**Interval Type-2 Intuitionistic Fuzzy Logic System for Time Series and Identification Problems - A Comparative Study**

Imo Eyoh¹, Jeremiah Eyoh² and Roy Kalawsky²

¹Department of Computer Science, University of Uyo, Akwa Ibom State, Nigeria
²School of Electrical, Electronics and Systems Engineering, AVRRC Research Group, Loughborough University, UK

**Abstract**

This paper proposes a sliding mode control-based learning of interval type-2 intuitionistic fuzzy logic system for time series and identification problems. Until now, derivative-based algorithms such as gradient descent back propagation, extended Kalman filter, decoupled extended Kalman filter and hybrid method of decoupled extended Kalman filter and gradient descent methods have been utilized for the optimization of the parameters of interval type-2 intuitionistic fuzzy logic systems. The proposed model is based on a Takagi-Sugeno-Kang inference system. The evaluations of the model are conducted using both real world and artificially generated datasets. Analysis of results reveals that the proposed interval type-2 intuitionistic fuzzy logic system trained with sliding mode control learning algorithm (derivative-free) do outperforms some existing models in terms of the test root mean squared error while competing favourable with other models in the literature. Moreover, the proposed model may stand as a good choice for real time applications where running time is paramount compared to the derivative-based models.

**Keywords**

Interval type-2 intuitionistic fuzzy set, Sliding mode control algorithm, Intuitionistic fuzzy set, Intuitionistic fuzzy index.

1. **Introduction**

The classical FS including both the T1FS [1] and T2FS [2]are defined using the membership functions. Literature has it that classical FSs are complementary sets [3]. This implies that to compute the non-membership of a classical FS, one has to take the complement of the set so defined. This may not always be the case in real life contexts because people are often times hesitant to pin-point or specify a single numerical value as doing so indicate strong commitments or evidence.

This brings the idea of intuitionistic FS (IFS) introduced by Atanassov in 1986 [4]. With IFS, a set can be described by three components namely: membership function, non-membership function and hesitation index. With these three representations, IFS becomes a more appropriate tool for dealing with imprecise and vague information. While the non-membership function captures additional information, the hesitation index makes the set description very intuitive and close to human intelligence than classical FS.
However, using single membership and non-membership values to model a concept may not capture enough information and uncertainty in the intuitionistic set definition. This is because changes in the perception of the same linguistic term will lead to frequent re-tuning of the membership and non-membership functions of IFSs. This may result to waste of resources and sub-optimal performance of the system under some environmental and operation conditions.

Atanassov and Gargov [5] extended the concept of the IFS to interval valued intuitionistic fuzzy set (IVIFS), with interval values similar to the notion of classical interval valued fuzzy set (IVFS) [6]. To obtain an IVIFS, the sum of upper membership and upper non-membership must not exceed 1. Based on the salient points discussed above for IFS and IVIFS, Eyoh et al., [7] introduced a rule-based IT2IFLS that merges the capabilities of IT2FLS with those of IFS. The IT2IFLS relaxes the single restriction of the IVIFS by allowing two constraints namely that the sum of upper membership and lower non-membership must not exceed 1, similarly the sum of lower membership and upper non-membership must not exceed 1. This makes IT2IFS more flexible than IVIFS.

**RELATED WORK**

Several methods such as gradient descent (GD), extended Kalman filter (EKF), decoupled EKF (DEKF) and the hybrid (DEKF and GD) have been applied so far for the optimization of IT2IFLS. The optimization may be structure or parameter optimization. In [7] IT2IFLS is proposed and the parameters of the model are updated using GD back-propagation method (a first order derivative-based learning algorithm). The proposed model is applied to non-linear system prediction with good results. The same model is also applied to time series prediction [8]. Recently, Luo et al., [9], proposed an evolving recurrent interval type-2 intuitionistic fuzzy neural network (IT2IFNN) and applied the model for online learning and time series prediction. The parameters of the IT2IFNN in [9] are optimized using EKF (a second-order derivative-based method). In [10], the DEKF is adopted for the optimization of the parameters of the IT2IFLS. Using the DEKF allows the parameters of the model to be grouped into vectors such as antecedent and consequent vectors so that interactions are allowed at the second order. Experimental results reveal that EKF-based learning models perform better than the GD-based in terms of prediction accuracy. In [11], a hybrid model of GD back-propagation and DEKF is employed for the adjustment of the parameters of IT2IFLS and the model applied to system identification problem. Recently also, Yuan and Luo [12] proposed an online evolving interval type-2 intuitionistic fuzzy LSTM-Neural Networks (eIT2IF-LSTMNN). In [12] and [13], the parameters of the models are optimized using GD-back-propagation and applied for regression problems. Other studies on IT2IFS is the one reported in [14], where arithmetic operations are defined for IT2IFS using generalized trapezoidal type-2 intuitionistic fuzzy numbers. The proposed model is used to solve a transportation problem. In this work, only the parameters of the IT2IFLS are optimized. According to [15], membership function (non-membership function) parameters of FLSs are very important in deciding the overall performance of the system.

The weakness of GD and EKF-based models is that the GD and EKF-based methods are both derivative-based approaches and involve the computation of partial derivatives for both the membership and non-membership functions which is tedious and time consuming. For large scale computation, these derivative-based approaches may not be the best choice. Unlike the derivative-based models, using SMC learning algorithm to update the IT2IFLS parameters is straightforward.
and relieves the system of the computational burden of calculating the partial derivatives of these parameters.

This paper proposes the optimization of the IT2IFLS parameters using derivative-free methodology with application to time series and prediction problems. In this study, a sliding mode control learning of the parameters of IT2IFLS is developed and used to solve time series and identification problems for the first time. The antecedent and the consequent parameters update benefits from the proposed learning algorithm.

The rest of the paper is structured as follows: In section 2, some underlying concepts are defined. Section 3 takes a view of interval type-2 intuitionistic fuzzy logic system. In Section 4, sliding mode control learning algorithm is defined with the update rule using sliding mode algorithm derived in Section 5. Experimental evaluation is carried out in Section 6 with the conclusion drawn in Section 7.

2. DEFINITIONS

**DEFINITION 1:** An IFS is totally defined by membership function, non-membership function and hesitation index \( \pi \) of element, \( x \in X \) such that \( 0 \leq \mu_{\lambda^*}(x) + \nu_{\lambda^*}(x) + \pi(x) = 1 \) [4].

When the intuitionistic index is 0, the IFS changed to classical FS. Many approaches have been adopted in the literature for constructing membership and non-membership functions of IFS. Some of the approaches include those reported in [16][17][18]. The IFS Gaussian membership and non-membership function in definitions proposed in [17] are adopted in this paper and are defined as follows:

\[
\mu(x) = (1 - \pi_c(x)) \ast \exp \left(-\frac{x-c}{2\sigma}\right)^2
\]

\[
\nu(x) = (1 - \pi_{var}(x)) \cdot \mu(x)
\]

**DEFINITION 2:** An IT2IFS, \( \tilde{\lambda}^* \) is a variant of IFS but membership and non-membership functions are themselves fuzzy and defined as \( \{\mu_{\lambda^*}(x), \mu_{\lambda^*}(x)\} \) and \( \{\nu_{\lambda^*}(x), \nu_{\lambda^*}(x)\} \) respectively for all \( x \in X \) with constraints: \( 0 \leq \mu_{\lambda^*}(x) + \nu_{\lambda^*}(x) \leq 1 \) and \( 0 \leq \mu_{\lambda^*}(x) + \nu_{\lambda^*}(x) \leq 1 \) [19]

For IT2IFS, two footprints of uncertainties (FOUs) are utilized which are membership function FOU and non-membership function FOU (see Figure 1).

\[
\text{FOU}_\mu(\lambda^*) = \bigcup_{x \in X} [\mu_{\lambda^*}, \bar{\mu}_{\lambda^*}]
\]

\[
\text{FOU}_\nu(\lambda^*) = \bigcup_{x \in X} [\nu_{\lambda^*}, \bar{\nu}_{\lambda^*}]
\]

As shown in Figure 1, for designing the non-membership function FOU, the lower membership function becomes the upper non-membership function while the upper membership function becomes the lower non-membership function [20].
3. INTERVAL TYPE-2 INTUITIONISTIC FUZZY LOGIC SYSTEM

An interval type-2 fuzzy logic system consists of intuitionistic fuzzifier, rulebase, inference engine, and output processing block (type reduction and defuzzifier) as shown in Figure 2. The intuitionistic fuzzifier converts crisp exogenous inputs into IT2IFS, with exactly four components namely lower membership, upper-membership, lower non-membership, and upper non-membership. Here, singleton fuzzification is considered. The hesitation index ensures that the sum of lower membership and upper non-membership functions in the input partition space is less than or equal to 1 and similarly, the sum of upper membership and lower non-membership functions is less than or equal to 1. In this study, the approach for constructing membership and non-membership functions reported in [17] is adopted and modified to reflect the type-2 version as follows:

\[
\mu_{\text{L}}(x) = \exp\left(-\frac{x-c}{\sigma_{\text{L}}}^2\right) * (1-\pi_c(x)) \tag{5}
\]

\[
\mu_{\text{U}}(x) = \exp\left(-\frac{x-c}{\sigma_{\text{U}}}^2\right) * (1-\pi_c(x)) \tag{6}
\]

\[
\nu_{\text{L}}(x) = (1-\pi_{\text{var}}(x)) - \frac{1}{\sigma_{\text{L}}} \tag{7}
\]

\[
\nu_{\text{U}}(x) = (1-\pi_{\text{var}}(x)) - \frac{1}{\sigma_{\text{U}}} \tag{8}
\]

where \(\pi_c\) and \(\pi_{\text{var}}\) are the intuitionistic fuzzy indices for center and variance respectively [17]. The \(\pi_c\) and \(\pi_{\text{var}}\) are small numbers in the range of 0 and 1. The intuitionistic fuzzy indices can be chosen by the user or randomly generated. In this study, \(\pi_c\) and \(\pi_{\text{var}}\) are randomly generated in the interval [0,1].

The intuitionistic fuzzy indices used in this study are expressed for IT2IFLS as follows:

\[
\pi_c(x) = \max(0, (1- (\mu_{\text{L}}(x) + \nu_{\text{L}}(x)))) \tag{9}
\]
\[ \mu_{\varphi}(x) = \max(0, 1 - (\mu_{\lambda^*}(x) + \nu_{\lambda^*}(x))) \]  

(10)

\[ \nabla_{\varphi}(x) = \max(0, 1 - (\mu_{\lambda^*}(x) + \nu_{\lambda^*}(x))) \]  

(11)

such that \(0 \leq \pi_c(x) \leq 1\) and \(0 \leq \pi_{\varphi}(x) \leq 1\)

![Diagram of Interval type-2 intuitionistic fuzzy logic system](image)

Figure 2: Interval type-2 intuitionistic fuzzy logic system [20]

The generic IT2IFLS *IF-THEN* rule is represented as follows:

\[ R_k : \text{IF } x_1 \text{ is } \tilde{A}_{ik}^* \text{ and ... and } x_n \text{ is } \tilde{A}_{nk}^* \text{ THEN } y_k = \sum_{i=1}^{n} w_{ik} x_i + b_k \]  

(12)

The generic rule can be formulated for membership and non-membership functions respectively as follows:

\[ R_k^\mu : \text{IF } x_1 \text{ is } \tilde{A}_{ik}^\mu \text{ and ... and } x_n \text{ is } \tilde{A}_{nk}^\mu \text{ THEN } y_k = \sum_{i=1}^{n} w_{ik}^\mu + b_k^\mu \]  

(13)

\[ R_k^\nu : \text{IF } x_1 \text{ is } \tilde{A}_{ik}^\nu \text{ and ... and } x_n \text{ is } \tilde{A}_{nk}^\nu \text{ THEN } y_k = \sum_{i=1}^{n} w_{ik}^\nu + b_k^\nu \]  

(14)

Where \(\tilde{A}_{ik}^s\)s, are IT2IFS, \(y_k\)are the \(k_{th}\) rule outputs for membership function and non-membership function, \(w\) and \(b\)are parameters of the consequent parts. The inference engine maps IT2IFS inputs to IT2IFS outputs using the combinations of the formulated rules. According to [7], the final output of a TSK-type IT2IFLS is defined as follows:

\[ y = (1 - \beta) \sum_{k=1}^{M} f_k^\mu y_k^\mu + (1 - \beta) \sum_{k=1}^{M} f_k^\nu y_k^\nu \]  

(15)

where:

\[ f_k^\mu = \frac{(f_k^\mu + f_k^\nu)}{\sum_{k=1}^{M} f_k^\mu + \sum_{k=1}^{M} f_k^\nu} \]  

(16)

\[ f_k^\nu = \frac{(f_k^\nu + f_k^\mu)}{\sum_{k=1}^{M} f_k^\nu + \sum_{k=1}^{M} f_k^\nu} \]  

(17)
and $f_k^\mu$ and $f_k^\nu$ are normalized firing signals for membership and non-membership functions respectively.

4. SLIDING MODE CONTROL LEARNING

The error of IT2IFLS is defined as follows:

$$E(t) = y^a(t) - y(t)$$  \hspace{1cm} (18)

The error shows the difference between the measured output and the predicted output of the model. The zero value of the error coordinate is shown as time varying sliding surface [21] as below:

$$S(E(t)) = E(t) = y^a(t) - y(t) = 0$$  \hspace{1cm} (19)

With this, the system which is in a sliding mode is guaranteed to be on a sliding. In this way, the IT2IFLS output will match the actual output as close as possible for all time $t > t_h$, where $t_h$ is the hitting time for $E(t) = 0$.

**DEFINITION 3:** A sliding motion will be on a sliding manifold $S(E(t)) = E(t) = 0$ after a time $t_h$ if the condition $S(t)\dot{S}(t) < 0$ is valid $\forall t$ in some nontrivial semi open subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$ [22][23]

5. PARAMETER UPDATE

In this study, the SMC learning is used to update the parameters of IT2IFLS. The antecedent parameters are center ($c$), lower membership function standard deviation ($\sigma$) and upper membership function standard deviation ($\bar{\sigma}$) while $w$ and $b$ are consequent parameters with $\beta$ as the user defined parameter. The update rules of the IT2IFLS parameters are adapted from [24] for IT2FLS as follows for the antecedent parameters:

For the membership function parameter update, we have the following:

$$\dot{c}_{ik} = \dot{x}_i + (x_i-c_{ik})\alpha_1 \text{sgn}(E)$$  \hspace{1cm} (20)

$$\dot{\sigma}_{ik} = - (\sigma_{ik} + \frac{\sigma_{ik}^3}{(x_i-c_{ik})^2})\alpha_4 \text{sgn}(E)$$  \hspace{1cm} (21)

$$\dot{\bar{\sigma}}_{ik} = - (\bar{\sigma}_{ik} + \frac{\bar{\sigma}_{ik}^3}{(x_i-c_{ik})^2})\alpha_4 \text{sgn}(E)$$  \hspace{1cm} (22)

where $\alpha_1$ is the learning rate and $E$ is the learning error. Since the lower non-membership assumes the upper membership function and vice versa, the standard deviation update for the upper membership function applies to standard deviation update for the lower non-membership function and vice versa.
The consequent parameters are updated as follows:

$$\dot{w}_{ik} = -X_i \frac{(1-\beta)f^\mu_k + \beta f^\nu_k}{M ((1-\beta)f^\mu_k + \beta f^\nu_k)} * \alpha \text{sgn}(E)$$  \hspace{1cm} (23)$$

Where $M = ((1-\beta)f^\mu_k + \beta f^\nu_k)^T$.

$$b_{ik} = -\frac{(1-\beta)f^\mu_k + \beta f^\nu_k}{M ((1-\beta)f^\mu_k + \beta f^\nu_k)} * \alpha \text{sgn}(E)$$  \hspace{1cm} (24)$$

In order to adjust the $\beta$ values, the following vectors are defined: $\tilde{F}^\mu(t) \equiv [\tilde{F}^\mu(t) + \tilde{F}^\mu(t)]$. For the lower membership function, we obtained: $\tilde{F}^\mu(t) = [\tilde{F}^\mu_1(t) \tilde{F}^\mu_2(t) ... \tilde{F}^\mu_M(t)]^T$ and $\tilde{F}^\mu(t) = [\tilde{F}^\mu_1(t) \tilde{F}^\mu_2(t) ... \tilde{F}^\mu_M(t)]^T$ for upper membership function. Similarly, for the non-membership functions: $\tilde{F}^\nu(t) \equiv [\tilde{F}^\nu(t) + \tilde{F}^\nu(t)]$ is specified and defined as $\tilde{F}^\nu(t) = [\tilde{F}^\nu(t) \tilde{F}^\nu(t) ... \tilde{F}^\nu_M(t)]^T$ and $\tilde{F}^\nu(t) = [\tilde{F}^\nu_1(t) \tilde{F}^\nu_2(t) ... \tilde{F}^\nu_M(t)]^T$ for lower and upper non-membership functions respectively. $Y = [y_1 y_2 ... y_M]$ represents rule consequent output vector. Then:

$$\beta = -\frac{1}{Y(\tilde{F}^\mu + \tilde{F}^\mu + \tilde{F}^\nu + \tilde{F}^\nu)} * \alpha \text{sgn}(E)$$  \hspace{1cm} (25)$$

In this study, the membership and non-membership function parameters are tuned using SMC learning algorithm.

6. Performance Evaluation

This study analyses two real world datasets and two artificially generated datasets and compares the performances with existing studies in the literature. For all experiments, two IT2IFS are utilized in order to reduce the complexity of the model. The value of the user-defined parameter $\beta$ is kept at 0.5 for all experiments which are carried out using MATLAB© 2019.

6.1. Australian new electricity market

The Australian new electricity market otherwise known as New South Wales (NSW) electricity market data is a real world dataset. Following from [25], four weeks corresponding to four Australian seasons of the year are adopted. These seasons are weeks in January, May, August and October given as in [25]:

- January 24 - January 30
- May 24 - May 30
- August 24 - August 30
- October 24 - October 30

The NSW electricity dataset was analysed in [25] using wavelet neural network. According to [25], the NSW electricity market in the year 2008 was chosen for the analysis because it was the largest. In [10], the same dataset was analysed using IT2IFLS trained with the decoupled extended Kalman filter algorithm. To aid comparison with other learning algorithms, the computational arrangement is the same as that used in [10] with 336 input data for each season,
231 data points are used for training while 100 data points are used for testing. The input-output vector is \([x(t-4), x(t-3), x(t-2), x(t-1); x(t+1)]\), where \(x(t+1)\) is the desired output. For the four inputs NSW electricity dataset, 16 rules are generated with 192 parameters. The dataset for each season is normalized to fall within the range of zero and one and the performance computed over 30 simulations with the root mean squared error (RMSE) and mean absolute error (MAE) as the performance metrics. The RMSE and MAE are computed as below:

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y^a - y^p)^2}
\]

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y^a - y^p|
\]

where \(N\) is the total number of test data, \(y^a\) and \(y^p\) are the actual and predicted outputs respectively.

<table>
<thead>
<tr>
<th>Period</th>
<th>Summer</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>IFLS-DEKF</td>
<td>IT2FLS-DEKF</td>
<td>IT2IFLS-GD</td>
<td>IT2IFLS-DEKF</td>
<td>IT2IFLS-SMC</td>
</tr>
<tr>
<td>RMSE Trn</td>
<td>0.0229</td>
<td>0.0243</td>
<td>0.0243</td>
<td>0.0225</td>
<td>0.0414</td>
</tr>
<tr>
<td>MAE Tst</td>
<td>0.1112</td>
<td>0.2284</td>
<td>0.1599</td>
<td>0.0979</td>
<td>0.0929</td>
</tr>
</tbody>
</table>

Table 1: Performance of different models during Summer season.

<table>
<thead>
<tr>
<th>Period</th>
<th>Autumn</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>IFLS-DEKF</td>
<td>IT2FLS-DEKF</td>
<td>IT2IFLS-GD</td>
<td>IT2IFLS-DEKF</td>
<td>IT2IFLS-SMC</td>
</tr>
<tr>
<td>RMSE Trn</td>
<td>0.0889</td>
<td>0.0891</td>
<td>0.0896</td>
<td>0.0871</td>
<td>0.1035</td>
</tr>
<tr>
<td>MAE Tst</td>
<td>0.0407</td>
<td>0.0410</td>
<td>0.0789</td>
<td>0.0409</td>
<td>0.0628</td>
</tr>
</tbody>
</table>

Table 2: Performance of different models during Autumn season.

<table>
<thead>
<tr>
<th>Period</th>
<th>Winter</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>IFLS-DEKF</td>
<td>IT2FLS-DEKF</td>
<td>IT2IFLS-GD</td>
<td>IT2IFLS-DEKF</td>
<td>IT2IFLS-SMC</td>
</tr>
<tr>
<td>RMSE Trn</td>
<td>0.0836</td>
<td>0.0846</td>
<td>0.0916</td>
<td>0.0791</td>
<td>0.1085</td>
</tr>
<tr>
<td>MAE Tst</td>
<td>0.0439</td>
<td>0.0429</td>
<td>0.0553</td>
<td>0.0422</td>
<td>0.0592</td>
</tr>
</tbody>
</table>

Table 3: Performance of different models during Winter season.
Table 4: Performance of different models during Spring season.

<table>
<thead>
<tr>
<th>Period</th>
<th>IFLS-DEKF</th>
<th>IT2IFLS-DEKF</th>
<th>IT2IFLS-GD</th>
<th>IT2IFLS-DEKF</th>
<th>IT2IFLS-SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>Trn</td>
<td>0.0715</td>
<td>0.0754</td>
<td>0.0802</td>
<td>0.0723</td>
</tr>
<tr>
<td></td>
<td>Tst</td>
<td>0.0960</td>
<td>0.0954</td>
<td>0.1333</td>
<td>0.0821</td>
</tr>
<tr>
<td>MAE</td>
<td></td>
<td>0.0342</td>
<td>0.0367</td>
<td>0.0477</td>
<td>0.0335</td>
</tr>
</tbody>
</table>

As indicated in tables 1 to 4, IT2IFLS-SMC outperforms other IT2IFLS models in terms of the test RMSE during the Summer and Spring seasons and competes favourable with other models in the other instances. The DEKF trained models tend to exhibit better prediction accuracy in terms of training RMSE in most instances while the type-1 version do perform better in some cases, an indication that a type-1 variant of FLS may work well in some instances depending on the type of data and level of uncertainty in the data. There is a loss in the performance SMC trained IT2IFLS in terms of the training RMSE. However, the computational cost of IT2IFLS-SMC is much more impressive than other competing models in all the cases. For example, the running times for IT2IFLS-SMC, GD and DEKF for one instance of NSW electricity data are 25.78secs, 28.03secs and 44.75secs respectively. Hence IT2IFLS-SMC may stands as an appropriate choice in real time applications where running time is paramount. However, using the SMC learning algorithm with the sign function exhibited some chattering effects in the system during the evaluation of the model. This might also explain the poor performance of IT2IFLS-SMC in the training RMSE.

6.2. Canadian Lynx Dataset

The Canadian lynx time series dataset consist of the number of lynx trapped in the McKenzie river annually in Northern Canada and is taken from period 1821 to 1934. To aid comparison with existing studies [26][27][28][10] the logarithms to base 10 of the Canadian time series is adopted with a periodicity of 10 years. The Canadian time series has a total of 114 sample points where 100 instances are used for training and the rest are used for testing. For a fair comparison with previous studies, the experiment is run for 2000 epochs. The quality of prediction is measured using the mean squared error (MSE) and MAE. The MSE is computed as below:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y^a - y^p)^2$$

(28)

While the MAE is as shown in (26).
Figure 3 (a) shows original Canadian lynx data and (b) transformed Canadian lynx data (log\(10\))

Table 5: Comparison of IT2IFLS with other models on Canadian lynx data

<table>
<thead>
<tr>
<th>Models</th>
<th>MSE(\text{Test})</th>
<th>MAE(\text{Test})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang’s ARIMA (^{26})</td>
<td>0.020486</td>
<td>0.112255</td>
</tr>
<tr>
<td>ANN (^{26})</td>
<td>0.020466</td>
<td>0.112109</td>
</tr>
<tr>
<td>ANN ((p,d,q)) (^{26})</td>
<td>0.013609</td>
<td>0.089625</td>
</tr>
<tr>
<td>Zhang’s Hybrid ARIMA/ANNs model (^{26})</td>
<td>0.017233</td>
<td>0.103972</td>
</tr>
<tr>
<td>Hybrid ARIMA/ERNN model (^{30})</td>
<td>0.009</td>
<td>-</td>
</tr>
<tr>
<td>SETAR (^{31})</td>
<td>0.014</td>
<td>-</td>
</tr>
<tr>
<td>FNN (^{31})</td>
<td>0.009</td>
<td>-</td>
</tr>
<tr>
<td>Generalised Hybrid ARIMA/ANNs model (^{32})</td>
<td>0.00999</td>
<td>0.085055</td>
</tr>
<tr>
<td>ANN/PNN model (^{27})</td>
<td>0.014872</td>
<td>0.079628</td>
</tr>
<tr>
<td>ARIMA/PNN model (^{27})</td>
<td>0.011461</td>
<td>0.084381</td>
</tr>
<tr>
<td>MNM-ANN-DEA (^{33})</td>
<td>0.00663</td>
<td>-</td>
</tr>
<tr>
<td>GA-BPNN (^{28})</td>
<td>0.013590</td>
<td>0.081477</td>
</tr>
<tr>
<td>DB-BPNN (^{28})</td>
<td>0.012899</td>
<td>0.080542</td>
</tr>
<tr>
<td>ANN Ensemble (^{34})</td>
<td>0.00715</td>
<td>-</td>
</tr>
<tr>
<td>RBF-AR (^{35})</td>
<td>0.0073</td>
<td>-</td>
</tr>
<tr>
<td>ADE-BPNN (^{28})</td>
<td>0.010392</td>
<td>0.070723</td>
</tr>
<tr>
<td>GMDH (^{36})</td>
<td>0.0082</td>
<td>0.0634</td>
</tr>
<tr>
<td>LS-SVM (^{36})</td>
<td>0.0074</td>
<td>0.0657</td>
</tr>
<tr>
<td>GLSSVM (^{36})</td>
<td>0.0056</td>
<td>0.0552</td>
</tr>
<tr>
<td>L&amp;NL-ANN (^{37})</td>
<td>0.006</td>
<td>-</td>
</tr>
<tr>
<td>IT2IFLS-GD</td>
<td>0.00463</td>
<td>0.0205</td>
</tr>
<tr>
<td>IT2IFLS-SMC</td>
<td>0.0045</td>
<td>0.0212</td>
</tr>
</tbody>
</table>
Figure 4 shows the actual and predicted values of Canadian lynx data. As shown in Table 5, IT2IFLS-SMC performs better than other existing models in the literature. The SMC-based IT2IFLS prediction accuracy is very close to the GD-based IT2IFLS in this problem domain.

6.3. System identification problem 1

Here, comparison of IT2IFLS-SMC learning with other existing methods in the literature using a system identification problem is conducted. In the first non-linear dynamic system identification, we adopt a second-order time-varying system defined as:

\[ y(t+1) = f(y(t), y(t-1), y(t-2), u(t), u(t-1)) \]

where

\[ f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_4 (x_5 - b) + c x_4}{a + x_2^2 + x_4^2} \]

(29)

and \( a, b, \) and \( c \) are time-varying parameters defined as below:

\[ a(t) = 1.2 - 0.2\cos(2\pi t / T) \]

(30)

\[ b(t) = 1.0 - 0.4\sin(2\pi t / T) \]

(31)

\[ c(t) = 1.0 - 0.4\sin(2\pi t / T) \]

(32)

Here, \( T = 1000 \) represents the total number of sample points. The parameters \( a, b \) and \( c \) take the value 1 as reported in [9]. Similar to [38], in order to keep a manageable number of parameters, only two inputs values are utilised which are \( u(t) \) and \( y(t) \) with \( y(t+1) \) as the desired output. The plot of \( u(t) \) versus \( y(t) \) is as shown in Figure 5.
Figure 5: Plot showing the relationship between the two input $u$ and $y$ for system identification 1.

The following control signal is used to test the performance of the non-linear system.

$$
\begin{cases}
\sin\left(\frac{\pi t}{25}\right) & t < 250 \\
1.0 & 250 \leq t < 500 \\
-1.0 & 500 \leq t < 750 \\
0.3 \sin\left(\frac{\pi t}{25}\right) + 0.1 \sin\left(\frac{\pi t}{32}\right) + 0.6 \sin\left(\frac{\pi t}{10}\right) & 750 \leq t < 1000 
\end{cases}
$$

(33)

The simulation is conducted for 1000 time steps with 100 training epochs. The performance metrics used is the RMSE. For the non-linear system identification, 4 rules are obtained with 40 tunable parameters.

Table 6: Performance comparison of IT2IFLS with other models on system identification 1

<table>
<thead>
<tr>
<th>Models</th>
<th>Number of rules</th>
<th>Training RMSE</th>
<th>Test RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSOFNIN</td>
<td>4</td>
<td>0.0248</td>
<td>0.078</td>
</tr>
<tr>
<td>RFNN</td>
<td>16</td>
<td>0.0114</td>
<td>0.0575</td>
</tr>
<tr>
<td>TRFN-S</td>
<td>3</td>
<td>0.0084</td>
<td>0.0346</td>
</tr>
<tr>
<td>DFNN</td>
<td>3</td>
<td>-</td>
<td>0.05</td>
</tr>
<tr>
<td>R1FNN</td>
<td>4</td>
<td>0.023</td>
<td>0.0465</td>
</tr>
<tr>
<td>HO-RNFS</td>
<td>3</td>
<td>0.0542</td>
<td>0.0815</td>
</tr>
<tr>
<td>RSEFNN-LF(zero)</td>
<td>6</td>
<td>0.0246</td>
<td>0.0383</td>
</tr>
<tr>
<td>RSEFNN-LF(first)</td>
<td>4</td>
<td>0.0199</td>
<td>0.0397</td>
</tr>
<tr>
<td>LSTM-IT2IFNN</td>
<td>4</td>
<td>0.0217</td>
<td>0.0343</td>
</tr>
<tr>
<td>eRIT2IFNN</td>
<td>5</td>
<td>0.0176</td>
<td>0.0254</td>
</tr>
<tr>
<td>eRIT2IFNN-A</td>
<td>5</td>
<td>0.0227</td>
<td>0.0265</td>
</tr>
<tr>
<td>eRIT2IFNN-B</td>
<td>5</td>
<td>0.0211</td>
<td>0.0247</td>
</tr>
<tr>
<td>IT2IFLS-SMC</td>
<td>4</td>
<td>0.0245</td>
<td>0.0267</td>
</tr>
</tbody>
</table>
Figure 6: Actual and predicted outputs for system identification problem 1

Shown in Table 6 is the performance comparison of IT2IFLS-SMC with other learning models in the literature. As shown in Table 6, the performance of IT2IFLS is comparative to those reported in the literature especially the evolving classical IT2FLS methods reported in [9]. Figure 6 shows the actual and predicted outcome for system identification problem 1.

6.4. System identification problem 2

For the second identification problem, a non-linear dynamic plant with longer input delay is adopted and defined as:

\[ y(t+1) = 0.72y(t) + 0.025y(t-1)u(t-1) + 0.01u^2(t-2) + 0.2u(t-3) \]  

Equation (34) involves a one-step ahead prediction using two previous outputs and four previous inputs. The test signal in (34) is adopted to test the quality of prediction. Similar to [12], we adopt the same training data and time steps as in system identification 1. To reduce the complexity of the system, two inputs \( u(t) \) and \( y(t) \) are passed into the IT2IFLS. Shown in Figure 6 is the relationship between the two inputs \( u \) and \( y \) for system identification 2.

Figure 7: Plot showing the relationship between the u and y in system identification problem 2
Table 7: Performance comparison of IT2IFLS on system identification problem 2 with other models in the literature

<table>
<thead>
<tr>
<th>Models</th>
<th>Number of rules</th>
<th>Training RMSE</th>
<th>Test RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSONFIN</td>
<td>6</td>
<td>0.03</td>
<td>0.06</td>
</tr>
<tr>
<td>RFNN</td>
<td>7</td>
<td>0.072</td>
<td>0.128</td>
</tr>
<tr>
<td>TRFN-S</td>
<td>3</td>
<td>0.0067</td>
<td>0.0313</td>
</tr>
<tr>
<td>WRFNN</td>
<td>5</td>
<td>0.0574</td>
<td>0.083</td>
</tr>
<tr>
<td>RIFNN</td>
<td>4</td>
<td>0.0125</td>
<td>0.0288</td>
</tr>
<tr>
<td>RSEFNN-LF(zero)</td>
<td>6</td>
<td>0.0221</td>
<td>0.0383</td>
</tr>
<tr>
<td>RSEFNN-LF(first)</td>
<td>4</td>
<td>0.0156</td>
<td>0.0279</td>
</tr>
<tr>
<td>LSTM-IT2IFNN</td>
<td>4</td>
<td>0.0136</td>
<td>0.0241</td>
</tr>
<tr>
<td>IT2IFLS-SMC</td>
<td>4</td>
<td>0.0153</td>
<td>0.0216</td>
</tr>
</tbody>
</table>

Figure 8: Actual and predicted outputs for system identification problem 2

Figure 7 shows the actual and predicted output for system identification problem 2. As shown in the figure, the predicted output closely follows the actual output indicating a good learning performance. As shown in Table 7, IT2IFLS-SMC performs well in this problem instance in terms of the test RMSE compared to other existing models in the literature.

7. CONCLUSION

This paper analysis time series and system identification problems using IT2IFLS trained with SMC learning algorithm. As demonstrated in the experimental studies, IT2IFLS-SMC outperforms some existing studies in the literature and competes favourably with others in some problem instances in terms of the test RMSE. For the training RMSE, IT2IFLS-SMC performs poorly in many cases. A careful look at all the Australian seasons prediction, IT2IFLS-SMC performs poorly compared to other learning models in terms of the training RMSE.
In the future, it will be worthwhile to look at using other derivative-free learning algorithms and possibly hybridize them and evaluate their predictive strengths. It will also be interesting to consider other membership functions outside the conventional membership functions such as elliptic and diamond-shaped membership functions for interval type-2 intuitionistic fuzzy set.

REFERENCES


AUTHORS

Eyoh, Imo Jeremiah received the B.Sc. and M.Sc. degrees in Computer Science from the University of Uyo and Ibadan, all in Nigeria in 1999 and 2005 respectively. She obtained her PhD in Computer Science from the University of Nottingham, United Kingdom in 2018. She is currently working in the Department of Computer Science, University of Uyo, as a Senior lecturer. Her main research interest is uncertainty modelling using fuzzy logic. In particular, she works in the area of type-2 fuzzy logic (classical and intuitionistic) with application to diverse problem domains. She has published many papers in reputable national and international journals and conferences. She is a member of Computer Professionals Registration Council of Nigeria (CPN) and Organisation for Women in Science for the Developing World (OWSD). She is also a member of IEEE.

Eyoh, Jeremiah Effiong received the B.Sc in Computer Engineering from the University of Uyo in 2006, Nigeria. He received the M.Sc in Control Engineering from the University of Sheffield, United Kingdom in 2008. He is currently working towards the Ph.D. degree in Electrical, Electronics and Systems Engineering at the Loughborough University, United Kingdom.

Short Biography