

Water leakage forecasting: The application of a modified fuzzy evolving algorithm

Birek, L. , Petrović, D. and Boylan, J.

Author post-print (accepted) deposited in CURVE March 2014

Original citation & hyperlink:

Birek, L. , Petrović, D. and Boylan, J. (2014) Water leakage forecasting: The application of a modified fuzzy evolving algorithm. Applied Soft Computing Journal, volume 14 (PART B): 305-315.

<http://dx.doi.org/10.1016/j.asoc.2013.05.021>

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version of the journal article, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

CURVE is the Institutional Repository for Coventry University

<http://curve.coventry.ac.uk/open>

Water leakage forecasting: the application of a modified fuzzy evolving algorithm

Lech Birek^a, Dobrila Petrovic^{a,*}, John Boylan^b

^a*Coventry University, Priory Street, CV1 5FB, Coventry, United Kingdom*

^b*Buckinghamshire New University, Queen Alexandra Road, HP11 2JZ, High Wycombe, United Kingdom*

Abstract

This paper investigates the use of evolving fuzzy algorithms in forecasting. An Evolving Takagi-Sugeno algorithm (eTS), which is based on a recursive version of the Subtractive algorithm is considered. It groups data into several clusters based on Euclidean distance between the relevant independent variables. The Mod eTS algorithm, which incorporates a modified dynamic update of cluster radii while accommodating new available data is proposed. The created clusters serve as a base for fuzzy If-Then rules with Gaussian membership functions which are defined using the cluster centres and have linear functions in the consequent i.e., Then parts of rules. The parameters of the linear functions are calculated using a weighted version of the Recursive Least Squares algorithm. The proposed algorithm is applied to a leakage forecasting problem faced by one of the leading UK water supplying companies. Using the real world data provided by the company the forecasting results obtained from the proposed modified eTS algorithm, Mod eTS, are compared to the standard eTS algorithm, exTS, eTS+ and fuzzy C-means clustering algorithm and some standard statistical forecasting methods. Different measures of forecasting accuracy are used. The results show higher accuracy achieved by applying the algorithm proposed compared to other fuzzy clustering algorithms and statistical methods. Similar results are ob-

*Computing Department, Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry CV1 5FB, UK, Telephone: 44 24 77659181, Fax: 44 24 76888585, d.petrovic@coventry.ac.uk

Email addresses: birekl@uni.coventry.ac.uk (Lech Birek), d.petrovic@coventry.ac.uk (Dobrila Petrovic), John.Boylan@bucks.ac.uk (John Boylan)

tained when comparing with other fuzzy evolving algorithms with dynamic cluster radii. Furthermore the algorithm generates typically a smaller number of clusters than standard fuzzy forecasting methods which leads to more transparent forecasting models.

Keywords: fuzzy If-Then rules, evolving fuzzy system, forecasting, evolving clustering, leakage

1. Introduction

Development and application of forecasting models to be used in real-world scenarios are often difficult tasks because of non-linear relationships between dependent and independent variables, measurement errors and incomplete datasets. In the last few decades, fuzzy logic has been successfully used to model complex, dynamic problems such as forecasting that would otherwise be difficult to accomplish using conventional mathematical approaches. The advantage of using fuzzy logic lays in its ability to express non-linear relations among variables, typically by combining several linear sub-models, expressed in the form of fuzzy If-Then rules. The rules are often generated based on experts knowledge. This, however, requires the presence of an experienced specialist, and it is usually time consuming and not feasible for large scale applications. One of the ways to overcome this problem is to use data clustering. The data points which share similar properties are grouped into clusters which in turn are a base for the fuzzy If-Then rules. The recent advances in the field of fuzzy clustering [1, 2] have allowed for real-time generation and update of fuzzy If-Then rules. This is particularly helpful in situations where it is important to adapt the rule structure to changing conditions as well as being able to control the way the clusters are generated in the real time.

In this paper, evolving fuzzy systems are applied to the forecasting of water leakage for one of the leading water supply companies in the UK. This approach is particularly useful for this application as it is often difficult to determine the relationship between the dependent and independent variables and it is important to adapt the forecasting model to changing conditions. Having a number of regions of operation, it is also inconvenient to generate different forecasting models for each one of these regions. This paper is novel in three ways: i) a fuzzy (Takagi-Sugeno) evolving algorithm is adapted in order to be suitable for a forecasting application, ii) the generation of the cluster radii is conducted in a different manner as compared to other existing

methods, and iii) fuzzy and statistical forecasting methods are evaluated and compared in the context of water leakage for the first time.

The paper is organized as follows. Section 2 provides a literature review on fuzzy logic based forecasting methods. Section 3 presents the evolving Takagi-Sugeno (eTS) algorithm, introduces the modifications of the standard eTS algorithm and explains how the algorithm is used in forecasting. In Section 4 the case study is presented which introduces the leakage forecasting problem and points out some of the factors influencing the leakage. The application of the algorithm proposed, the results obtained and comparison with some other fuzzy forecasting methods, as well as some standard statistical methods such as Naive, Holt-Winters and linear regression are presented in Section 5. The paper closes with some concluding remarks and indications of future work.

2. Literature review

The application of fuzzy sets theory to forecasting has been of interest to many researchers around the world. One of the first applications of fuzzy sets was in forecasting tax revenue using the language patterns extracted from the available data [3]. Further work has been done in using language values in forecasting. In [4] Song and Chissom proposed the fuzzy time series approach; the method involved defining and partitioning the available data space into a universe of discourse which was fuzzified and used in the forecasting process. The method was applied and investigated in [5, 6] and further modifications were introduced in [7, 8].

Another area of fuzzy logic which found its application in forecasting has been fuzzy clustering, where historical data is grouped into clusters based on a distance measure. The clusters are then used to generate fuzzy If-then rules which are applied in a forecasting process. Considerable research has been done in the application of the well-known fuzzy C-means algorithm. In [9], the fuzzy C-means algorithm is used to generate different lengths of intervals to be used in conjunction with the fuzzy time series approach described above. In [10], the extended fuzzy C-means algorithm was applied and used to help forecast newspaper demand in order to reduce newspaper losses. Use of fuzzy C-means algorithm to automatically build a fuzzy If-then rule structure from the clusters was researched in [11]. Fuzzy If-then rules were used to obtain forecasts based on calculated weights of the activated rules.

The method was applied to temperature forecasting based on the previously recorded temperature and cloud density, as well as to the forecast of the Taiwan stock market index. The drawback of using fuzzy C-means, however, was that the method required setting the number of clusters in advance. This led to time consuming process of finding the number of clusters which would produce the best results. One of the solutions presented in [12] is to use the fuzzy possibilistic approach. Another solution, proposed by Yager and Filev [13], called the Mountain method, helped in estimating the initial coordinates of the cluster centres to be used in other clustering algorithms, such as C-means. Other clustering algorithms emerged directly from the method proposed, such as the Subtractive clustering algorithm [14], used for identification of the Takagi-Sugeno fuzzy models from the available data.

The extension of the Subtractive clustering algorithm called eTS, presented in [15], allowed for the dynamic update of the cluster structure by using a notion of the potential of each data sample. The resulting fuzzy If-then rules, as well as the consequent parameters of the resulting Linear Equations, were also updated in a recursive way. Different versions and modifications of the algorithm and various applications have been researched since then. In [16], a simplified version of the eTS algorithm was introduced, which utilized scatter instead of the potential and included ageing of the rules. In [17], exTS algorithm was presented, which involved the recursive update of the cluster radii according to a data distribution. eTS+ algorithm, presented in [2], further enhanced the eTS method with the on-line monitoring of the quality of generated clusters, on-line structure simplification and on-line input selection. More methods have been developed presenting different approaches, such as neural fuzzy systems (SaFIN [18] and DENFIS [19]), modification to standard algorithms, like C-means [20], Gustafson-Kessel [21] and Gath-Geva [22] clustering, and new ones, such as FLEXFIS algorithm [23]. Complexity reduction was addressed in [24], by analysing the approach of removing local redundancies during the training period. Evolving fuzzy systems were also applied and evaluated on classification problems [25]. The on-line evolving fuzzy classifiers presented there could be used with different model architectures. Some practical applications are described in [26, 27, 28]. A review of different fuzzy evolving approaches which emerged over the past decade is presented in [1].

One of the areas which did not receive enough attention is the application of evolving fuzzy algorithms to forecasting problems. In [27], the eTS al-

gorithm was applied to a time series data set from a Neural Network (NN) competition. The method was successfully applied, but the accuracy of the proposed approach was not compared with any other method. In [29], the evolving fuzzy approach was also applied to a NN competition, with a focus on proposing different top-down approaches to improve the accuracy of a detailed forecast by aggregating daily reading into a weekly time-series. The results for the petrol volume sales estimation using the recursive Gustafson-Kessel clustering were presented in [30]. Those studies show that the evolving fuzzy methods can be used in forecasting applications. However, the obtained results are not compared with any of the widely accepted statistical methods in terms of the forecasting accuracy. Another issue is that the fuzzy evolving algorithms are very often applied to data generated by mathematical models. The lack of comparison and the use of the generated data makes it difficult to assess the usability of the fuzzy forecasting methods in the practical forecasting application.

This study aims at addressing those issues by focusing on the application of a Mod eTS algorithm to a real life problem of one year ahead multivariate water leakage forecasting. The data used consists of multiple independent and dependent variables, with visible seasonal events. The evolving fuzzy algorithm which has been used is based on the eTS with dynamic update of cluster radii. Although the dynamic radii update has already been explored in the previous extensions to eTS, the update process has been changed to accommodate to the features of this application in particular to take into account the importance of the high leakage during certain seasonal periods, which has the highest influence on the average yearly leakage values. The performance of the proposed approach and other fuzzy forecasting and statistical methods are investigated and the results are compared.

3. Mod eTS algorithm

Notation:

k	time instance
i	cluster index
$x_k = [x_{k1} \ x_{k2} \ \dots \ x_{kh}]$	data point (vector)
$c_i = [c_{i1} \ c_{i2} \ \dots \ c_{ih}]$	cluster centre
$P(x_k)$	potential of data point x_k
$P_k(x_k)$	potential of data point x_k at time instance k
α	parameter used in Subtractive clustering potential calculation
β	parameter used in Subtractive clustering potential update
γ	parameter used in the update of radii in Mod eTS
r	radius of a cluster
$r_{k,ij}$	radius of cluster i , variable j at time instance k
Q	high number used to initialise the covariance matrix
σ^2	variance of the Gaussian function
ε	threshold for accepting a data point as a cluster centre
μ_i	activation degree of rule i
λ_i	firing degree of rule i
ψ	regressor vector of the data matrix Ψ
θ	parameter vector of the parameter matrix Θ
Cov_k	covariance matrix at time instance k

3.1. Fuzzy rules generation with Subtractive clustering

Before introducing the modified evolving Takagi-Sugeno algorithm for forecasting, we will first focus on the standard, offline Subtractive clustering algorithm. The Subtractive clustering algorithm [14] is a fuzzy model-based identification method which uses a modification of Mountain clustering method proposed by Yager and Filev [13]. Mountain clustering was initially used to help in estimation of the number of cluster centres for the fuzzy C-means

algorithm. In the fuzzy C-means, coordinates of the cluster centres are obtained as a result of minimizing the cost function which takes into account the distance between cluster centres and data samples and the corresponding membership degrees. This however, requires the number of clusters to be specified in advance.

In the Subtractive clustering algorithm, the cluster centres are automatically generated from the collected training data through calculation of the potential of data samples. Potential defines the influence of the data point on the surrounding data space, i.e., gives an indication on the number of other data points which are in its close proximity. It depends on the cluster spread, i.e. its radius. The obtained cluster structure can then be used to build a fuzzy model based on Takagi-Sugeno inference and a fuzzy weighted Least Squares (LS) estimation of the parameters of the consequent i.e., Then part of rules. The potential P of the training data point x_k is calculated as follows:

$$P(x_k) = \sum_{l=1}^n e^{-\alpha \|x_k - x_l\|^2} \quad (1)$$

$$\alpha = \frac{4}{r^2} \quad (2)$$

where n is a number of data points and r is a cluster radius chosen empirically, which defines the influence of the data samples on the potential and $k = 1, \dots, n$. A high value of the potential of the data sample x_k indicates that there are other data samples being in close proximity to x_k (shorter distances $\|x_k - x_l\|$ between data vectors, calculated as the Euclidean norm in h dimensional space). The data sample x_k with the highest potential $P_{ref} = \max(P(x_k))$ is always chosen to be the first cluster centre $c_1 = x_k$. After every step, the potential of the remaining data samples is updated to account for the new cluster centre as follows:

$$P(x_l) = P(x_l) - P(c_i) e^{-\beta \|x_l - c_i\|^2} \quad (3)$$

$$\beta = \frac{4}{r_b^2} \quad (4)$$

for $l = 1, \dots, n$, where r_b is a positive constant set to be greater than r (a good choice is $r_b = 1.5r$ [14]) and $P(c_i)$ is the potential of the newly obtained cluster centre c_i . The clustering is considered to be finished when the ratio between the highest potential P_{ref} and the potential $P(x_k)$ of currently considered data sample x_k is lower than certain threshold ε , i.e., $\varepsilon > \frac{P(x_k)}{P_{ref}}$.

The value of ε affects the results considerably, as choosing it to be too small results in generating too many clusters, whereas if it is too large, not enough data samples will become cluster centres and consequently not enough fuzzy If-Then rules will be created [14].

When the clustering process is finished, the resulting structure can be used to generate fuzzy If-Then rules. Each cluster corresponds to one fuzzy rule with the cluster centre coordinates being the focal points of Gaussian membership functions of the antecedent, i.e. If part of the fuzzy rule. The consequent, i.e., Then part is in the form of a linear function and Takagi-Sugeno inference is applied. Rule i has the following form:

$$\begin{aligned} &\text{IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ AND } \dots \text{ AND } x_h \text{ is } A_{ih} \\ &\text{THEN } y_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ih}x_h + a_{ih+1} \end{aligned} \quad (5)$$

where $x = [x_1 \ x_2 \ \dots \ x_h]$ is a data sample, $j = 1, \dots, h$ is a number of input values and A_{ij} represents the Gaussian membership function:

$$f(x_j, c_{ij}, \sigma_{ij}^2) = e^{\frac{-(x_j - c_{ij})^2}{2\sigma_{ij}^2}} \quad (6)$$

with c_{ij} being the centre and σ_{ij}^2 a variance of the Gaussian membership function.

The parameters of all linear functions of the consequent parts of fuzzy rules are estimated using the LS algorithm. The activation degree μ_i of each rule is obtained as the membership degree of inputs of the considered data point x_k belonging to the corresponding Gaussian membership functions:

$$\mu_i = e^{-\alpha \|x_k - c_i\|^2} \quad (7)$$

where $\alpha = 1/2\sigma^2$. The firing degree of each rule is then obtained as follows:

$$\lambda_i = \frac{\mu_i}{\sum_{l=1}^R \mu_l} \quad (8)$$

where R is the total number of fuzzy rules (clusters). The firing degrees are combined with input values to create the data matrix Ψ used in the LS algorithm.

$$\Psi = \begin{bmatrix} \lambda_1 x_1^e & \lambda_2 x_1^e & \dots & \lambda_R x_1^e \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1 x_n^e & \lambda_2 x_n^e & \dots & \lambda_R x_n^e \end{bmatrix} \quad (9)$$

where x_k^e is an extended data vector with h input values:

$$x_k^e = [x_{k1} \ x_{k2} \ \cdots \ x_{kh} \ 1] \quad (10)$$

The $\Theta = [\theta_1^T \ \theta_2^T \ \cdots \ \theta_c^T]$ consists of vectors of estimated parameters of the consequent part of each rule where $\theta_1 = [a_{11} \ a_{12} \ \cdots \ a_{1h} \ a_{1h+1}]$, $\theta_2 = [a_{21} \ a_{22} \ \cdots \ a_{2h} \ a_{2h+1}]$, \dots , $\theta_R = [a_{R1} \ a_{R2} \ \cdots \ a_{Rh} \ a_{Rh+1}]$. The parameters are chosen so that the sum of squared errors between real output values and the outputs estimated by using fuzzy rules is the lowest:

$$J = (Y - \Psi^T \Theta)^{-1} (Y - \Psi^T \Theta) \quad (11)$$

The parameters can be estimated by pseudo inverse:

$$\Theta = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (12)$$

Knowing the values of the dependent variables (inputs) x^e , the generated model can be used to predict the output based on the assessment of the firing degrees λ_i using the estimated parameters Θ . The output is estimated as:

$$y = \sum_{i=1}^R \lambda_i \theta_i x^e \quad (13)$$

The flow-chart of the Subtractive algorithm is given in Fig. 1. The method can be applied to multivariate forecasting problems, such as the one of leakage forecasting which will be described later in the paper.

In case of forecasting for several periods ahead, it may be useful to update the model dynamically as new data becomes available. This gives the advantage of gradually evolving the model as opposed to rebuilding it each time, which may consume much more time. The modified evolving version of the Subtractive clustering algorithm is described in the next section.

3.2. Modified Evolving Takagi-Sugeno (Mod eTS) algorithm

The Evolving Takagi-Sugeno (eTS) algorithm [15] is based on the Subtractive algorithm with modifications which allow the gradual update of the antecedent part of the fuzzy If-Then rules, as well as the consequent parameters through a global learning via weighted Recursive Least Squares (RLS) algorithm. The local learning, described in more details in [2] can also be used, which proves to have more locally interpretable rules and sometimes

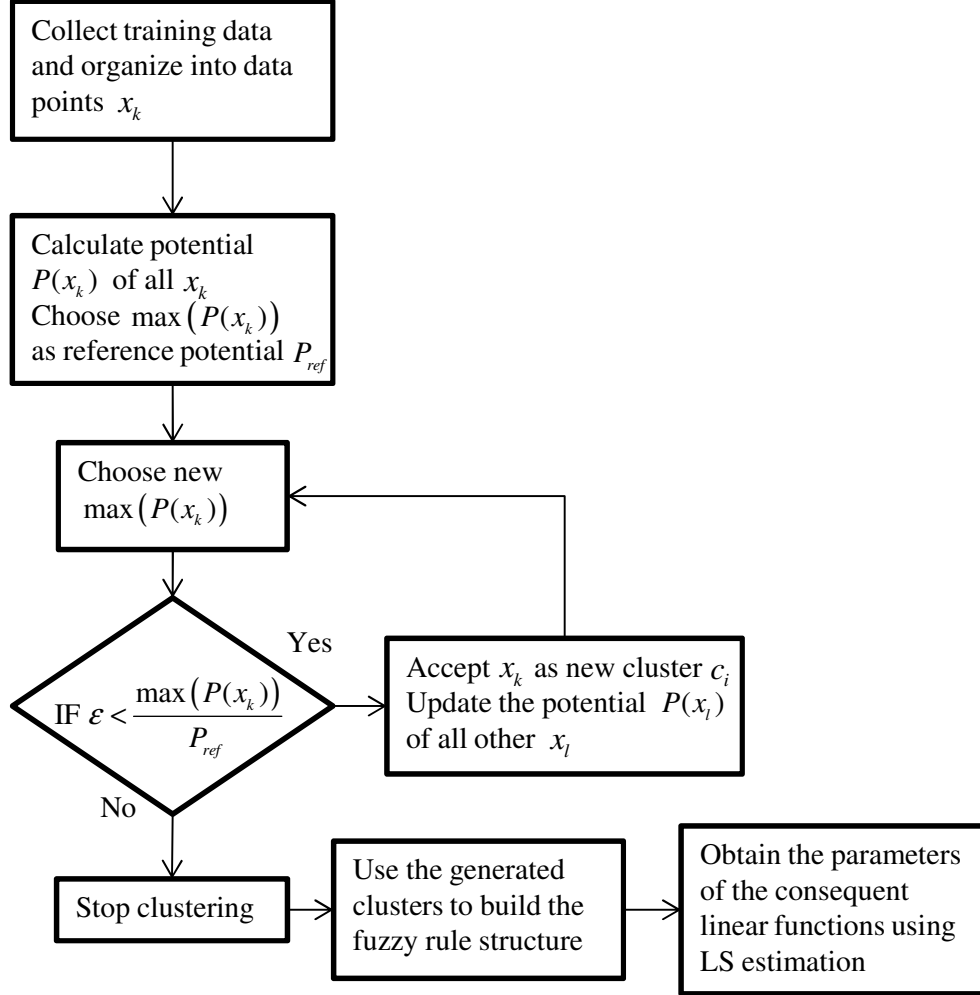


Fig. 1. The flow-chart of the Subtractive clustering algorithm.

may produce better results. In this description however, for the sake of simplicity, the global learning is demonstrated. Both global and local learning versions of the algorithm are later compared in terms of forecasting accuracy. In the classic eTS each cluster has one predefined radius. In this paper, the eTS incorporates the dynamic radius adjustment of each input variable of each cluster. Introducing a radius for each cluster dimension which is dynamically changing allows for a better coverage of the data by clusters, as radii may differ in different dimensions. It also limits the number of clusters

(and consecutively rules) as it is not necessary to create more clusters when data is already well covered by the existing ones. It is worth noting, that the idea of dynamic radii is not new, as it has been included in the extensions of the classic eTS approach in the form of exTS in [17] and eTS+ in [2]. The comparison of the radii updates in different fuzzy evolving algorithms, including Mod eTS is given in Table 1.

Table 1

Comparison of the different radii updates in different fuzzy evolving algorithms.

Method	Characteristic	Equation
exTS	use of the	$r_{k,ij} = \alpha r_{(k-1),ij} + (1 - \alpha) s_{k,lj},$
	scatter	l - closest cluster index, α - parameter, $s_{k,lj}$ - scatter: $s_{k,ij} = \sqrt{\frac{1}{N_{k,i}} \sum_{l=1}^{N_{k,i}} (c_{ij} - x_{lj})^2},$ $N_{k,i}$ - support of the i^{th} cluster
eTS+	use of the	$r_{k,ij} = \alpha r_{(k-1),ij} + (1 - \alpha) s_{k,ij},$
	scatter	α - parameter, $s_{k,ij}$ - scatter: $s_{k,ij} = \sqrt{\frac{1}{N_{k,i}-1} \sum_{l=1}^{N_{k,i}-1} (c_{ij} - x_{lj})^2},$ $N_{k,i}$ - support of the i^{th} cluster
Mod eTS	use of the cluster centre	$r_{k,ij} = \gamma r_{(k-1),ij} + (1 - \gamma) c_{ij},$ γ - parameter

Data points which are outside of the radii have little or no influence on the potential of this cluster [14]. As the cluster structure is evolving all the time, a different way of calculating the potential of the data samples needs to be introduced. The potential is calculated based on the Cauchy function (the approximation of Gaussian function)[31]. This particular type of function is used as it is inversely proportional to the distance and allows for recursive calculation [32]:

$$P_k(x_k) = \frac{1}{1 + \frac{1}{k-1} \sum_{l=1}^{k-1} \sum_{j=1}^{h+1} (x_{lj} - x_{kj})^2} \quad (14)$$

This function is inversely proportional to the squared distances between the current data point and the previously collected data. The potential increases with the number of the previous data points being in the proximity of the

current data point. The potential is also influenced by the number of currently available points; the more the points the smaller the term $1/(k-1)$ is, which leads to an overall increase in the potential. This feature allows data points which describe the region of data better than some of the already created clusters to become new clusters for that region.

The algorithm can be initiated with a previously generated cluster structure or it can start from the first obtained data point. The data points should be organized in row vectors with inputs followed by resulting output:

$$x_k = [x_{k1} \ x_{k2} \ \cdots \ x_{kj} \ \cdots \ x_{kh} \ y_k] \quad (15)$$

where k is the data point index, $j = 1, \dots, h$ is the input index and y is a resulting output. In addition to this, the initial values of the spread or cluster radius $r_{1,1,j}$ for each input and output needs to be set up in advance. The radii will be updated dynamically as new clusters are added to a cluster structure. The radii are updated using the parameter γ which is set in advance empirically.

The proposed algorithm contains the following steps.

1. Normalise the incoming data point x_k so that the range of all the data points, $x_{kj} \in [0, 1]$:

$$x_{kj} = \frac{x_{kj} - \min(x_{\underline{j}})}{\max(x_{\underline{j}}) - \min(x_{\underline{j}})} \quad (16)$$

The $\min(x_{\underline{j}})$ and $\max(x_{\underline{j}})$ denote the minimum and the maximum of all previously obtained points of j^{th} variable (component of data point x_k). As the variables differ in their maximum and minimum values, it is recommended that they are normalized [30]. Although the information of the maximum and minimum values of data may not be available for the user in real life, it is often possible to estimate the maximum and minimum values from some subset (training set) of available data or those values can be chosen empirically based on the expertise. Another approach to that problem has been described in [30] where the normalization was done through normalization constants for each variable, and in [2], where standardization was done based on the recursive calculation of mean and standard deviation.

2. If the algorithm starts from an empty rule base and the first data point x_1 is considered, then:

- a) Initialize the cluster structure c_1 with the first data point $x_1 : c_1 = x_1$
- b) Set the potential $P_1(c_1)$ of the first cluster to 1: $P_1(c_1) = 1$.
The formula for the potential $P_k(x_k)$ is given by:

$$P_k(x_k) = \frac{k-1}{(k-1)(\vartheta_k+1) + \sigma_k - 2\nu_k} \quad (17)$$

where:

$$\begin{aligned} \sigma_k &= \sum_{l=1}^{k-1} \sum_{j=1}^{h+1} (x_{lj})^2 = \sigma_{k-1} + \vartheta_{k-1} \\ \vartheta_k &= \sum_{j=1}^{h+1} (x_{kj})^2 \\ \beta_{kj} &= \sum_{l=1}^{k-1} x_{lj} = \beta_{k-1j} + x_{k-1j} \\ \nu_k &= \sum_{j=1}^{h+1} x_{kj} \beta_{kj} \end{aligned} \quad (18)$$

Initialize the values used to calculate the potential (Eq. (17)) of the data point in the next steps: $\vartheta_k = 0, \sigma_k = 0, \nu_k = 0, \beta_k = [0 \ 0 \ \dots \ 0]$, where size of the β_k vector is the same as the size of x_k

- c) Create the first rule based on the cluster centre c_1 :

$$\begin{aligned} &\textbf{IF } x_{11} \textbf{ is } A_{11} \textbf{ AND } x_{12} \textbf{ is } A_{12} \textbf{ AND } \dots \textbf{ AND } x_{1h} \textbf{ is } A_{1h} \\ &\textbf{THEN } y_1 = a_{11}x_{11} + a_{12}x_{12} + \dots + a_{1h}x_{1h} + a_{1h+1} \end{aligned} \quad (19)$$

with A_{1j} being a Gaussian membership function calculated using Eq. (6).

- d) Set the number of rules R to 1: $R = 1$
- e) Calculate the variance σ^2 (σ measures the width of the Gaussian function) of the Gaussian membership functions (Eq. (6)) of each fuzzy set in the first cluster based on the starting cluster radius $r_{1,1j}$:
$$\sigma_{1j}^2 = \frac{r_{1,1j}(\max(x_{-j}) - \min(x_{-j}))}{\sqrt{8}}.$$
- f) Set the parameter vector θ_1 of the consequent linear equation of the first rule to 1: $\theta_1 = [1 \ 1 \ \dots \ 1]^{h+1}$, where $h+1$ indicates the number of columns in the vector (h being a number of inputs)
- g) Set the global parameter matrix Θ of the consequent linear equations to $\theta_1, \Theta = \theta_1$
- h) Set the fuzzy weight λ_1 to 1

- i) Initialize covariance matrix Cov_1 with a high number Q multiplied by the $(h + 1) \times (h + 1)$ identity matrix:

$$Cov_1 = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix}^{h+1 \times h+1} \quad (20)$$

- j) Initialize a regressor vector ψ_1 which will store weighted input values for the parameter estimation purpose $\psi_1 = [0 \ 0 \ \dots \ 0]^{h+1}$, $\Psi_1 = \psi_1$
3. If the cluster and rule structure is already initiated or $k > 1$:
- a) Calculate the value of the potential $P_k(x_k)$ of the data point x_k using the recursive version of the Cauchy type function (Eq. (14))
- b) Update the radii of all created clusters:

$$r_{k,ij} = \gamma r_{(k-1),ij} + (1 - \gamma) c_{ij} \quad (21)$$

The update of the clusters is conducted in every step for all clusters. Each dimension (input variable) of the cluster is updated separately. Similar approach was implemented in [17] and [2], where radius was calculated using the local scatter over the input data which resembles the variance. In this case we do not calculate the scatter, but instead use the value of the centre in the considered dimension. This approach has been dictated by the importance of high values of inputs and is problem specific. The higher the cluster centre value is, the bigger the radius and the higher the resulting potential will be. This will promote the clusters created in the regions with high values which is crucial for leakage application as high values of leakage contribute significantly to the average leakage over the year. It is worth noting, that it is also important to choose appropriate smoothing parameter γ and initial value of the radius.

- c) Update the potential of all already established clusters. The potentials of already created centres depend on the distances to all data points, therefore, it is necessary to update them.

$$P_k(c_i) = \frac{(k - 1) P_{k-1}(c_i)}{k - 2 + P_{k-1}(c_i) + P_{k-1}(c_i) \sum_{j=1}^{h+1} \left(\frac{c_{ij} - x_{kj}}{r_{k,ij}} \right)} \quad (22)$$

d) The following steps carry on the process of deciding when to add a new cluster, change the already existing one or when to leave the cluster structure unchanged:

- 1: Compare the potential of the candidate cluster $P_k(x_k)$ with the updated potentials of all previously selected clusters
- 2: **if** $P_k(x_k) > \max(P_k(c_i))$ **or** $P_k(x_k) < \min(P_k(c_i))$ **then**
- 3: $d_{\min} = \min_i(dist_i)$, where $dist_i = \left\| \frac{c_i - x_k}{r_i} \right\|^2$, d_{\min} is the minimum distance $dist_i$ between the data point x_k and the cluster with the centre c_i , r_i is the radii of the i^{th} cluster;
- 4: **if** $d_{\min} < 0.5$ **then** {change the already existing cluster}
- 5: The closest cluster centre c_i is replaced by the current data point x_k (the corresponding rule is also changed);
- 6: The potential of the changed cluster is replaced by the potential $P_k(x_k)$ of the data point x_k ;
- 7: The linear parameters of the consequent part of the rule which was created from the replaced cluster remain the same, as well as the covariance matrix Cov_k ;
- 8: **else** {add new cluster}
- 9: New cluster is added with the coordinates of data point x_k and the potential $P_k(x_k)$;
- 10: $R = R + 1$;
- 11: The initial vector of linear parameters θ_R is obtained based on the weighted average of all vectors from remaining fuzzy rules;

$$\theta_R = \sum_{i=1}^{R-1} \lambda_i \theta_i \quad (23)$$

- 12: The global covariance matrix Cov_k needs to be extended and reset (see Eq. (24) with $\rho = (R^2 + 1) / (R^2)$ being a resetting factor based on the current number of rules R and cov representing the elements of the covariance matrix at step $k - 1$;
- 13: **end if**
- 14: **else**
- 15: Ignore the data point x_k and proceed further;
- 16: **end if**

$$Cov_k = \begin{bmatrix} \rho cov_{11} & \cdots & \rho cov_{1R(n+1)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \rho cov_{R(n+1)1} & \cdots & \rho cov_{R(n+1)R(n+1)} & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & Q & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & Q \end{bmatrix} \quad (24)$$

4. Update the parameters of the consequent part using the global RLS algorithm.
 - a) Update the data matrix Ψ_k using the firing degrees $\lambda_1, \lambda_2, \dots, \lambda_R$ calculated using Eq. (7-8) and the extended input data vector x_k^e (Eq. (10)), $\tau_i = [\lambda_i x_{k1} \quad \lambda_i x_{k2} \quad \cdots \quad \lambda_i x_{kh} \quad \lambda_i]$, $i = 1, \dots, R$:

$$\Psi_k = [\tau_1^T \quad \tau_2^T \quad \cdots \quad \tau_R^T]^T \quad (25)$$

- b) Apply the RLS algorithm to estimate the parameters of the linear consequent part of the fuzzy If-Then rules:

$$L = \frac{Cov_k \Psi_k}{1 + \Psi_k^T Cov_k \Theta_{k-1}} \quad (26)$$

$$\hat{\varepsilon} = y_k - \Psi_k^T \Theta_{k-1} \quad (27)$$

$$\Theta_k = \Theta_{k-1} + L \hat{\varepsilon} \quad (28)$$

$$Cov_{k+1} = Cov_k - L \Psi_k^T Cov_k \quad (29)$$

5. Estimate the output for the next period based on the input values and obtained parameter estimates given in Eq. (13) or in the vector form:

$$\hat{y}_{k+1} = \Psi_{k+1}^T \Theta_k \quad (30)$$

6. Collect next data vector and go to step 3.

Fig. 2 illustrates the flow-chart of the Mod eTS. The algorithm can be tailored to be used in the forecasting application which is discussed in the next section.

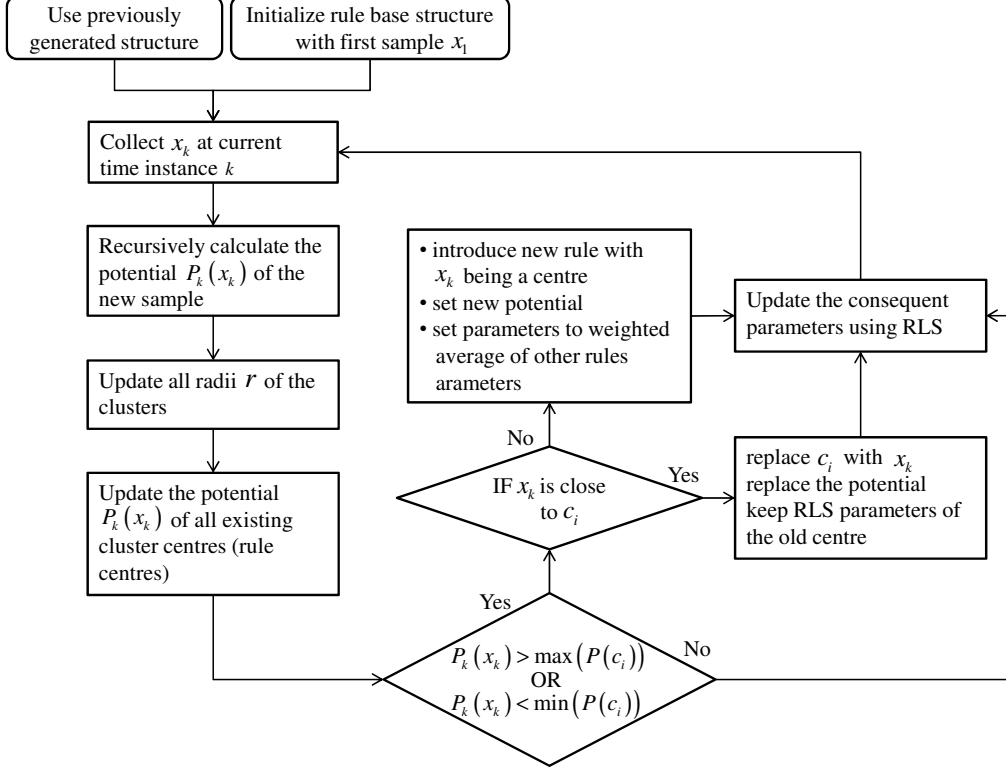


Fig. 2. The flow-chart of the Mod eTS algorithm.

3.3. Mod eTS algorithm in forecasting

The algorithm described in Section 3.2 can be adapted and used in multi-variate forecasting for n -periods ahead. First, the available historical data should be arranged into data points forming data vectors. The algorithm can start from the first data point and the cluster formed by the first data point, and afterwards, the rule structure is gradually generated through an evolving process until all k data points are assessed. Assuming that the input values $x_{(k+1)1}, x_{(k+1)2}, \dots, x_{(k+1)h}$ are given, the forecast for the next period \hat{y}_{k+1} can be calculated using the generated fuzzy If-Then rules and the estimated parameters through Takagi-Sugeno inference (Eq. (30)). In the next step the vector of data is created using the obtained forecast \hat{y}_{k+1} in place of the output, i.e.:

$$x_{k+1} = [x_{(k+1)1} \quad x_{(k+1)2} \quad \cdots \quad x_{(k+1)h} \quad \hat{y}_{k+1}] \quad (31)$$

The new data point x_{k+1} is then used to update the rule structure through the modified eTS as described in Section 3.2. The approach is similar to the rolling forecast method where generated forecasts for period k are used to obtain a forecast for the next period $k + 1$. The process continues until all desired forecasts $\hat{y}_{k+1}, \dots, \hat{y}_{k+n}$ are obtained. The flow-chart of the Mod eTS algorithm for forecasting is given in Fig. 3.

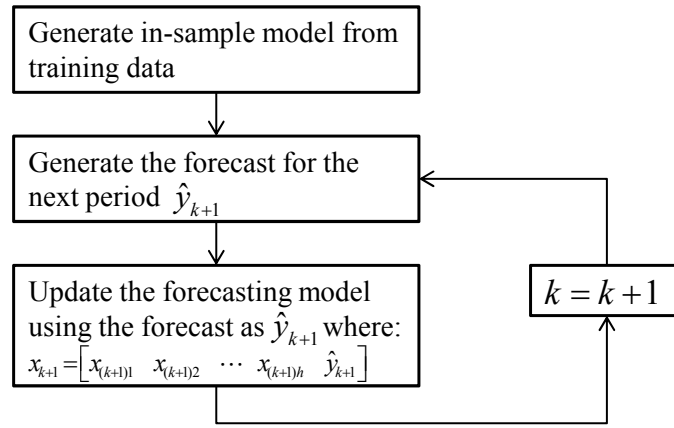


Fig. 3. The flow-chart of the Mod eTS algorithm for forecasting.

4. Case study

4.1. Setting the scene

One of the main concerns of water companies nowadays is maintaining required leakage levels. This is an issue especially visible in countries with an old water network, such as the UK. A considerable part of the UKs water supply networks date back to the beginning of the 20th century and experience frequent interruption in services due to structural failures [33]. As all of the water service companies in the UK have been privatized, the Office of Water Services (OFWAT) has been established to make sure that these companies provide a certain level of service. Water companies are required to present regular leakage forecasts and plan the resource effort in order to decrease it to an acceptable level.

Leakage may be divided into two groups. Background leakage (apparent losses) accounts for utility operation (company own water usage), meter inaccuracies and data errors; these losses cannot be measured with the current technology. Real losses, on the other hand, are connected with physical losses of water from the distribution network through leakage, but also through storage overflows or unaccounted use of water and can be controlled by frequent checks and proper maintenance.

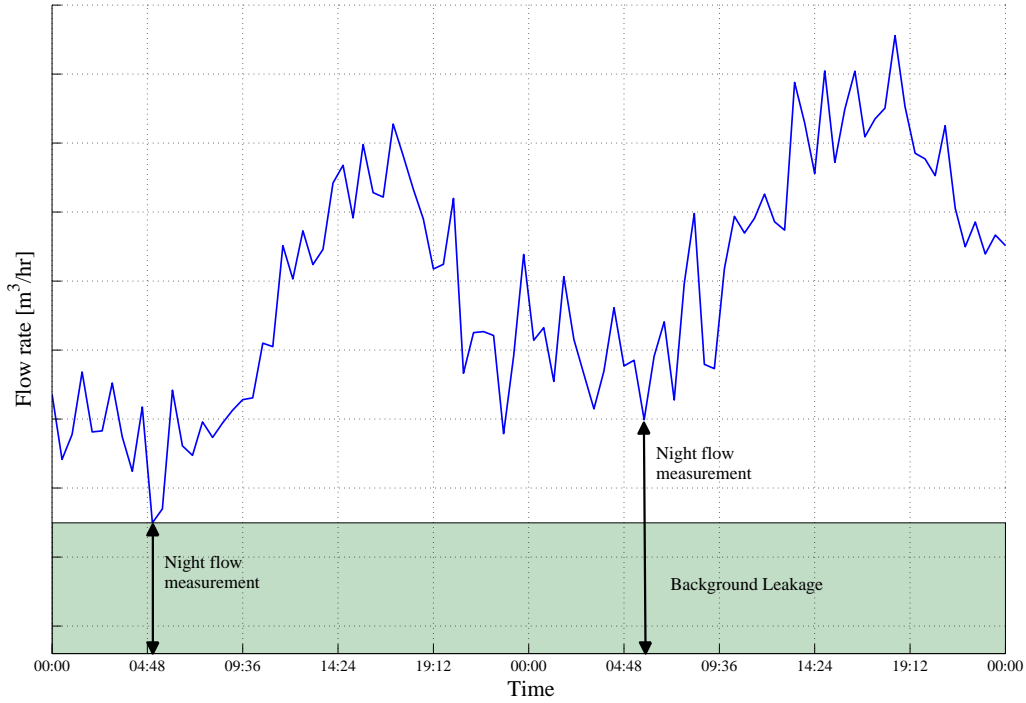


Fig. 4. Leakage distribution throughout the day.

The level of leakage is usually estimated through night flow measurements (Fig. 4), where flow rates during the night, when the water demand is usually at its lowest level, are compared on a day by day basis. Any difference in the night flows may be considered as a leakage and is investigated. Although considerable investments have been made, lack of metered households is a major factor of errors in the leakage estimation as it is very difficult to distinguish between sudden increase in demand (for example, during a hot summer period) and the actual leakage.

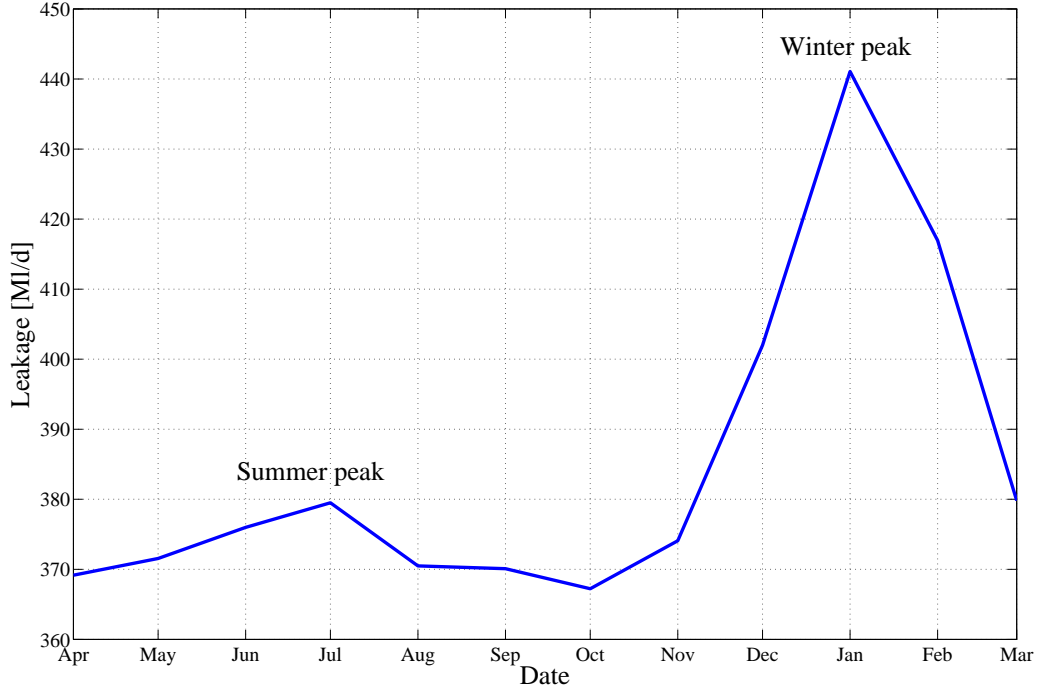


Fig. 5. Typical leakage distribution throughout the year.

Another factor which makes leakage detection and estimation difficult is that it does not remain constant throughout the year and is prone to seasonal factors such as temperature and rainfall. Typically, two peaks may be distinguished during the year (Fig. 5). The smaller peak occurs around the summer period and may be explained by an increase in demand due to unaccounted night use and an increase in temperature which causes pipes to expand and results in displacements, leading to pipe bursts. The much more significant peak is usually observed in the winter, when sudden decreases in temperature over few days cause extensive pipe bursts.

4.2. Forecasting the leakage

Leakage forecasts are produced one year ahead, from April until March, as this is a requirement set by OFWAT. Together with the expected leakage, the company presents the resource effort expressed in the form of Equivalent Service Pipe Bursts (*ESPBs*). *ESPB* relates to the number of leaks found by engineering teams (*ESPB detected*) or reported by customers (*ESPB reported*).

Different types of pipes have different flow rates and therefore the leakage from these pipes cannot be compared in a straightforward manner. Therefore, *ESPBs* represent monthly repair figures recalculated to account for different flow rates.

$$ESPB_{type} = leaks_{type} \times \frac{flowRate_{type}}{flowRate_{CSP}} \quad (32)$$

where *type* is a type of a pipe (Mains, Communication, etc.), *leaks* is the number of leaks, *flowRate* (in $\frac{m^3}{hr}$) is a flow rate and *flowRate_{CSP}* is a flow rate of Communication Service Pipe which is used as a reference value for recalculation.

In addition to *ESPB* numbers, the Natural Rate of Rise (*NRR*) is used to indicate seasonality in the leakage. *NRR* relates to the underlying rate at which leakage increases within a network in the absence of any leak repairs. It is calculated based on each years expected starting leakage, *NRR Profile*, which accounts for seasonal factors (peaks in summer and winter periods) and annual *NRR* (an overall increase in leakage throughout the year). Similar to *ESPBs*, it is also created for detected and reported leakages and the sum of both represents total *NRR* or *NRR_t*.

4.3. Issues with producing forecasts

In general, producing forecasts for a long period ahead, such as 12 months is a difficult task, especially when one of the crucial factors is weather. One of the ways to overcome this problem is to rely on historical data and the assumption that leakage can be influenced by the number of leaks found and fixed through detection and customer feedback (*ESPB* numbers). Relying on the weather forecasts is not feasible, as 12 month ahead weather forecasts are not sufficiently accurate. Instead, the seasonal factors, such as general decrease of temperature during the winter, or less rain during the summer, are included in the *NRR* figures which increase throughout the year with a high peak in a winter season. Another issue is the general influence of *ESPB* numbers (both detected and reported) and the relationship between them. It is difficult to establish if investment made in detecting leaks and, therefore, increasing *ESPB detected* numbers has a great influence on leakage values or if it is due to other factors including *ESPB reported* numbers. The approach to that problem taken in this paper is to gradually group the data vectors containing all relevant input variables into clusters which share similar properties in order to compensate for these uncertain relationships between them.

5. Results and discussion

The Mod eTS algorithm is applied to the leakage forecasting problem. The historical data consists of 9 real-world datasets and was kindly provided by Severn Trent Water, one of the leading water supplying companies in the UK. It includes 60 monthly readings of leakage, *ESPB detected*, *ESPB reported* and *NRR* values collected over 5 years (2006 – 2011). The leakage values were collected from District Metered Areas (DMAs) on a weekly basis and were averaged for each month of the year. The *ESPB* values were summed each month and split between detected and reported cases. *NRR* values were provided by the company on a monthly basis. Eight regional and one overall company dataset were used to test the forecasting algorithm proposed.

The datasets are divided into two parts: 4 years (48 months) of data is used for a training period and 1 year (12 months) for out of sample testing. Data vectors $x_k : k = 1, \dots, 48$, are created from each dataset and consist of four input values: y_{k-1} - previously obtained leakage, $u1_k$ - *ESPB detected* numbers, $u2_k$ - *ESPB reported* numbers and $u3_k$ - *NRR* values. The output value y_k is observed leakage.

$$x_k = [y_{k-1} \quad u1_k \quad u2_k \quad u3_k \quad y_k] \quad (33)$$

The choice of the parameters to include as inputs was dictated by the availability of the data and the discussion we had with the company experts. The cluster structure is built using training data of 48 months separately for each of 9 datasets, using the algorithm described in Section 3.2. The forecasting process is carried out as described in Section 3.3, using the last 12 months of data for a 12 month ahead period on a monthly basis. We use three typical measures of error.

R^2 is called the coefficient of determination and represents how well the estimated forecast value fits the actual value of leakage. The higher the value of R^2 , the greater the similarity to the historical observations:

$$R^2 = 100 \left(1 - \frac{\|y - \hat{y}\|^2}{\|y - \bar{y}\|^2} \right) \quad (34)$$

where \hat{y} is obtained forecast vector and \bar{y} is the mean of all historical values of leakage. When the value of R^2 is below zero it means that the fit is worse than the simple mean of the historical values.

We also calculate the Mean Absolute Percentage Error ($MAPE$) and Root Mean Squared Error ($RMSE$), which are amongst the most used error measures. In this case, the lower the values of the $MAPE$ and $RMSE$ (the lower the forecasting error) the better:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (35)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (36)$$

$MAPE$ error measure is used due to its ability to express the result regardless of the scale at which the measure is computed as the result is presented in percentage. Also, the considered time-series have no zero values, which makes the use of $MAPE$ possible. $RMSE$ is used as the error is expressed in the same scale as the data.

5.1. Comparison with classical fuzzy and some statistical forecasting methods.

To assess the accuracy of the Mod eTS algorithm we compare it with some other fuzzy forecasting methods: standard eTS, which does not involve cluster radius update, Subtractive clustering algorithm (Subclust) and C-means fuzzy forecasting algorithm described in [11]. We also use some standard statistical methods: Multiple linear regression (MLR), Seasonal Naive (S-Naive) method, which takes into account the last 12 month values of leakage and simply applies it as a forecast for the next period, Autoregressive model of order 1 AR(1), as well as Holt-Winters method (HWM) which is one of the most popular and successfully applied exponential smoothing methods which can address both trend and seasonality. The memory requirements and computational speed are not included as performance indicators in this study, but the algorithms in general do not put high stress on the PC and memory shortages were not an issue. The parameters of the algorithms are optimized based on the MAPE error computed over both training and testing data.

Fig. 6 shows a plot of the real leakage and the input variables together with the forecast obtained using the Mod eTS, eTS and C-means methods for one of the regions of company operation. Table 2 shows the accuracy results using different forecasting methods and 3 different error measures for training and

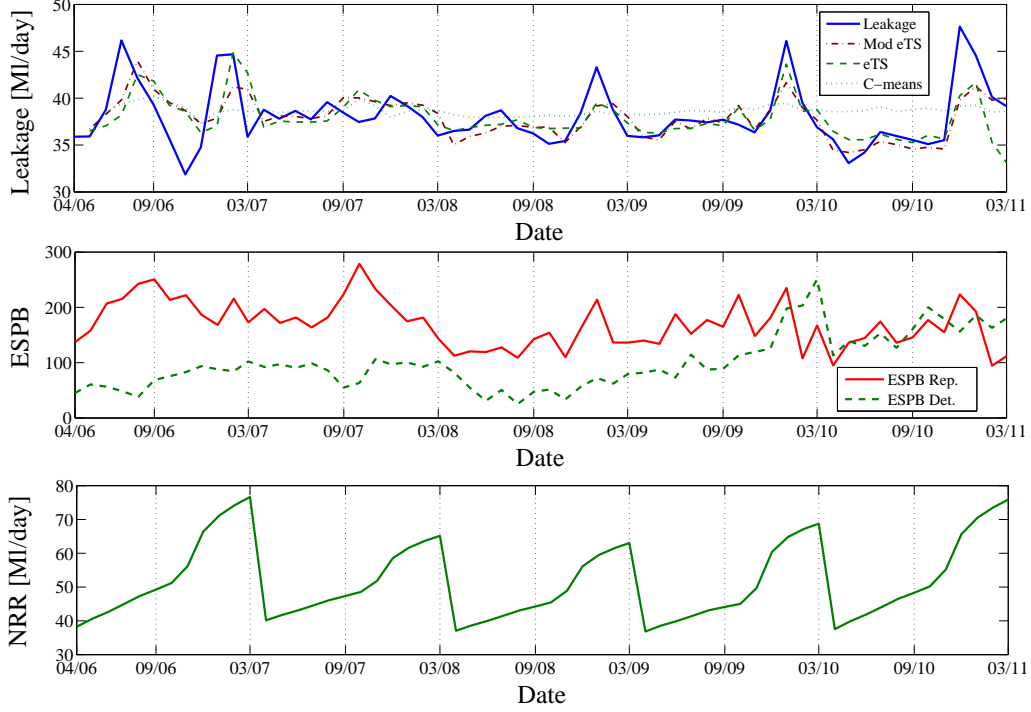


Fig. 6. Leakage, forecast, ESPB and NRR plot of 5 years of data for one of the regions of operation.

testing periods. It can be seen that Mod eTS has the highest accuracy out of all tested methods in out-of-sample period considering all 3 error measures with lowest number of clusters out of all fuzzy forecasting methods.

To assess the accuracy of the method over a bigger number of datasets we calculate the accuracy of the forecast for all 9 datasets and then take the average of the results obtained. These are presented in Table 3. The results show that the proposed method produces forecasts with higher R^2 values and lower $MAPE$, in terms of testing (out-of-sample data) than other fuzzy forecasting methods and most statistical methods. It is worth noting that the values of leakage over the last 12 months of the training period are very similar to the leakage of the testing period. This results in HWM having the highest accuracy in the testing periods considering all 3 accuracy measures due to 12 month seasonality value considered in HWM. This may not always be the case, as depending on weather and the control policy, the leakage may

Table 2

Accuracy results of different fuzzy and statistical forecasting methods for a dataset from one of the regions.

Method	R^2 train	R^2 test	$MAPE$ train	$MAPE$ test	$RMSE$ train	$RMSE$ test	clusters
Mod eTS	26.53	62.32	4.48	3.83	2.46	2.58	2
eTS	20.00	37.79	4.43	5.72	2.57	3.32	3
Subclust	100.00	-436.08	0.00	15.11	0.00	9.74	35
C-means	7.20	0.73	5.32	9.70	2.78	4.19	7
MLR	22.40	35.45	4.41	6.88	2.53	3.38	
S-Naive	-97.46	42.29	5.51	5.92	2.90	3.20	
AR(1)	11.91	0.83	4.74	9.04	2.69	4.19	
HWM	51.52	43.41	3.33	4.82	1.67	3.16	

Table 3

Accuracy results of different fuzzy and statistical forecasting methods averaged over 9 datasets.

Method	R^2 train	R^2 test	$MAPE$ train	$MAPE$ test	$RMSE$ train	$RMSE$ test	clusters
Mod eTS	57.78	21.81	3.57	7.47	3.89	11.88	3
eTS	52.54	-5.24	3.13	8.25	5.04	11.50	10.1
Subclust	97.55	-120.47	0.31	12.50	0.55	14.35	30.3
C-means	-5.88	-8.25	5.99	10.12	6.55	12.67	7
MLR	54.03	-26.57	3.65	10.30	3.98	14.20	
S-Naive	-42.80	17.79	6.59	7.38	7.38	9.33	
AR(1)	32.11	-35.53	4.13	10.97	5.22	13.34	
HWM	10.27	39.11	5.60	5.78	6.52	8.98	

be considerably different from year to year. It is also important to mention, that HWM uses only time-series data of leakage, without taking into account other input variables, therefore is not really suitable for this application due to the existing need of understanding the inherent relationships between the considered inputs. The forecasting error of Mod eTS for the training period is lower in comparison to the statistical methods, but *MAPE* is slightly higher than that of the Subtractive clustering and eTS algorithm which achieves lower *RMSE* errors and higher R^2 values over the training period, due to its off-line nature. This however comes at the cost of the higher number of generated clusters for those types of methods and higher forecasting errors for testing period, which is the most important measure in forecasting. The Mod eTS generates the least clusters on average over 9 datasets than all other considered fuzzy forecasting methods. Having less clusters and similar or higher accuracy indicates that the proposed method may be a suitable choice if a trade-off between the complexity and accuracy needs to be achieved. This is particularly important in this application, when not only lower error values are of paramount importance but also the simplicity of the forecasting model. Reducing unnecessary complexity contributes to better understanding of the relationship between the dependent and independent variables as it can be assumed that lower number of rules improves the overall interpretability of the forecasting model [1].

5.2. Comparison with other fuzzy evolving algorithms.

Table 4

Accuracy results of considered fuzzy evolving algorithms averaged over 9 datasets.

Method	<i>RMSE</i> train	<i>RMSE</i> test	<i>RMSE</i> test <i>min</i>	<i>RMSE</i> test <i>max</i>	<i>RMSE</i> test var	clusters train	clusters test
Mod eTS G	4.54	14.88	2.61	66.27	384.01	4.11	0.44
Mod eTS L	4.10	14.86	5.11	68.33	405.44	5.11	0.11
eTS	6.63	15.51	4.38	57.10	274.35	4.38	0.38
exTS	4.05	15.57	3.40	63.40	346.95	2.25	0.13
eTS+	4.05	15.43	3.40	63.40	352.47	2.00	0.00

The same datasets are used to compare Mod eTS with both local and global

learning and other evolving algorithms: eTS [15], exTS [17] and eTS+ [2]. In this case, the parameters of the algorithms are optimized with error calculated over the training period. The computed *RMSE* for the forecasts, together with min, max values and the variance are presented in Table 4. The best results for testing period are achieved by the Mod eTS for both global and local learning, followed by the eTS+. eTS+ however generated the least number of rules on average due to more advanced method of keeping control over the size of the rule base through the mechanism of disabling some of the obsolete rules. The minimum, maximum values of errors are also calculated for the test period, with Mod eTS in Global learning mode having the smallest *RMSE*. It might be interesting to notice that the variance of the errors shows that eTS has the smallest spread of errors out of all considered methods, with Mod eTS, both global and local learning, having the highest. A further study needs to be conducted over the bigger dataset to evaluate the performance of the algorithm.

Fig. 7 presents the evolution of the rule base for one of the datasets. Mod eTS in global and local mode will always have the same rule structure for a training period, as the only difference between them is the way the parameters of the resulting linear equations are optimized, which should not have influence on the cluster generation process. For testing, however, differences may occur, as the forecasting process involves the use of generated forecasts in the inputs, which may influence the way the clusters will be established. In this particular case eTS and eTS+ generated the same numbers of rules; however, the steps to achieve that number were different. Less rules have been obtained from Mod eTS and exTS, with both algorithms generally adding the new rule at the same data point. It is worth noting that in this case no clusters were generated during the testing period, which is not always the case for all tested datasets.

It is worth noting, that due to the way the radii are updated in Mod eTS, the obtained results may not be comparable for the problems with different time-series characteristics, for example, with varying trend.

6. Conclusions

In this paper, an evolving fuzzy algorithm (Mod eTS) with a modified recursive cluster radii update is proposed. A number of standard fuzzy and evolving algorithms and some widely accepted statistical methods are com-

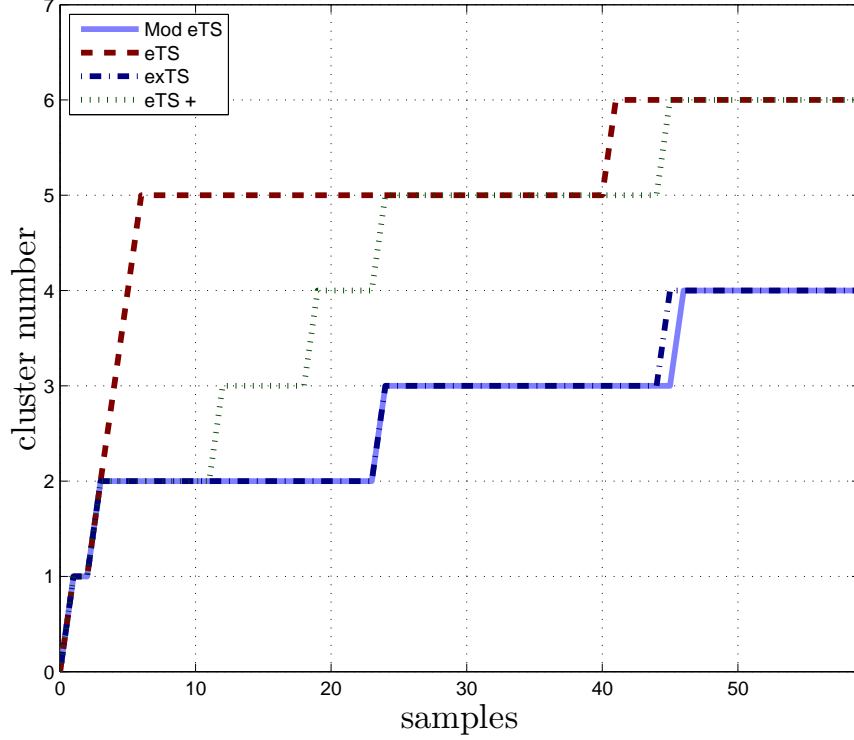


Fig. 7. Rule evolution of Mod eTS, eTS, exTS and eTS+ for one of the datasets.

pared and applied to a leakage forecasting problem for the first time. The proposed modification utilizes cluster centres in radii update to take into account the influence of data points with higher values due to the importance of the high leakage events. The recursive modification of cluster radii in each dimension in the evolving process allows for more accurate data coverage. Fuzzy If-Then rules are generated from clusters, which are used to obtain leakage forecasts. The modification results in a smaller number of clusters compared to standard fuzzy forecasting methods without the dynamic radii update, which improves the interpretability of the fuzzy rule structure. The algorithm can be used effectively in forecasting and performs well on the tested datasets. Results obtained by applying the method to real-world leakage data indicate that the proposed method generally performs better than

other methods for testing data. Compared with other standard fuzzy clustering forecasting methods the Mod eTS generates less rules, which yielded (in this case) better results for test data. The comparison with other fuzzy evolving algorithms, which also utilize radii update, showed slightly lower *RMSE* values for testing period but generated more rules. The benefits of local learning over the global approach in terms of the forecasting accuracy are shown. Good results achieved for the testing period using standard time-series methods such as S-Naive and HWM can be explained by similar leakage patterns in the testing period and the last season of training period, however, this may not always be the case.

Further work needs to be conducted to understand the relation between *ESPB* detected and leakage as it seems that it does not have a huge influence on the leakage values. Work will also be done to improve rule management in terms of removing obsolete and inactive rules, to further simplify the obtained model. Maintaining leakage targets in the forecast is crucial for the company and will be under consideration in the future. Finally, forecasting accuracy achieved by applying Mod eTS will be verified on other datasets as well, to assess a problem specific approach to rule update algorithm.

Acknowledgements

The authors would like to acknowledge the help and support provided by Severn Trent Water and, in particular, Mr Alan Tripp and Mr Pat Spain for their time and expertise, as well as providing the data which was used to conduct the experiments.

References

- [1] E. Lughofer, *Evolving Fuzzy Systems Methodologies, Advanced Concepts and Applications*, volume 266 of *Studies in Fuzziness and Soft Computing*, 1st ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [2] P. Angelov, D. P. Filev, N. Kasabov (Eds.), *Evolving Intelligent Systems*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2010.
- [3] E. Shnaider, A. Kandel, The use of fuzzy set theory for forecasting corporate tax revenues, *Fuzzy Sets and Systems* 31 (1989) 187–204.

- [4] Q. Song, B. S. Chissom, Fuzzy time series and its models, *Fuzzy Sets and Systems* 54 (1993) 269–277.
- [5] Q. Song, B. S. Chissom, Forecasting enrollments with fuzzy time series Part I, *Fuzzy Sets and Systems* 54 (1993) 1–9.
- [6] Q. Song, B. S. Chissom, Forecasting enrollments with fuzzy time series Part II, *Fuzzy Sets and Systems* 62 (1994) 1–8.
- [7] S.-M. Chen, Forecasting enrollments based on fuzzy time series, *Fuzzy Sets and Systems* 81 (1996) 311–319.
- [8] S.-M. Chen, Forecasting enrollments based on high-order fuzzy time series, *Cybernetics and Systems* 33 (2002) 1–16.
- [9] S.-M. Chen, K. Tanuwijaya, Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques, *Expert Systems with Applications* 38 (2011) 10594–10605.
- [10] G. Cardoso, F. Gomide, Newspaper demand prediction and replacement model based on fuzzy clustering and rules, *Information Sciences* 177 (2007) 4799–4809.
- [11] S.-M. Chen, Y.-C. Chang, Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques, *Information Sciences* 180 (2010) 4772–4783.
- [12] N. Pal, K. Pal, J. Keller, J. Bezdek, A possibilistic fuzzy c-means clustering algorithm, *IEEE Transactions on Fuzzy Systems* 13 (2005) 517–530.
- [13] R. R. Yager, D. P. Filev, Approximate clustering via the mountain method, *IEEE Transactions on Systems, Man, and Cybernetics* 24 (1994) 1279–1284.
- [14] S. L. Chiu, Fuzzy model identification based on cluster estimation, *Journal of intelligent and Fuzzy systems* 22 (1994) 827–841.
- [15] P. Angelov, An approach for fuzzy rule-base adaptation using on-line clustering, *International Journal of Approximate Reasoning* 35 (2004) 275–289.

- [16] P. Angelov, D. P. Filev, Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models, The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05. (2005) 1068–1073.
- [17] P. Angelov, X. Zhou, Evolving Fuzzy Systems from Data Streams in Real-Time, in: 2006 International Symposium on Evolving Fuzzy Systems, volume 00, IEEE, 2006, pp. 29–35.
- [18] S. W. Tung, C. Quek, C. Guan, SaFIN: a self-adaptive fuzzy inference network., IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 22 (2011) 1928–40.
- [19] N. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, Fuzzy Systems, IEEE Transactions on 10 (2002) 144–154.
- [20] D. Dovžan, I. Skrjanc, Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes., ISA transactions 50 (2011) 159–69.
- [21] D. Dovžan, I. Škrjanc, Recursive clustering based on a Gustafson-Kessel algorithm, Evolving Systems 2 (2010) 15–24.
- [22] H. Soleimani-B., C. Lucas, B. N. Araabi, Recursive Gath-Geva clustering as a basis for evolving neuro-fuzzy modeling, Evolving Systems 1 (2010) 59–71.
- [23] E. Lughofer, FLEXFIS: A Robust Incremental Learning Approach for Evolving TakagiSugeno Fuzzy Models, IEEE Transactions on Fuzzy Systems 16 (2008) 1393–1410.
- [24] E. Lughofer, J.-L. Bouchot, A. Shaker, On-line elimination of local redundancies in evolving fuzzy systems, Evolving Systems 2 (2011) 165–187.
- [25] P. Angelov, E. Lughofer, X. Zhou, Evolving fuzzy classifiers using different model architectures, Fuzzy Sets and Systems 159 (2008) 3160–3182.
- [26] P. Angelov, X. Zhou, Real-time joint Landmark Recognition and Classifier Generation by an Evolving Fuzzy System, in: 2006 IEEE International Conference on Fuzzy Systems, volume 44, IEEE, 2006, pp. 1205–1212.

- [27] J. Andreu, P. Angelov, Forecasting time-series for NN GC1 using Evolving Takagi-Sugeno (eTS) Fuzzy Systems with on-line inputs selection, in: International Conference on Fuzzy Systems, IEEE, Barcelona, 2010, pp. 1–5.
- [28] E. Lughofer, P. Angelov, Handling drifts and shifts in on-line data streams with evolving fuzzy systems, *Applied Soft Computing* 11 (2011) 2057–2068.
- [29] I. Luna, R. Ballini, Top-down strategies based on adaptive fuzzy rule-based systems for daily time series forecasting, *International Journal of Forecasting* 27 (2011) 708–724.
- [30] D. Dovzan, V. Logar, I. Skrjanc, Solving the sales prediction problem with fuzzy evolving methods, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012, pp. 1–8.
- [31] P. Angelov, D. Filev, An approach to online identification of Takagi-Sugeno fuzzy models., *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 34 (2004) 484–98.
- [32] P. Angelov, A fuzzy controller with evolving structure, *Information Sciences* 161 (2004) 21–35.
- [33] D. Savic, G. Walters, P. G. Ashcroft, A. Arscott, Hydroinformatics technology and maintenance of UK water networks, *Journal of Quality in Maintenance Engineering* 3 (1997) 289–301.

List of figures and tables with labels

Fig. 1. The flow-chart of the Subtractive algorithm.

Fig. 2. The flow-chart of the Mod eTS algorithm.

Fig. 3. The flow-chart of the Mod eTS algorithm for forecasting.

Fig. 4. Leakage distribution throughout the day.

Fig. 5. Typical leakage distribution throughout the year.

Fig. 6. Leakage, forecast ESPB and NRR plot of 5 years of data for one of the regions of operation.

Fig. 7. Rule evolution of Mod eTS, eTS, exTS and eTS+ for one of the datasets.

Table 1 Comparison of the different radii updates in different fuzzy evolving algorithms.

Table 2 Accuracy results of the different fuzzy and statistical forecasting methods for a dataset from one of the regions.

Table 3 Accuracy results of the different fuzzy and statistical forecasting methods averaged over 9 datasets.

Table 4 Accuracy results of considered fuzzy evolving algorithms averaged over 9 datasets.