(a) Constructing input FS under low level of noise

(b) Constructing input FS under high level of noise

Fig. 3. Illustration of constructing input FSs based noise estimation in a sliding window.

B. A specific framework instance for time series prediction

In this paper, 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 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 paper, 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 paper, 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 non-singleton FSs which 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 visualises how each non-singleton input FS is constructed according to the estimated noise levels by using the proposed sliding window procedure. Fig 3a 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, Fig. 3b, 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 (Mackey-Glass 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.

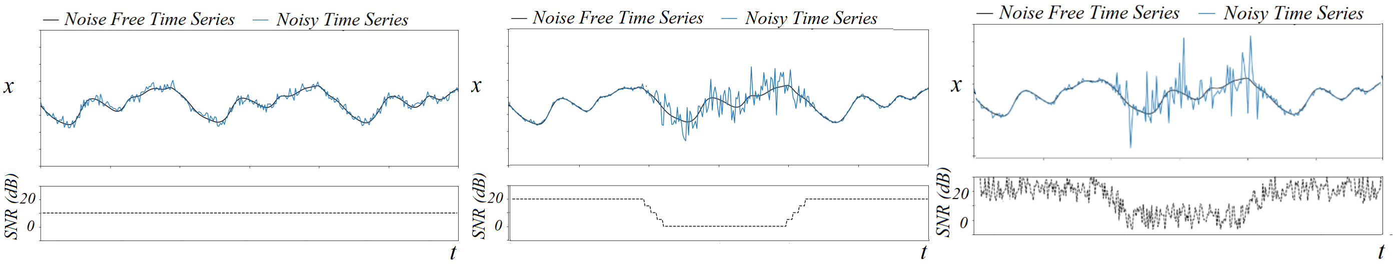
Specifically, the Mackey-Glass (MG) time series is generated by using the nonlinear time delay differential equation:

$$\frac{dx(t)}{dx)} = \frac{ax(t-\tau)}{1 + x^{10}(t-\tau)} - bx(t), \tag{10}$$

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

$$\dot{x} = \sigma(y - x) \qquad \dot{y} = x(p - z) - y \qquad \dot{z} = xy - bz, \tag{11}$$



(a) Stable Noise

(b) Mixed Stable Noise

(c) Variable Noise

Fig. 4. Illustration of the three instances of a test time series generated with different noise level scenarios.

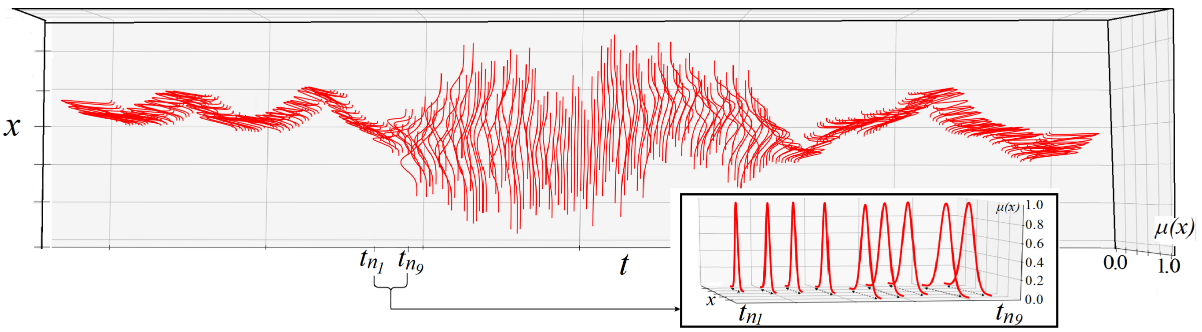


Fig. 5. A sample set of the all generated input FSs in the *Mixed Stable Noise* series (Fig. 4b). Note: the inset figure it tilted for better visibility.

where the dots denote the next values for the three variables x, y, z in the time series. σ, b and p are respectively set to 10, $\frac{8}{3}$ and 28, as suggested in [51].

In order to make a comprehensive and fair comparison between the proposed adaptive framework and non-adaptive 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 sub-stages) in Fig. 6. We proceed by providing additional detail for each of the key design aspects.

1) 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 10th value is predicted. The details of the model training can be found in the supplementary materials S-II.

Two key training approaches are explored in this paper 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 which has however been subjected to the injection of a predefined level of noise as detailed in the given experiments.

2) 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. 4a); 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. 4b); 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. 4c.

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

3) 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 $(\widehat{\sigma_n})$ is used as the standard deviation parameter of the x_t non-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* (Fig. 4b) and the generated input FSs (as generated step-by-step across the time series prediction process) are visualised.

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 (σ_{20}), 10 dB (σ_{10}) and 0 dB (σ_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 non-adaptive approaches.

4) NSFLS Prediction: After generating the adequate input FSs to be associated with the most recent crisp input to the (NS)FLSs, the last 9 input FSs are processed to predict the next crisp point in the time series (and thus the crips 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

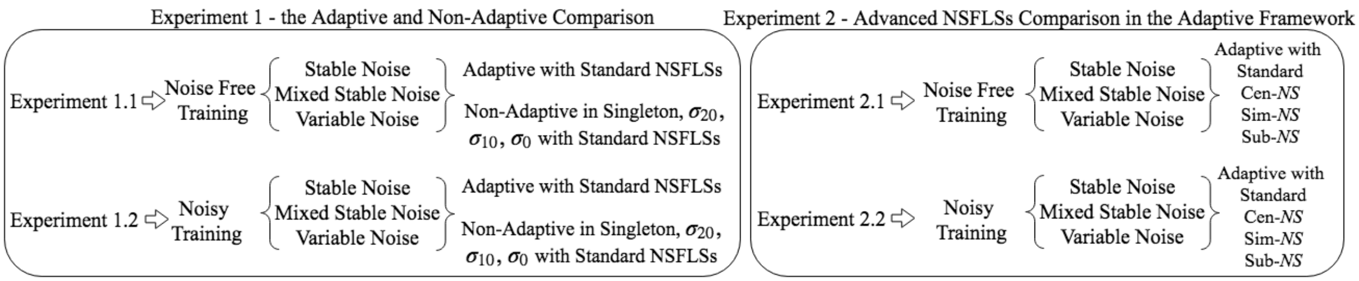


Fig. 6. Experiments 1 and 2, each with its two sub-stages.

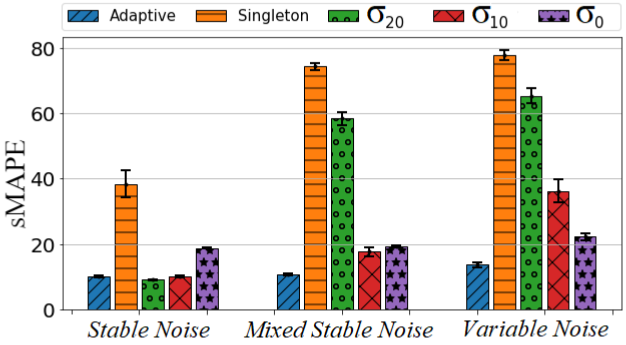


Fig. 7. Experiment 1.1- MG Noise-free trained *standard*NSFLS, with Adaptive and Non-Adaptive fuzzification, sMAPE prediction result comparison. Confidence intervals shown as black lines

t-conorm respectively. Then the centre of gravity technique is utilised in the defuzzification step. The details of the NSFLS configuration can be found in the

supplementary materials S-II. The performance of each FLS is assessed using the 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 randomisation 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, non-adaptive NSFLS approach with pre-defined input FS standard deviations (for pre-defined 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 summarised in Fig. 6. Also, as mentioned in Section IV-2, in order to analyse the proposed framework behaviour 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.

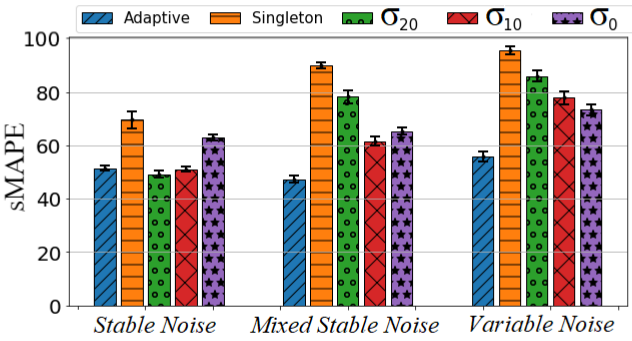


Fig. 8. Experiment 1.1-Lorenz Noise-free trained *standard*NSFLS,with Adaptive and Non-Adaptive fuzzification, sMAPE prediction result comparison.

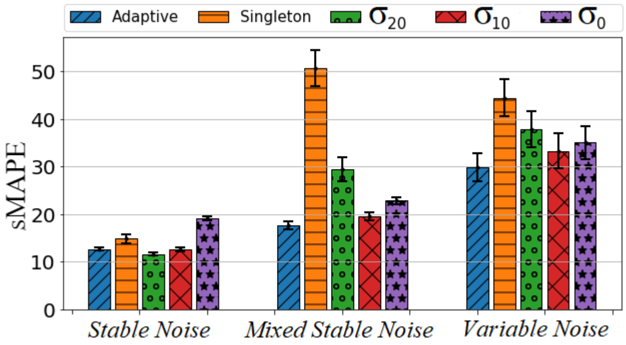


Fig. 9. Experiment 1.2 - MG Noisy trained *standard* NSFLS, with Adaptive and Non-Adaptive fuzzification, sMAPE prediction result comparison.

A. Experiment 1 - Adaptive and Non-Adaptive Comparison

In Experiment 1, prediction results for the ADONiS framework are compared to different trials of non-adaptive 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 (Fig. 4) are used for individual forecasting experiments.

First, the *Stable Noise* test series (Fig. 4a) is used to compare the adaptive and non-adaptive 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 σ_{20} and σ_{10} fuzzification value results and provide generally low error (which is intuitive as 10dB noise was injected into this time series), while the singleton approach and the σ_0 fuzzification lead to higher error levels.

Second, the *Mixed Stable Noise* test series (Fig. 4b) 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 non-adaptive 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 the current paper.

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

2) Experiment 1.2 Noisy Training: After completing three sets of experiments with the noise-free trained models, a more

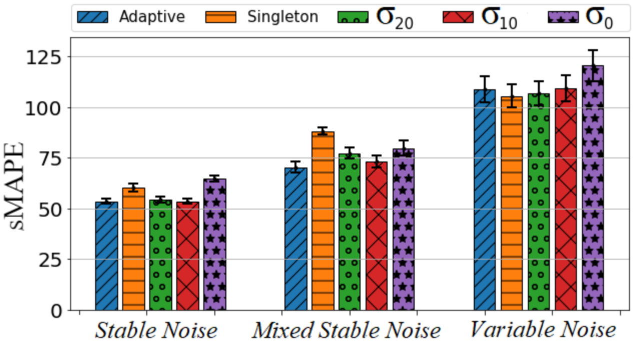


Fig. 10. Experiment 1.2 - Lorenz Noisy trained *standard* NSFLS, with Adaptive and Non-Adaptive fuzzification, sMAPE prediction result comparison.

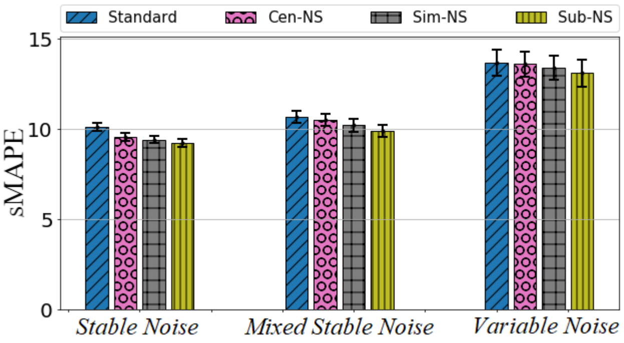


Fig. 11. Experiment 2.1 - MG Noise-Free Trained Advanced *standard*, *cen*-NS, *sim*-NS and *sub*-NS prediction results with adaptive NSFLSs.

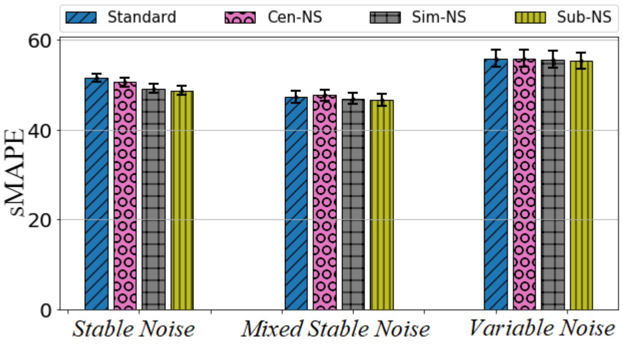


Fig. 12. Experiment 2.1 - Lorenz Noise-free trained advanced *standard*, *cen*-NS, *sim*-NSand *sub*-NSprediction results with adaptive NSFLSs.

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. 4b, the training dataset is generated based on the same structure and noise levels. The sMAPE results are shown in Fig. 9 for MG and Fig. 10 for Lorenz Time series predictions

In these experiments, first, 10 dB noise is injected into the training datasets and the system rule generation is completed. Thereafter, the corresponding *Stable Noise* test dataset which includes 10 dB noise is used in the testing. The sMAPE results are shown in Fig. 9 and Fig. 10. We note that the adaptive and the two non-adaptive (σ_{20} and σ_{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 (Fig. 4b). 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 non-adaptive systems are shown in Fig. 9 and Fig.10. Though a variety of pre-fixed σ values are evaluated as bases of comparison for the adaptive technique, the proposed adaptive technique has the the lowest sMAPE values.

Further, for the third case, a similar dataset to the *Variable Noise* (Fig. 4c) dataset is used for training. As the results in Fig. 9 and Fig. 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*-NSand 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*-NScomposition 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 sMAPE values than the rest of *sim*-NS, *cen*-NS and *standard* NSFLSs counterparts.

2) Experiment 2.2 Noisy Training: In the noisy training experiments (See Figs. 13 and 14), while the *sub*-NS shows better performance under stable noise level in testing, the *sim*-NS outperforms it under unstable Mixed Stable Noise and Variable Noise testing conditions.

VI. Discussion

When the model rules are generated as part of noise-free training (Experiment 1.1, see Figs. 7 and 8), the noise levels in the test series can be considered as unexpected/unseen for the model. When a stable 10 dB noise level is used in the testing dataset, there are no substantial differences between the adaptive and some non-adaptive systems for the MG and Lorenz experiments. As the test series contains 10 dB noise level, manually adjusting input FSs to the 10 dB σ_{10} values can be argued to provide an unfair or at least unreasonable advantage to the non-adaptive 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 non-adaptive variants.

In Experiment 1.2, we explore the scenario where predefined noise levels are employed during training (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 which include the actual noise levels for which the non-adaptive 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, Experiment 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.

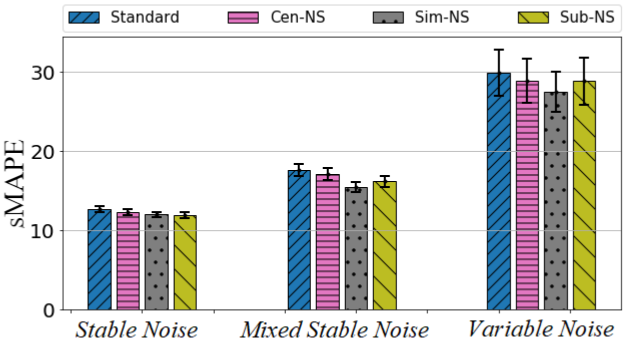


Fig. 13. Experiment 2.2 - MG Noisy Trained Advanced *standard*, *cen*-NS, *sim*-NSand *sub*-NSprediction results with adaptive NSFLSs.

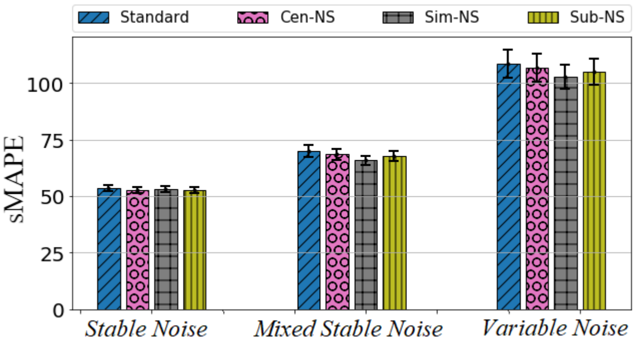


Fig. 14. Experiment 2.2 - Lorenz Noisy trained advanced *standard*, *cen*-NS, *sim*-NS and *sub*-NS prediction results with adaptive NSFLSs.

As a core part of the rationale underpinning this paper is the strong focus on appropriate managing uncertainty affecting system inputs, we have explored (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 vs 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 (pre-defined) fuzzification for a noise level of σ_{20} or σ_{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 and Future Work

In this paper, for the first time, a complete framework (ADONiS) is 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 Non-adaptive 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. Further, four recently introduced advanced variants of NSFLSs, designed to capture the interaction of input and antecedent FSs with high fidelity, are evalauted 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 modelling 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. Further, 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 paper. Also, evolving fuzzy systems and parameter optimisation 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 multi-variate time series with different types of noise or robotic sensor applications such as [47].

Acknowledgment

This research was supported by a School of Computer Science Studentship from the University of Nottingham.

References

[1] X. Liu, M. Tanaka, and M. Okutomi, “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,” *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[3] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR Upper Saddle River, 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] ——, “Nonlinear time-series analysis with non-singleton fuzzy logic systems,” in *Computational Intelligence for Financial Engineering, 1995., Proceedings of the IEEE/IAFE 1995*. IEEE, 1995, pp. 47–56.

[6] ——, “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 Processing*, vol. 80, no. 6, pp. 913 – 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. Jafarinezhad and M. Shahbazian, "System identification of a nonlinear continuous stirred tank heater based on type-2 fuzzy system," in *2011 Chinese Control and Decision Conference (CCDC)*, May 2011, pp. 1869–1874.

[10] Y. Chen, D. Wang, and S. Tong, "Forecasting studies by designing mamdani 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 comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system," in *2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, April 2011, pp. 126–133.

[12] A. B. Cara, C. Wagner, H. Hagras, H. Pomares, and I. Rojas, "Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 459–476, June 2013.

[13] Z. H. Akpolat, "Non-singleton fuzzy logic control of a dc motor," *Journal of Applied Sciences*, 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," *Engineering Applications of Artificial Intelligence*, 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 *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, vol. 2015Novem, 2015, p. 18.

[16] C. Wagner, A. Pourabdollah, J. McCulloch, R. John, and J. M. Garibaldi, "A similarity-based inference engine for non-singleton fuzzy logic systems," in *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on.* IEEE, 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 *International Joint conference on computational Intelligence (IJCCI)*, 2017.

[18] D. Pekaslan, J. M. Garibaldi, and C. Wagner, "Exploring subsethood to determine firing strength in non-singleton fuzzy logic systems," in *Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on.* IEEE, 2018.

[19] ——, "Noise parameter estimation for non-singleton fuzzy logic systems," in *2018 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2018.

[20] G. Andonovski, G. Mušič, S. Blažič, and I. Škrjanc, "Evolving model identification for process monitoring and prediction of non-linear systems," *Engineering applications of artificial intelligence*, 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 *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, April 2011, pp. 62–69.

[22] P. Angelov, *Autonomous Learning Systems.* Wiley Online Library, 2013.

[23] I. Škrjanc, S. Blažič, and P. Angelov, "Robust evolving cloud-based pid control adjusted by gradient learning method," in *2014 IEEE Conference on Evolving and Adaptive Intelligent Systems(EAIS)*, June 2014, pp. 1–8.

[24] W. Wang and J. Vrbanek, 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 Networks, and Learning Systems.* 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.*, no. 4, pp. 450–454, 1966.

[27] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, 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," *Physics Letters 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 Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 93–104, Jan 2016.

[30] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 146, no. 2, pp. 80–84, Aug 1999.

[31] P. Magrez and P. Smets, "Fuzzy modus ponens: A new model suitable for applications in knowledge-based systems," *International Journal of Intelligent Systems*, vol. 4, pp. 181 – 200, 06 1989.

[32] B. Bouchon-Meunier, J. Delechamp, C. Marsala, and M. Rifqi, "Several forms of fuzzy analogical reasoning," in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 1, July 1997, pp. 45–50 vol.1.

[33] V. Cross and T. Sudkamp, "Geometric compatibility modification," *Fuzzy Sets and Systems*, vol. 84, no. 3, pp. 283–299, 1996.

[34] F. Esteva, P. Garcia, L. Godo, and R. Rodríguez, "A modal account of similarity-based reasoning," *International Journal of Approximate Reasoning*, vol. 16, no. 3-4, pp. 235–260, 1997.

[35] L. Godo and R. Rodriguez, "A fuzzy modal logic for similarity reasoning," pp. 33–48, 01 1999.

[36] A. Pourabdollah, C. Wagner, J. H. Aladi, and J. M. Garibaldi, "Improved uncertainty capture for nonsingleton fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1513–1524, Dec 2016.

[37] I. B. Turksen and Z. Zhong, "An approximate analogical reasoning approach based on similarity measures," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 6, pp. 1049–1056, Nov 1988.

[38] I. Turksen and Z. Zhong, "An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets," *Fuzzy Sets and Systems*, vol. 34, no. 3, pp. 323 – 346, 1990.

[39] S. Raha, N. R. Pal, and K. S. Ray, "Similarity-based approximate reasoning: methodology and application," *IEEE Trans. Syst., Man, Cybern. - Part A: Systems and Humans*, vol. 32, no. 4, pp. 541–547, July 2002.

[40] S. Raha, A. Hossain, and S. Ghosh, "Similarity based approximate reasoning: fuzzy control," *Journal of Applied Logic*, vol. 6, no. 1, pp. 47–71, 2008.

[41] S. M. Chen, "A new approach to handling fuzzy decision-making problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 6, pp. 1012–1016, Nov 1988.

[42] S.-M. Chen, "A weighted fuzzy reasoning algorithm for medical diagnosis," *Decision Support Systems*, vol. 11, no. 1, pp. 37 – 43, 1994.

[43] S. Kabir, C. Wagner, T. C. Havens, D. T. Anderson, and U. Aickelin, "Novel similarity measure for interval-valued data based on overlapping ratio," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2017)*, July 2017.

[44] D. S. Yeung and E. C. Tsang, "Improved fuzzy knowledge representation and rule evaluation using fuzzy petri nets and degree of subsethood," *International journal of intelligent systems*, vol. 9, no. 12, pp. 1083–1100, 1994.

[45] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, "A comparative study on the control of quadcopter uavs by using singleton and non-singleton fuzzy logic controllers," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2016, pp. 1023–1030.

[46] ——, "Similarity-based non-singleton fuzzy logic control for improved performance in uavs," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2017, pp. 1–6.

[47] ——, "Input uncertainty sensitivity enhanced nonsingleton fuzzy logic controllers for long-term navigation of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 725–734, April 2018.

[48] M. B. Menhaj, A. A. Suratgar, and M. Karrari, "A modified dynamic non-singleton fuzzy logic system for nonlinear modeling," in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, vol. 6. IEEE, 1999, pp. 4357–4361.

[49] E. P. de Aguiar, F. M. de A. Nogueira, M. M. B. R. Vellasco, and M. V. Ribeiro, "Set-membership type-1 fuzzy logic system applied to fault classification in a switch machine," *EEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2703–2712, Oct 2017.

[50] A. Pourabdollah, R. John, and J. M. Garibaldi, "A new dynamic approach for non-singleton fuzzification in noisy time-series prediction," in *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–6.

[51] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[52] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, 1992.

[53] N. Meade and J. Armstrong, "Long range forecasting: From crystal ball to computer (2nd edition)." *The Journal of the Operational Research Society*, vol. 37, p. 533, 05 1986.