ReFuzzTiD: A Recurrent Neurofuzzy Model for Anomaly Detection in Time Series

George Kandilogiannakis
Dept. of Informatics & Computer Engineering
University of West Attica
Egaleo, Greece
gkandilogiannakis@uniwa.gr

Paris Mastorocostas
Dept. of Informatics & Computer Engineering
University of West Attica
Egaleo, Greece
mast@uniwa.gr

***Abstract*— In this paper a recurrent neurofuzzy model is proposed, for time series anomaly detection. ReFuzzTiD comprises fuzzy rules whose consequent parts are simple three-layer neural networks with internal feedback at the neurons of the hidden layer. ReFuzzTid is trained by the Simulated Annealing Dynamic Resilient Propagation algorithm. The model learns the dynamics of the time series such that it can classify them by detecting the anomaly points. A comparative analysis with a series of time series anomaly detection models is given, highlighting the characteristics of the proposed detector.**

***Keywords*— time series anomaly detection, neurofuzzy model, recurrent neural network, internal feedback**

I. Introduction

Time series data have been gaining increasing attention during the last years since there exist in many sectors of everyday life, like healthcare, recreation, traveling and logistics. Stemming from the fact that we are living the big data era, a lot of companies are investing on data mining in order to improve services and maximize their profits. These time series are characterized by periodicity, seasonality, trend and irregularity, therefore the detection of point anomalies is quite a difficult task.

Several techniques have been proposed in literature, varying from probabilistic and statistical models to computational intelligence techniques. Statistical ARMA, ARIMA and ARMAX models are widely used, exhibiting the advantage of low-cost implementation. An ARIMA model was presented in [1] for detecting anomalies in wireless sensor networks for traffic control. Breunig, Kriegel, Ng and Sander proposed LOF (Local Outlier Factor, [2]) for local density-based anomaly detection, which was followed by an improved version named COF (Connectivity based Outlier Factor, [3]). Isolation forest (iForest, [4]) is an effective model based anomaly detection technique, which aims at separating an anomalous instance from normal instances by randomly partitioning a tree and randomly selecting the features. One Class Support Vector Machine (OCSVM) constituted the basis for many applications and variants [5]. Additionally, companies like Yahoo have made advances in the field, releasing tools such as EGADS [6], where a two-stage system is used: a time series modeling module that provides an expected value at a time point, followed by an anomaly detection module that inputs the expected value and outputs most probable anomalies.

During the last decade, computational intelligence has contributed to the field of time series analysis, providing very promising results. Long Short Term Memory networks, either standalone [7] or in co-operation with recurrent neural networks [8], work as predictive models for anomaly detection. Deep learning-based systems were recently proposed as time series anomaly detectors; in [9] a deep neural network in combination with wavelet and Hilbert transform is introduced, while in [10] a convolutional neural network is employed for detecting time series anomalies in an unsupervised or semi-supervised manner. Very recently, a fuzzy neural network was proposed for detection of anomalies in large-scale cyberattacks [11].

In view of the above, ReFuzzTiD (Recurrent Fuzzy Time-Series Detector) is proposed for time series anomaly detection. The fuzzy rule base employs rules with consequent parts that are three-layer neural networks, where recurrence is introduced by feeding back the outputs of the neurons of the hidden layer locally. ReFuzzTiD aims at learning the dynamics of time series and at being able to detect the anomalous points. The training process is conducted by use of Simulated Annealing Dynamic Resilient Propagation (SA-DRPROP), a method designed specifically for recurrent structures that alleviates the disadvantages of standard gradient-based methods. In order to reduce the dimensionality of the input space, data are preprocessed by employing Principal Component Analysis (PCA). ReFuzzTiD does not have the structural complexity of other computational intelligence-based approaches and requires a reduced number of data points (only 40% of time series data are used as training data).

The rest of the paper is organized as follows: In Section II the architecture of ReFuzzTiD is described and its particular features are discussed. The structure and the most important features of the learning algorithm are illustrated in section III. Section IV hosts the simulation analysis, where two benchmark problems, including multivariate time series with different percentages of anomaly, are given. Fuzzy models resulted from grid partition of the input space as well as from the application of fuzzy C-means algorithm are examined. A comparative analysis is also conducted, including well-established previous approaches. From these tests it becomes evident that ReFuzzTiD

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

is a promising detector, since it exhibits a very efficient detection performance, similar or superior to those of its competing rivals, while its structure is rather moderate. This detection efficiency can be attributed to the ability of internal feedback connections to capture the dynamics of time series. The concluding remarks are given in Section V.

II. The Architecture of ReFuzzTiD

The neurofuzzy model employed in this work was introduced in [12] and consists of fuzzy rules whose consequent parts are small recurrent neural networks. For the sake of simplicity, let us consider an m-input-single-output system, which consists of fuzzy rules in the form:

$$\begin{aligned}&IF\ x_1(n)\ \text{is}\ A_1\ \text{AND ... AND}\ x_m(n)\ \text{is}\ A_m\\&THEN\ q(n)=q(\boldsymbol{x}(n))\end{aligned} \tag{1}$$

where $\boldsymbol{x}(n)=[x_1,x_2,...,x_m]^T$ is the input vector, n represents the datum index and A_j are the fuzzy sets of the particular rule. In a typical Takagi-Sugeno-Kang fuzzy rule [13], the output of the rule is a linear function of the inputs. In general, the rule output can be any continuous and derivable nonlinear function. In the present case, the following structural characteristics of the model are as follows:

- The premise and defuzzification parts are described by

$$\mu_l(n)=f_\mu(\boldsymbol{x}(n);\ \boldsymbol{m}_l,\boldsymbol{\sigma}_l) \tag{2a}$$

$$q(n)=f_y(\mu_1(n),...,\mu_R(n),q_1(n),...,q_R(n)) \tag{2b}$$

with R being the number of rules and $\mu_l(n)$ being the firing strength of the l-th rule. This two parts are static. The premise parameter vectors are $\boldsymbol{m}_l=[m_{l1},...,m_{lm}]^T$ and $\boldsymbol{\sigma}_l=[\sigma_{l1},...,\sigma_{lm}]^T$. The fuzzy sets are selected to be Gaussian membership functions:

$$\mu_{A_i^l}(x_i(n))=\exp\left\{-\frac{1}{2}\cdot\frac{(x_i(n)-m_{il})^2}{\sigma_{il}^2}\right\} \tag{3}$$

- The degree of fulfillment of each rule is the algebraic product of the membership functions of the fuzzy sets that form the rule.
- The defuzzified output of output of ReFuzzTiD is extracted employing the weighted average:

$$y(k)=\frac{\sum_{j=1}^{R}\mu_j(n)\cdot q_j(n)}{\sum_{j=1}^{R}\mu_j(n)} \tag{4}$$

- The consequent parts are internally recurrent, since $q(\boldsymbol{x}(n))$ is implemented by a three-layer recurrent neural network in the form of m–Hd–1. The network has two static layers: a linear input layer and a typical neuron at the output layer. The hidden layer consists of neurons with local output unit feedback, as shown in Fig. 1. There is no feedback connections of the fuzzy rule's output or external feedback from the model's output.

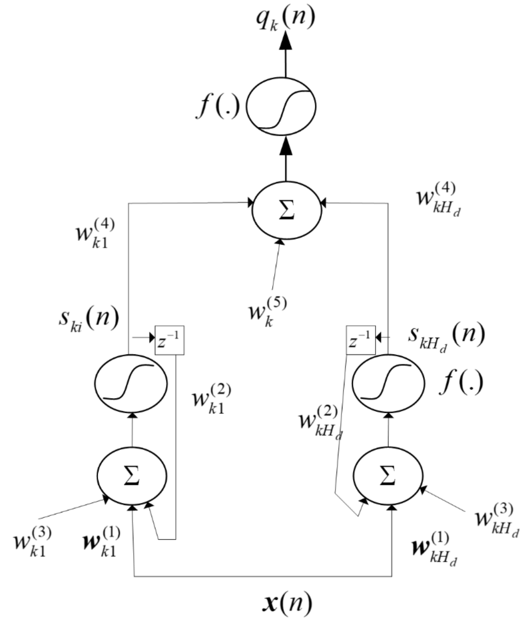


Fig. 1. Configuration of the consequent part of the fuzzy rule.

The consequent part of the k-th fuzzy rule operates using the following equations:

$$s_{ki}(n)=f\left(\sum_{j=1}^{m}\left[w_{kij}^{(1)}\cdot x_j(n)\right]+w_{ki}^{(2)}\cdot s_{ki}(n-1)+w_{ki}^{(3)}\right) \tag{5a}$$

$$q_k(n)=f\left(\sum_{j=1}^{H_d}\left[w_{kj}^{(4)}\cdot s_{kj}(n)\right]+w_k^{(5)}\right) \tag{5b}$$

where the following notation is used:

- f is the activation function of the neurons. It is implemented by the chosen to be the hyperbolic tangent $\frac{e^z-e^{-z}}{e^z+e^{-z}}$.
- $s_{ki}(n)$ is the output of the i-th hidden neuron of the k-th rule at time n.
- $q_k(n)$ is the output of the k-th fuzzy rule.
- $w_{kij}^{(1)}$, $w_{ki}^{(2)}$ and $w_{ki}^{(3)}$ are the synaptic weights and the bias terms, respectively, at the hidden layer of the fuzzy rules.

- $w_{kj}^{(4)}$ and $w_k^{(5)}$ are the synaptic weights and the bias terms, respectively, at the output layer of the fuzzy rules.

In order to classify ReFuzzTiD as a recurrent neurofuzzy model, the recurrent neurofuzzy models that exist in literature [14], [15], could fall into four categories:

(a) Schemes with external feedback schemes, where the inputs are fed with delayed values of the system's outputs [16].

(b) Systems where the rules have dynamic premise parts [17]. There is feedback either at the membership functions, or at the degrees of fulfillment.

(c) Models where the consequent parts of the fuzzy rules have feedback connections [18].

(d) Mixed schemes, where cascaded modules belonging to category *(b)* have feedback connections as well [19].

The present model belongs to the third class, and it was selected due to the fact that it preserves the local learning character of the classic TSK model, since the recurrent neural networks at the consequent parts of the fuzzy rules can be considered as subsystems that are fuzzily interconnected through the fuzzy rules and the defuzzification part. There is no link in time between the rules, since the networks are locally recurrent. At each rule the premise part sets the operating region and the recurrent neural network aims to track the dynamics of the internal states of unknown system within each region. It should be noted that the recurrent networks intrinsically cooperate, since the regions are defined in a fuzzy way, therefore usually more than one rules contribute to input-output mapping at each data point.

The advantages of local output feedback were highlighted in [18] and [20], where this kind of feedback exhibited improved identification characteristics compared to dynamic elements that have local synapse feedback or external feedback. The system in [18], DFNN, shares the same underlying philosophy with ReFuzzTiD, as far as internal feedback at the consequent part of the fuzzy rules is concerned. However, it has highly complex consequent parts, that comprise neural networks with Finite Impulse Response and Infinite Impulse Response synapses at the hidden layer as well as at the output layer. As shown in [12], a simple recurrent structure to the consequent part of the fuzzy rules can perform very efficiently with a significantly lower computational burden.

III. The Training Algorithm

The proposed detector is a recurrent model, therefore the training algorithm should be able to handle the time dependencies that the feedback loops create. In order to do so, the Simulated Annealing Dynamic Resilient Propagation (SA-DRPROP, [21]) is employed, since it improves the learning abilities of the standard gradient-based learning methods, by efficiently handling the problem of trapping to local minima and by searching a broader weight space. Based on RPROP [22] and SARPROP [23], which were designed for static neural networks, SA-DRPROP is briefly discussed in the sequel:

Let us consider a training data set of P input–output data and let $\frac{\partial^+ E(t)}{\partial w_i}$, $\frac{\partial^+ E(t-1)}{\partial w_i}$ be the partial derivatives of an error function E with respect to a model's adaptive weight w_i at the present and the previous iterations, respectively. In SA-DRPROP each weight has its own update step size. During the learning process, the adjustments of the step sizes is based on the sign of the respective derivative at the current and the previous iterations. Therefore, the size of the parameter gradient does not have any effect on the adaptation process, since the temporal behavior of the gradient only matters. SA-DRPROP can be implemented according to the following pseudo-code:

```
(a) For every weight $w_i$ initialize the step sizes $\Delta_i^{(1)} = \Delta_0$
Repeat
    (b) For every weight $w_i$ compute the SA-DRPROP error gradient: $\frac{\partial^+ E(t)}{\partial w_i} - a_1 \cdot SA \cdot \frac{w_i}{1+w_i^2}$
    (c) For each weight $w_i$, update step sizes:
        (c.1) If $\frac{\partial^+ E(t)}{\partial w_i} \times \frac{\partial^+ E(t-1)}{\partial w_i} > 0$
              Then $\Delta_i^{(k)} = \min\left\{ \eta^+ \cdot \Delta_i^{(t-1)}, \Delta_{\max} \right\}$
        (c.2) Else if $\frac{\partial^+ E(t)}{\partial w_i} \times \frac{\partial^+ E(t-1)}{\partial w_i} < 0$
              Then
                  If $(\Delta_i^{(t)} < a_2 \cdot SA^2)$
                  Then
                      $\Delta_i^{(t)} = \max\left\{ \eta^- \cdot \Delta_i^{(t-1)} \cdot 0.8 \cdot r \cdot SA^2, \Delta_{\min} \right\}$
                  Else $\Delta_i^{(t)} = \max\left\{ \eta^- \cdot \Delta_i^{(t-1)}, \Delta_{\min} \right\}$
        (c.3) Else $\Delta_i^{(t)} = \Delta_i^{(t-1)}$
    (d) Update weights $w_i$: $\Delta w_i(t) = -sign\left( \frac{\partial^+ E(t)}{\partial w_i} \right) \cdot \Delta_i^{(t)}$
Until convergence
```

The step sizes are bounded by $\Delta_{\min}, \Delta_{\max}$ and the increase and attenuation factors $n^+ \in [1.05,\ 1.3]$ and $n^- \in [0.5,\ 0.9]$, respectively.

As far as the simulated annealing parameters are concerned, $SA = 2^{-k \cdot Temp}$ (*Temp* is the temperature), r takes random values within [0, 1], while a_1 is set to a small number (e.g. 0.01) and a_2 is set around 0.5 (e.g. 0.4).

In SA-DRPROP the notion of simulated annealing is implemented at steps *(b)* and *(c)*: noise is added to the weight update values when there are two successive changes to the sign of the error gradient, and the magnitude of the update value is less than a threshold, which is proportional to the SA term. Therefore, the adaptation process is moderately affected, since the modification of the weight update by noise takes place only when the weight is threatened to fall to a local minimum, thus allowing the weight to overcome this minimum.

In (*b*) a weight decay term, which modifies the error surface, is added to the error gradient. In this way, in the beginning of the learning process weights with lower values are favored. As training proceeds, this decay term diminishes in order to allow the increase of bigger weights, leading to the exploration of error surface that were previously unavailable.

The error gradients are extracted as follows:

- The premise parts and the output layer of the consequent parts of the fuzzy rules is static, therefore the gradients of the error function with respect to m_{ij} , σ_{ij} , $w_{kj}^{(4)}$ and $w_k^{(5)}$ are derived by use the classic chain rule, using standard partial derivatives:

$$\frac{\partial E}{\partial m_{ij}} = -\frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(n)-\hat{y}_d(n)\right]\cdot\left[q_i(n)-y(n)\right]\cdot \frac{\mu_i(n)\cdot\left[x_j(n)-m_{ij}\right]}{\sum_{k=1}^{R}\mu_k(n)}\} \tag{6a}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = -\frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(n)-\hat{y}(n)\right]\cdot\left[q_i(n)-y(n)\right]\cdot \frac{\mu_i(n)\cdot\left[x_j(n)-m_{ij}\right]^2}{\sigma_{ij}^3\cdot\sum_{k=1}^{R}\mu_k(n)}\} \tag{6b}$$

$$\frac{\partial E}{\partial w_{kj}^{(4)}} = \frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(n)-\hat{y}(n)\right]\cdot\frac{\mu_k(n)\cdot f'(n,k)\cdot x_{kj}(n)}{\sum_{i=1}^{R}\mu_i(n)}\} \tag{6c}$$

$$\frac{\partial E}{\partial w_{k}^{(5)}} = \frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(n)-\hat{y}(n)\right]\cdot\frac{\mu_k(n)\cdot f'(n,k)}{\sum_{i=1}^{R}\mu_i(n)}\} \tag{6d}$$

with $f'(n,k)$ being the derivative of $q_k(n)$, with respect to its arguments.

- Since there are feedback connections at the hidden layer of the consequent parts of the fuzzy rules, ordered derivatives [24] are necessary for the extraction of the error gradients with respect to the weights of the hidden layer's neurons:

$$\frac{\partial^{+} E}{\partial w_{kij}^{(1)}} = \sum_{n=1}^{P}\lambda_{ki}(n)\cdot f'(n,k,i)\cdot x_j(n) \tag{7a}$$

$$\frac{\partial^{+} E}{\partial w_{ki}^{(2)}} = \sum_{n=1}^{P}\lambda_{ki}(n)\cdot f'(n,k,i)\cdot s_{ki}(n-1) \tag{7b}$$

$$\frac{\partial^{+} E}{\partial w_{ki}^{(3)}} = \sum_{n=1}^{P}\lambda_{ki}(n)\cdot f'(n,k,i) \tag{7c}$$

$$\lambda_{ki}(n) = \lambda_{ki}(n+1)\cdot f'(n+1,k,i)\cdot w_{ki}^{(2)} + \frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(n)-\hat{y}(n)\right]\cdot\frac{\mu_k(n)\cdot f'(n,k)\cdot w_{ki}^{(4)}}{\sum_{i=1}^{R}\mu_i(n)}\} \tag{8}$$

$$\lambda_{ki}(P) = \frac{2}{P}\cdot\sum_{n=1}^{P}\{\left[y(P)-\hat{y}(P)\right]\cdot\frac{\mu_k(P)\cdot f'(P,k)\cdot w_{ki}^{(4)}}{\sum_{i=1}^{R}\mu_i(P)} \tag{9}$$

The aforementioned Lagrange multipliers facilitate the calculation of the error gradients [25]. $f'(n,k,i)$ is the derivative of $x_{li}(k)$ with respect to its arguments. Equation (8) is a backward difference equation: first the boundary conditions in (9) for $n = P$ are calculated, and then they are solved in a backward manner for $n = P-1,...,1$.

IV. SIMULATION RESULTS

A. Experimental setup and ReFuzzTiD's features

The proposed detector is applied to two well-known anomaly detection problems: Mammography and Pima. The multivariate datasets are available at UCI Machine Learning Repository [26] and OpenML [27]. As mentioned in [4] and [14], they contain known anomaly cases as ground truth. Moreover, they are selected due to the fact that Mammography has a rather low anomaly percentage of 2%, while Pima has a high percentage of approximately 35%. Thus, ReFuzzTiD will be tested in time series with different characteristics.

All nominal and binary attributes are removed and the data sets for evaluation contain continuous values. Mammography data set has six features and anomalous are considered to be all the data instances that belong to class. In the case of Pima, which is an eight-feature diabetes data set, collected at the National Institute of diabetes and digestive and kidney diseases, the data instances that belong to class "pos" are classified as anomalous. The label "pos" corresponds to patients suffering from diabetes.

We are using only 40% of each time series as training set and the rest 60% of the data set as test set. *k*-fold cross validation is applied (*k* = 5) and training lasts for 1000 epochs. A preprocessing stage is set, in order to reduce the input space and, thus, the model's complexity. Principal component analysis is carried out, reducing the input vector by 1/3 for the case of Mammography (from 6 to 4). For the Pima case the reduction is even greater, since for the first three principal components the proportion of explained variance is greater than 97%.

Consequently the resulting ReFuzzTiD are four input–one output and three input–one output fuzzy systems for Mammography and Pima, respectively. The output threshold for normal / abnormal data points is set to 0.5 for both problems, since our intention is to have a common framework, regardless of the nature of the problem in hand. In this perspective, both ReFuzzTiDs have hidden layers comprising 6 neurons and are trained by the same parameter set, hosted in Table I.

TABLE I. LEARNING PARAMETERS

Temp	n^{+}	n^{-}	$\Delta_{\min}$	$\Delta_{\max}$	Δ_0	a_1	a_2
1.2	1.05	0.5	1E-4	0.5	1E-2	0.01	0.4

The performance of ReFuzzTiD is evaluated by use of the following metrics:

- The Root Mean Squared Error, which is also the error function employed in the extraction of the error gradient.

$$RMSE = \sqrt{\frac{1}{P}\sum_{n=1}^{P}[y(n)-\hat{y}(n)]^2} \qquad (10)$$

with $y(k)$ and $\hat{y}(k)$ being the output of ReFuzzTiD and the actual datum, respectively, for the n-th sample. P is the number of samples. It should be noted that the value of the RMSE does not always represent good classification results and is used in co-operation with other criteria. This metric evaluates the modelling capabilities of the proposed system, i.e. it measures in a quadratic manner the discrepancy between the actual data series and the one produced by ReFuzzTiD.

- The average F_{score}, defined as

$$F_{score} = 2 \cdot \frac{Precision \; - \; Recall}{Precision \; + \; Recall} \qquad (10)$$

where

$$Precision = \frac{True\ Positive}{True\ Positive \; + \; False\ Positive} \qquad (11a)$$

$$Recall = \frac{True\ Positive}{True\ Positive \; + \; False\ Negative} \qquad (11b)$$

- AUC (area under the ROC curve), which represents the degree of separability of normal from abnormal time series.

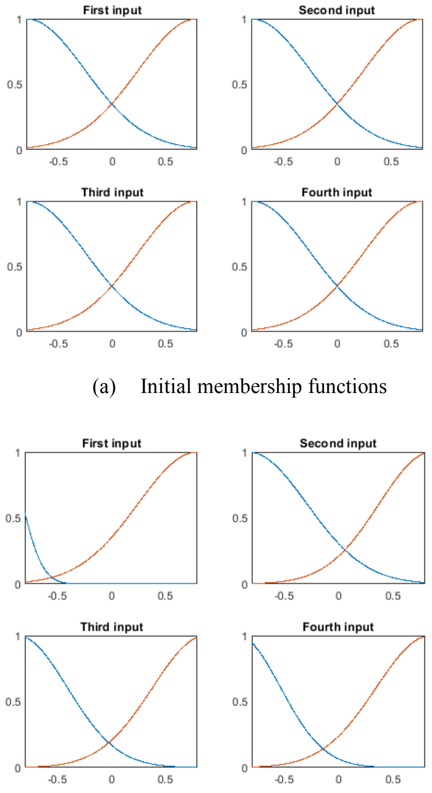
B. Evaluation of ReFuzzTiD

Two ways of input space partition and determination of the number of fuzzy rules are examined: *(a)* grid partition with two fuzzy sets assigned to each input axis. The membership functions of the fuzzy sets are overlapped at membership value equal to 0.35. *(b)* Application of Fuzzy C-means [28], where the standard deviations of the fuzzy sets are derived as stated in [29].

The initial and final membership functions for the case of Mammography are displayed in Fig. 2, where grid partition has been conducted. A severe change of the fuzzy sets for the first input and mild changes for the rest three inputs are noticed, in order to cover the parts of the input space where most of the data are concentrated.

The performance of ReFuzzTiD with respect to the testing data sets is hosted in Table II. For the case of Mammography, grid partition and FCM has provided practically the same detection results, while FCM with a significantly reduced number of rules performed very efficiently as well. Two models resulted from FCM are reported, in order to point out the fact that FCM can provide economical models.

As far as Pima is concerned, both models performed similarly for a small number of fuzzy rules, while the 16-rule model performed superiorly, leading to the conclusion that this kind of time series with high percentage of anomaly requires a detector of increased complexity.



(a) Initial membership functions

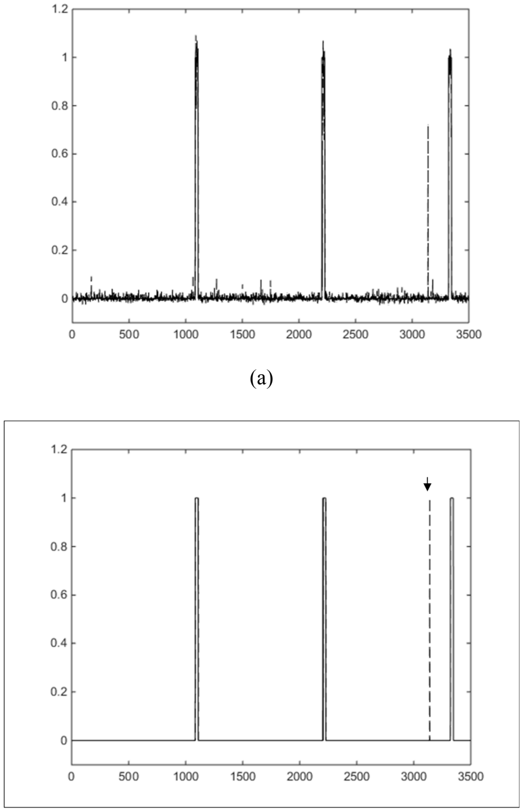
(b) Final membership functions

Fig. 2. Grid partition for the case of Mammography.

TABLE II. PERFORMANCE OF REFUZZTID (TESTING DATA)

Input Partition	*Mammography*			*Pima*		
	RMSE	*Fscore*	*AUC*	*RMSE*	*Fscore*	*AUC*
Grid	0.063 16 rules	0.90 16 rules	0.94 16 rules	0.444 8 rules	0.55 8 rules	0.67 8 rules
FCM	0.072 10 rules	0.83 10 rules	0.92 10 rules	0.456 6 rules	0.57 6 rules	0.68 6 rules
FCM	0.071 16 rules	0.89 16 rules	0.93 16 rules	0.436 16 rules	0.60 16 rules	0.70 16 rules

In order to have a visual representation of the detector's performance, a part of 3500 samples of a time series, along the output of ReFuzzTiD are depicted in Fig. 3a (the actual time series is plotted in solid line and the model's output in dotted line). It is clear that the neurofuzzy model is capable of tracking the time series, thus identifying the anomalous points. The output after the threshold is shown in Fig. 3b. As it can be seen, for this part of the time series the two data series are indistinguishable, with the exception of false positive (marked with an arrow).



(a)

(b)

Fig. 3. ReFuzzTiD output for the case of Mammography.

C. *Comparison with previous works*

Three well-known methods and a recent deep learning-based approach are used as competing rivals. In particular, iForest [4], OCSVM [5], LOF [2] and DeepAnT [10], are applied to the problems of Mammography and Pima. Moreover, DFNN [17] is included to the pool of comparing methods, since it constitutes a system that shares the same underlying philosophy with ReFuzzTiD, but is far more complex.

The model-based methods (iForest, OCSVM, DeepAnT and DFNN) use 40% of the actual data for training and the rest for testing. As far as the other parameters are concerned, the selections made in [10] are adopted. The learning parameters of DFNN are taken from [17], while two structures are reported: *(a)* unit delay at both the IIR and FIR synapses (DFNN$_1$) and *(b)* unit delay for the FIR synapses and a delay of two time steps for the IIR synapses (DFNN$_2$). The results in terms of AUC are hosted in Table III.

From Table III it becomes evident that ReFuzzTiD compares very favorably. For the case of Mammogrpahy, DeepAnT outperforms the competing rivals, followed by DFNN and ReFuzzTid. However, the proposed model attains a relatively high AUC value, while having quite a reduced complexity. As far as Pima is concerned, ReFuzzTiD clearly performs superiorly, with DFNN being second, leading to the conclusion that the high percentage of anomaly and the demanding temporal relations of the time series require models that could efficiently identify these dynamics.

TABLE III. COMPARATIVE PERFORMANCE EVALUATION (AUC)

Method	***Mammography***	***Pima***
iForest	0.85	0.40
OCSVM	0.89	0.26
LOF	0.72	0.48
DeepAnT	**0.99**	0.31
DFNN$_1$	0.93	0.62
DFNN$_2$	0.94	0.61
ReFuzzTiD	0.93	**0.70**

V. CONCLUSIONS

A dynamic neurofuzzy model for time series anomaly detection has been proposed. ReFuzzTiD is a system with fuzzy rules with nonlinear consequent parts, since small recurrent neural networks with local output feedback provide the rule outputs. Training is performed via a hybrid algorithm, which combines a gradient-based method with simulated annealing. The detector has been tested on multivariate time series with different percentages of anomaly, exhibiting a very efficient detection performance, compared to a series of existing detectors. Moreover, ReFuzzTiD has a moderate model structure, since the existence of internal feedback connections are capable of capturing the dynamics of time series. As far as future work is concerned, structure learning and parameter tuning by use of genetic algorithms are two steps towards a fully automated model building process.

ACKNOWLEDGMENT

This work was supported in part by the Research Committee of University of West Attica.

REFERENCES

[1] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," Int. J. Distrib. Sensor Networks, vol. 12, 2016.

[2] M. Breunig, H.-P. Kriegel, R. Ng, J. Sander, "LOF: Identifying density-based local outliers." Proc. ACM SIGMOD Int. Conf. Management of Data, vol. 29, pp. 93-104, 2000.

[3] J. Tang, Z. Chen, A. Fu, and D. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in Advances in Knowledge Discovery and Data Mining, Berlin: Springer, 2002, pp. 535-548.

[4] F. Liu, K. Ting, and Z.-H. Zhou, "Isolation forest," Proc. 8th IEEE Int. Conf. Data Mining, pp. 413-422, 2008.

[5] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," Neural. Comput., vol. 13, pp. 1443-1471, 2001.

[6] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," Proc. 21st ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, pp. 1939-1947, 2015.

[7] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series, Proc. Eur. Symp. Artif. Neural Networks", vol. 23, pp. 89-94, 2015.

[8] Z. Lipton, D. Kale, C. Elkan, and R. Wetzell, "Learning to diagnose with LSTM recurrent neural networks," Proc. Int. Conf. Learning Representations, 2016.

[9] S. Kanarachos, S.-R. Christopoulos, A. Chroneos, and M. Fitzpatrick, "Detecting anomalies in time series data via a deep learning approach combining wavelets, neural networks and Hilbert Transform," Expert Syst. Appl., vol. 85, pp. 292-304, Nov. 2017.

[10] P. Souza, A. Guimaraes, T. Rezende, V. Araujo, V. Araujo, "Detection of anomalies in large-scale cyberattacks using neural networks," MDPI AI, vol. 1, pp. 92-116, 2020.

[11] M. Munir, S. Siddiqui, A. Dengel, and S. Ahmed, "DeepANT: A deep learning approach for unsupervised anomaly detection in time series," IEEE Access, vol. 7, pp. 1991–2005, 2019.

[12] P. Mastorocostas, and C. Hilas, "ReNFFor: a recurrent neurofuzzy forecaster for telecommunications data," Neural Comp. & Appl., vol. 22, pp. 1727-1734, 2013.

[13] T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst., Man, and Cybern., vol. 15, pp. 116–132, Jan.-Feb. 1985.

[14] K. Shihabudheen, and G. Pillai, "Recent advances in neuro-fuzzy system: a survey," Knowledge-Based Syst., vol. 152, pp. 136-162, 2018.

[15] V. Ojha, A. Abraham, and V. Snasel, "Heuristic design of fuzzy inference systems: A review of three decades of research," Eng. Appl. of Art. Int., vol. 85, pp. 845-864, 2019.

[16] S. Jassar, Z. Liao, and L. Zhao, "A recurrent neuro-fuzzy system and its application in inferential sensing," Applied Soft Comp., vol. 11, pp. 2935-2945, 2011.

[17] C.-F. Juang, Y.-Y. Lin, C.-C. Tu, "A recurrent self-evolving fuzzy neural network with local feedbacks and its application to dynamic system processing," Fuzzy Sets & Syst., vol. 161, pp. 2552-2568, 2010.

[18] P. Mastorocostas, and J. Theocharis, "A recurrent fuzzy neural model for dynamic system identification," IEEE Trans. Syst., Man, and Cybern. – Part B, vol. 32, pp. 176-190, Apr. 2002.

[19] D. Stavrakoudis, and J. Theocharis, "Pipelined recurrent fuzzy networks for nonlinear adaptive speech prediction," IEEE Trans. Syst., Man, and Cybern. – Part B, vol. 37, pp. 1305-1320, Oct. 2007.

[20] A. Tsoi, and A. Back, "Locally recurrent globally feedforward networks: a critical review of architectures," IEEE Trans. Naural Networks, vol. 5, pp. 229–239, March 1994.

[21] P. Mastorocostas, and I. Rekanos, "Simulated annealing dynamic RPROP for training recurrent fuzzy systems," Proc.14th IEEE Int. Conf. Fuzzy Systems, pp. 1086-1091, 2005.

[22] M. Riedmiller, and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," Proc. IEEE Int. Joint Conf. Neural Networks, pp. 586-591, 1993.

[23] N. Treadgold, and T. Gedeon, "Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm," IEEE Trans. Neural Networks, vol. 9, pp. 662-668, July 1998.

[24] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University, 1974.

[25] S. Piche, "Steepest descent algorithms for neural network controllers and filters," IEEE Trans. Neural Networks, vol. 5: pp. 198-212, March 1994.

[26] https://archive.ics.uci.edu/ml/datasets/Diabetes.

[27] https://www.openml.org/d/310.

[28] J. Besdek, R. Ehrlich, W. Full, "FCM: the fuzzy c-means clustering algorithm," Comp. & Geosc., vol. 10, pp. 191-203, 1984.

[29] T. Zhou, F.-L. Chung, and S. Wang, "Deep TSK fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data," IEEE Trans. Fuzzy Systems, vol. 25, pp. 1207–1221, Oct. 2017.