ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250202002322
Original scientific paper

A MapReduce Approach to Model Big Data with Fuzzy Functions Identified Based on Fuzzy C-Means Algorithm

Ahmet ARTUT*, Adem GÖLEÇ

Abstract: Recently, big data has become increasingly important in the fields of scientific research and application. However, due to the characteristic features of big data such as high volume, velocity, variety, variability, value, and complexity, processing it with traditional analysis methods is quite a challenging process. In this context, frameworks like MapReduce are commonly used in the modeling of big data and in parallel and distributed data processing techniques. In this study, it is aimed to use fuzzy functions based on the fuzzy c-means (FCM) algorithm under the MapReduce architecture for modeling systems based on large data sets. In the study, it is explained in detail how the FCM algorithm is parallelized in the mapping phase; subsequently, it is demonstrated how the data is reduced in the reduce phase and how the fuzzy functions are derived. The proposed approach demonstrates the effectiveness of fuzzy functions within the MapReduce framework in modeling systems based on various large datasets. Additionally, the success of the methodology has been thoroughly discussed through the evaluation of the obtained fuzzy functions and performance analysis.

Keywords: big data, fuzzy c-means, fuzzy functions, map reduce, systems modeling

1 INTRODUCTION

Clustering is the process of dividing data with similar characteristics into subsets based on the properties of the objects, thereby separating them into homogeneous groups [1]. Unlike traditional statistical methods, most clustering algorithms do not rely on the statistical distribution of the data. Therefore, they can be effectively applied in situations where there is very little prior knowledge [2]. Clustering algorithms emerge as important tools in detecting and mathematically characterizing the behavior of systems in various disciplines. Representing a complex system with a mathematical model is becoming increasingly difficult and is not always possible. Hybrid algorithms that integrate clustering techniques with fuzzy set theory offer both simple and powerful approaches to modeling complex systems. The clustering process helps represent the essence of hidden relationships within data sets and contributes to a deeper understanding of underlying structures by enabling the identification of natural groupings. At the same time, fuzzy sets provide a robust mechanism for dealing with uncertainty and enable the extraction of precise conclusions from observations that inherently contain uncertainty. One of the challenges of modeling systems with big data is the uncertainty in the existing datasets. Therefore, the main strategy for effectively capturing and representing systems associated with uncertain and vague data is the use of fuzzy system models. This system modeling approach was first proposed by Zadeh. Fuzzy system models are created by defining the relationships between system inputs and outputs using fuzzy sets [3]. The most commonly used classical approach in fuzzy system modeling is the "fuzzy rule base", which are verbal structures that describe the relationships between system inputs and outputs. The fuzzy rule base explains the relationship between the input and output with rules expressed in the form of "IF... THEN" [4]. Initially proposed by Zadeh, this approach was later developed with new methodologies suggested by two different research teams. In the method proposed by the first research team, Sugeno and Yasukawa, fuzzy sets are defined for both input variables (antecedents) and output variables (consequents). Generally, researchers use either linear or constant functions to express the consequent part. In this approach, both the antecedent part of the fuzzy rules (i.e., the "IF" part) and the consequent part (i.e., the "THEN" part) have been determined using expert knowledge and especially fuzzy clustering algorithms like FCM [5]. Similarly, the second research group, Takagi and Sugeno [6], made a significant contribution to the determination of fuzzy sets that form the antecedent part of fuzzy rules. In this process, fuzzy clustering algorithms such as FCM have been utilized in addition to the information obtained from experts. Additionally, they have added a new dimension to the fuzzy system modeling process by using functional approximation methods in predicting the output functions in the consequent part of the rules [6]. Classical fuzzy system modeling approaches typically rely on expert knowledge for creating rule-based fuzzy inference systems [7]. Therefore, to create self-learning systems by reducing expert intervention, Türkşen [8] proposed the fuzzy function approach instead of rule-based inference systems. Unlike rule-based fuzzy inference systems that express verbal relationships, the fuzzy function approach creates a functional modeling form corresponding to verbal modeling. Göleç and colleagues [9] emphasized the effectiveness of fuzzy functions and stated that the use of membership values in the inference system eliminates the need for a rule-based structure. When working with big data, distributed and parallel computing-based approaches are often preferred for the effective processing and analysis of data. One of the most popular frameworks for big data, the MapReduce approach, divides the processing into two main stages [10]. The first stage is the Mapping (Map) stage, which is responsible for partitioning the original dataset and processing each partition in parallel. The second stage is the Reduction (Reduce) stage. It is responsible for combining the results obtained from the mapping phase, as well as integrating any necessary new behaviors. This approach, while making the unique characteristics of the data more effective, can lead to some drawbacks when working with imbalanced datasets. The partitioning of the original dataset, especially during this process, causes the formation of small data subsets, which leads to the "data skew" problem [11]. Therefore, recognizing such challenges highlights the importance of

developing strategies to effectively manage imbalanced datasets in the context of parallel processing frameworks. In this study, the MapReduce-based FCM algorithm processed the data from systems with large datasets. The obtained fuzzy clusters were weighted using the rule weighting method proposed by Del Rio and colleagues [12]. By selecting the cluster with the maximum weight within each data section, a reduced data set has been obtained. Then, using this reduced dataset, a fuzzy system model was created with the fuzzy function approach proposed by Türkşen [8]. The other sections of the article are structured as follows: The second section presents a comprehensive literature review of the articles found in the relevant literature. In the third section, the fundamentals of big data and the MapReduce framework are explained. Then, in the fourth chapter, fuzzy system models are examined in detail. In the fifth chapter, the FFFCM-BigData algorithm for fuzzy system modeling is presented. Finally, the sixth chapter contains the experimental analysis.

2 LITERATURE REVIEW

Traditional clustering algorithms encounter significant limitations when processing large datasets. These algorithms exhibit insufficient scalability with high-volume data and have high computational costs in terms of processing time and memory usage. To address these issues, parallel clustering algorithms have emerged as an effective solution in big data applications. Ludwig [2] emphasized the importance of developing parallel clustering algorithms that are both effective and scalable, producing high-accuracy results for big data processing. Havens and colleagues [13] focused on the problem of classifying large datasets that exceed computer memory capacity; they proposed an extension of the FCM clustering method to enable its use with large-scale data and evaluated their developed method using three different techniques. Among these methods were non-sampling-based extensions, incremental techniques that make successive transitions on data subsets, and methods based on core versions of FCM that rely on sampling strategies.

Ayed and colleagues [14] conducted a comprehensive review of the clustering methods commonly found in the literature, comparing classical, fuzzy, and big data clustering algorithms. Additionally, they presented ideas for creating a scalable and noise-resistant clustering system based on type-2 fuzzy clustering methods. Del Rio and colleagues [12] proposed a Chi-FRBCS-BigData algorithm that uses the MapReduce framework for combining learning and rule bases. This algorithm is a linguistic fuzzy rule-based classification system. López and colleagues [15] developed a Chi-FRBCS-BigData algorithm that can effectively handle imbalanced large-scale data. This algorithm is a fuzzy rule-based classification system that addresses the uncertainty arising from big data sources and takes into account the learning of underrepresented classes. The method uses the MapReduce framework to distribute the computational tasks of the fuzzy model and integrates cost-sensitive learning techniques into the design to create a structure that reflects data imbalance. Azar and Hassanien [16], on the other hand, performed dimensionality reduction, feature selection, and classification tasks using a linguistic fuzzy logic-based artificial-biological classifier. To demonstrate the performance of their proposed artificial-fuzzy classifier, they used four real-world datasets and conducted a comparative evaluation with other classifiers on various classification problems. Ludwig [2] has studied the parallelizability and scalability of the FCM fuzzy clustering algorithm. In the parallelization process, the MapReduce paradigm was used, and it was explained how the principles of Mapping (Map) and Reducing (Reduce) could be effectively implemented. Additionally, a scalability analysis was conducted by increasing the number of computing nodes, thereby demonstrating the performance of the parallel FCM application.

Fernández and colleagues [17] analyzed the main recommendations on this topic by examining fuzzy model designs; they discussed the issues encountered with existing algorithms, as well as data distribution, parallel processing, and the representation of information in fuzzy form. Labib [18] examined both fuzzy and crisp classification techniques within the MapReduce framework; additionally, he evaluated the outputs of the proposed systems comparatively with the methods included in existing studies. Fuzzy techniques such as fuzzy *k*-nearest neighbor and precise techniques such as support vector machines and *k*-nearest neighbor have been used. In the processing of large-scale data, the MapReduce paradigm has been preferred. Wang and colleagues [19] focused on the use of fuzzy clustering techniques in big data processing problems and analyzed the advantages these techniques offer in detail. Additionally, they have addressed the current challenges encountered in the big data processing process and made some predictions accordingly. According to certain principles, they have stated that fuzzy clustering will transform into even more innovative and promising environments in big data processing applications. Fernández and colleagues [20] examined the relationship between data sparsity resulting from data sampling in MapReduce and the number of labels in fuzzy variables. In particular, they considered that as the number of partitions in the initial sample increases, the level of detail required to achieve good performance might also increase. Elkano and colleagues [21] aimed to design a new fuzzy rule-based classification system for large-scale data classification. It has been observed that the algorithm they developed shares certain fundamental similarities with the CHI-BD algorithm. Therefore, they have evaluated the suitability of the CHI-BD algorithm for MapReduce paradigms.

3 BIG DATA AND MAPREDUCE

Nowadays, new research methods aimed at improving the collection, analysis, and modeling of large-scale data are being discovered. However, the discovery and development of thoughts and approaches related to big data require long-term innovation and effort [19]. In this context, the MapReduce algorithm is a method that significantly contributes to the analysis and processing of big data [23]. Thanks to this algorithm, thoughts and approaches related to the big data era can be developed more quickly and effectively; valuable contributions are

being made to information-based and intelligent decision-making processes in various sectors.

3.1 Big Data

The volume and diversity of data worldwide are increasing at an unprecedented rate in human history. With the penetration of internet technologies and social media into every aspect of our lives, individuals now continuously generate data even during their daily activities. This large and multifaceted data flow obtained from various sources has led to the emergence of the concept of "Big Data". Big data is a new concept that encompasses heterogeneous data of varying volumes, which cannot be effectively processed by traditional data processing techniques, and consists of various forms of digital content [24].

Big Data is the totality of structured, semi-structured, and unstructured data produced with high volume, high velocity, and high variety. Therefore, the three fundamental elements that characterize the phenomenon of big data are variety, velocity, and volume. Some sources also emphasize additional elements such as accuracy (veracity) and value (value). These factors, while making the process of analyzing and evaluating big data challenging, can be leveraged to harness the potential of big data through appropriate approaches and analytical methods [25].

3.2 MapReduce Programming Model

MapReduce is a distributed programming model developed by Google in 2004 for processing large datasets. This model is designed for writing large-scale, scalable, and fault-tolerant data applications. The MapReduce model is based on two fundamental operations: the map function and the reduce function. In the first stage, the input data is processed by the map function to create intermediate results. Then, these intermediate results are combined using the reduce function to obtain the final output [23]. The MapReduce system, which operates as a parallel and distributed computing model, processes in five steps.

The "Map" phase in the MapReduce architecture represents the processors that enable the parallel processing of data. Each processor is assigned an input key specified as K1, and all input data associated with this key is transmitted. The Map code provided by the user is executed exactly once for each $K1$ key, and its output is produced by organizing it according to the $K2$ key. The "Reduce" component consists of reduction processors; each processor is assigned a $K2$ key, and all data from the Map phase associated with this key is transferred to this processor. The Reduce code provided by the user is executed exactly once for each $K2$ key produced by the Map phase. As a result, the MapReduce system collects all the Reduce outputs, sorts them by the $K2$ key, and thus obtains the final output [26]. Fig. 1 illustrates a typical MapReduce programming model, exemplifying the mapping and reducing steps.

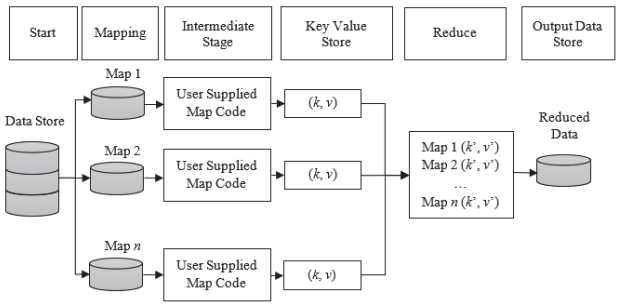

Figure 1 The MapReduce programming model

4 FUZZY SYSTEM MODELING

In the mid-1960s, Zadeh's pioneering work on fuzzy sets introduced an approach that replaced probabilistic uncertainty, leading to the dawn of a new era in the field of knowledge discovery. Later, fuzzy sets and fuzzy logic began to be used in system modeling applications.

4.1 Fuzzy System Modeling with Fuzzy Rule-Based

The prominent approach in fuzzy system modeling is fuzzy rule-based system modeling, which attempts to express the relationship between the inputs and outputs of a system through fuzzy sets. The fuzzy rule-based model expresses this relationship using rules in the form of "IF... THEN" [4]. Zadeh [27] defined the fuzzy rule-based model in a general expression as shown in Eq. (1).

$$R{:}\,\mathrm{ALSO}_{i=1}^{K}\left(\mathrm{IFantecedent}_i\,\mathrm{THENconsequent}_i\right) \tag{1}$$

Tagaki-Sugeno [6] represents a special case of the fuzzy rule-based system proposed by Zadeh [27]. In this approach, as expressed in Eq. (2), the right side of the rule is represented as a regression equation, while the left side is shown as a fuzzy set.

$$R{:}\,\mathrm{ALSO}_{i=1}^{K}\left(\mathrm{IFantecedent}_i\,\mathrm{THEN}y_i = a_i x^T + b_i\right) \tag{2}$$

4.2 The System Modeling Based on Fuzzy Function

Grinder and Bandler [28] were the first researchers to introduce fuzzy functions as a structure that provides connection or overlap between our sensory-based representation systems. Sasaki [29], Demirci [30], and Demirci with Recasens [31] have conducted studies on the mathematical theory and fundamental properties of fuzzy functions and have formulated the structures of fuzzy functions along with input variables that include membership values.

Hathaway and Bezdek [32] proposed the Fuzzy C-Regression (FCR) model within fuzzy rule-based system models, reducing dependence on expert intervention and creating self-learning systems. First, fuzzy clusters were determined using the FCM method; then, an FCR equation was created for each cluster with the help of the FCR algorithm.

Höppner and Klawonn [33] developed an integrated clustering structure by combining the FCM and FCR algorithms into a single clustering scheme. To prevent the

effect of harmonics, they developed a nonlinear regression model by updating the FCM clustering algorithm.

The fuzzy function approach, proposed by Türkşen [8] as an alternative to traditional fuzzy rule-based inference systems, has been further developed by Çelikyılmaz and Türkşen [34-37]. Unlike the models proposed by Hathaway and Bezdek [32] and Höppner and Klawonn [33], this new approach significantly improves the prediction performance of fuzzy functions by incorporating not only the original input variables but also the membership values obtained from each set and their appropriate transformations into the model. Thus, the approach has been distinctly differentiated from existing models both structurally and functionally.

In this study, the structures of fuzzy functions were created using the Classical FCM algorithm and the IFCM method developed by Çelikyılmaz and Türkşen [34-37] with Least Squares Estimation (LSE) and Support Vector Machines (SVM).

4.2.1 Standard Fuzzy C-Means Algorithm

Data representing a system can be defined as a set of observations in the form of (X_k, Y_k), k = 1, …, n. Here, Xk represents the input variables, while Y_k represents the output variables:

$$X_k = \left(x_{jk} \mid j = 1, ..., p; k = 1, ..., n \right) \tag{3}$$

In Eq. (3), x_{jk} represents the measurement value of the j-th input variable in the k-th example. The standard FCM algorithm clusters the data fuzzily by minimizing the following objective function:

$$\begin{aligned} &\min = J(U, V) = \sum_{k=1}^{n}\sum_{i=1}^{c} (u_{ik})^m \left(\|X_k - v_i\| \right)_A , \\ &\text{s.t.} \\ &0 \le u_{ik} \le 1, \ \forall i, k, \\ &\sum_{i=1}^{c} u_{ik} = 1, \ \forall k, \\ &0 \le \sum_{k=1}^{n} u_{ik} \le nd, \ \forall i, \end{aligned} \tag{4}$$

here: U - Membership matrix (u_{ik} - degree of membership of the k. data point in the i-th cluster), V - Set of cluster centers, m : Fuzziness exponent ($m > 1$), v_i - center of the i-th cluster, $\| \cdot \| A$ - Extended Euclidean norm.

The constraints are as follows: $0 \le u_{ik} \le 1$, $\forall i, k$; $i = \sum cu_{ik} = 1$, $\forall k$; $0 < k = \sum nu_{ik} < nd$, $\forall i$.

These constraints ensure that the sum of the membership values of each data point distributed across all clusters equals 1, and that no cluster is empty or contains too much data.

For the determined optimal m* and c*, the cluster centers are obtained as follows using the FCM algorithm:

$$\underbrace{v_{X|Y_j}}_{m^*} = \left(x_{1j}^c, \ x_{2j}^c, ..., x_{pj}^c, y_j^c \right) \tag{5}$$

$$\underbrace{v_{X_j}}_{m^*} = \left(x_{1j}^c, \ x_{2j}^c, ..., x_{pj}^c \right) \tag{6}$$

Eq. (5) represents the cluster center in the common space that includes both input and output variables, while Eq. (6) represents the center calculated only based on the input variables.

$$\begin{aligned} &u_{ik} = \left(\sum_{j=1}^{c^*} \left(\frac{X_k - v_{X,i}}{X_k - v_{X,j}} \right)^{\frac{2}{m-1}} \right)^{-1}, \\ &\mu_{ik} = \{ u_{ik} \ge \alpha \} \end{aligned} \tag{7}$$

Eq. (7) is used to calculate the optimal membership values, and the unwanted harmonic components are filtered using the α-level cut method, while the membership values are corrected using Eq. (8):

$$\psi_{ij}\left(x_j\right) = \frac{\mu_{ij}\left(X_j\right)}{\sum_{i'=1}^{c} \mu_{i'j}\left(X_j\right)} \tag{8}$$

This process reduces noise in the data, allowing for more meaningful and highly predictive membership values to be obtained.

4.2.2 Improved Fuzzy C-Means Algorithm

The IFCM algorithm uses the following new objective function, unlike the traditional FCM algorithm:

$$\begin{aligned} &\min J_m^{IFCM} = \sum_{i=1}^{c}\sum_{k=1}^{n} \left(U_{ik}^{imp} \right)^m d_{ik}^2 + \\ &+ \sum_{i=1}^{c}\sum_{k=1}^{n} \left(U_{ik}^{imp} \right)^m \left(y_k - h_i\left(\tau_{ik}, \hat{w}_i\right) \right)^2 \end{aligned} \tag{9}$$

here: U_{ik}^{imp} - improved membership degree, d_{ik}^2 - the distance between the k. data point and the center of the i-th cluster, defined as $d^2 = ||x_k y_k - u_i(xy)||^2$, which measures the positional accuracy of each input-output data vector. y_k - output value of the k-th data, h_i - fuzzy function modeled for the i-th cluster, τ_{ik} - parameter containing additional information such as time delay or system state, $\hat{w}_i$ - estimated model parameters for the i-th cluster.

The main purpose of this objective function is to learn the appropriate clusters and membership values by minimizing the error in the fuzzy functions that model both the relationship of the data with the clusters and the behavior of the system.

4.2.3 Fuzzy Functions with Least Squares Estimation (FF-LSE)

Let a dataset A be {(2, 5), (3, 7), (4, 9), (5, 11)}. This representation shows that each input (X) and output (Y) pair are together. The matrix consisting of the membership values of the i-th set of dataset A is obtained as follows:

$$\Gamma_i = \left(\psi_{ij} \mid i = 1, ..., c^*; j = 1, ..., p\right) \tag{10}$$

When applying Standard FCM to divide the dataset into two clusters, let the following membership matrix be obtained: $U = \begin{bmatrix} 0.9 & 0.8 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.8 & 0.9 \end{bmatrix}$; 1. vector of membership values of the set: Γ_1 = [0.9, 0.8, 0.2, 0.1]; 2. vector of membership values of the set: Γ_2 = [0.1, 0.2, 0.8, 0.9].

In Eq. (11), the possible augmented input matrix for the FF-LSE estimation of the fuzzy function structure definitions is defined.

$$X_{ij}^{'} = \left[1, \Gamma_i, X\right] = \begin{bmatrix} 1 & \psi_{i1} & x_{i1} \\ \vdots & \vdots & \vdots \\ 1 & \psi_{ip} & x_{ip} \end{bmatrix} \tag{11}$$

A separate augmentation input matrix $X_i^{'}$ is created for each cluster. For example, the augmentation matrix for the 1st cluster is as follows.

$$X_1^{'} = \begin{bmatrix} 1 & 0.9 & 2 \\ 1 & 0.8 & 3 \\ 1 & 0.2 & 4 \\ 1 & 0.1 & 5 \end{bmatrix}$$

The output vector is the same for both sets.

$$\mathrm{Y} = \begin{bmatrix} 5 \\ 7 \\ 9 \\ 11 \end{bmatrix}$$

Membership values, original input values, and their appropriate transformations are used as input for the FF-LSE-based estimation for each cluster. Eq. (12) represents the fuzzy function of a single-input and single-output model.

$$Y_i = \beta_{i0} + \beta_{i1}\Gamma_i + \beta_{i2}X_{ij} \tag{12}$$

The model shown in Eq. (12) includes only the membership value Γ_i and the input variable X_{ij}, and represents the i-th rule corresponding to the i-th interactive set in the three-dimensional attribute space [Y_i, Γ_i, X] formed by these magnitudes. The estimation of the parameters of this model is done as follows using the FF-LSE approach:

$$\beta_i^* = \left(X_{ij}^{'T} X_i^{'}\right)^{-1} \left(X_{ij}^{'T} Y_i\right) \tag{13}$$

Using Eq. (13), regression coefficients are found for each cluster. For example, the fuzzy function coefficients of the 1st Set are found as $\beta_1^* = \left[1.2, 3.5, 2.1\right]$ and $\beta_2^* = \left[0.5, -1.2, 2.3\right]$ as a result of matrix operations. Therefore, the fuzzy function of the first set is defined as Y_i = 1.2 + 3.5Γ_i + 2.1X. For the first observation:

$$Y_1^* = 1.2 + 3.5 \cdot 0.9 + 2.1 \cdot 2 \rightarrow Y_1^* = 8.55$$

$$Y_2^* = 0.5 - 1.2 \cdot 0.1 + 2.3 \cdot 2 \rightarrow Y_2^* = 4.98$$

The overall output value is obtained by weighting the predicted output values of each cluster according to their respective membership degrees, as shown in Eq. (14).

$$Y^* = \frac{\sum_{i=1}^{c^*} \psi_i Y_i^*}{\sum_{i=1}^{c^*} \psi_i} \tag{14}$$

Using Eq, (14), the overall output value is calculated using the predicted output values of both sets: $Y^* = \frac{\left(0.9 \cdot 8.55\right) + \left(0.1 \cdot 4.98\right)}{\left(0.9 + 0.1\right)} \rightarrow Y^* = 8.198$ is calculated.

4.2.4 Fuzzy Functions with Support Vector Machines (FF-SVM)

In the FF-SVM approach, which is a regression model based on fuzzy functions, the data is first divided into c^* clusters using the FCM algorithm, and the membership degree of each data point to each cluster is calculated. These membership values are expressed as the matrix Γ_i = (ψ_{ij}).

For each cluster, a new augmented input matrix $\Phi_i(X, \Gamma_i) = [1, \Gamma_i, X]$ is created. This matrix includes the constant term, the original input variables, and the membership values themselves or some transformations of them. The generated matrix is used to train local nonlinear fuzzy functions for each cluster using the SVR method.

In FF-SVM, the main goal is to create a special prediction function that takes into account the membership values for each cluster and to obtain a general output by combining these functions. Therefore, the standard SVR model

$$Y_i = \omega \cdot \phi\left(x\right) + b \tag{15}$$

the expression is fuzzified and integrated as follows within the FF-SVM framework:

$$Y_i = \omega_i \cdot \phi\left(\Phi_i\right) + b_i \tag{16}$$

here: Φ_i - augmented input matrix for the i-th cluster, ω_i - weight vector for the i-th cluster, b_i - bias term for the i-th cluster, $\phi(\Phi_i)$ - mapping function in high-dimensional space (with the help of the kernel function)

Thanks to this structure, the model makes predictions not only based on the original data features but also through local functions that are adapted according to the membership degree of each data point to the relevant cluster. Thus, a local fuzzy function is defined for each cluster.

The overall output of the model is obtained by weighting the predictions obtained from all clusters by their respective membership values:

$$Y^* = \hat{y}_k = \frac{\sum_{i=1}^{c^*} \psi_{ik} \hat{y}_{ik}}{\sum_{i=1}^{c^*} \psi_{ik}} \quad (17)$$

In conclusion, the FF-SVM method provides a stronger prediction performance by learning both the distribution of the data according to clusters and the nonlinear relationships specific to each cluster.

5 FFFCM-BIGDATA ALGORITHM: A MAPREDUCE DESIGN BASED ON FUZZY C-MEANS CLUSTERING FOR MODELING WITH FUZZY FUNCTIONS

This section presents the FFFCM-BigData algorithm developed to tackle big data clustering problems. Fig. 2 shows the process of creating fuzzy functions in FFFCM-BigData according to the MapReduce schema.

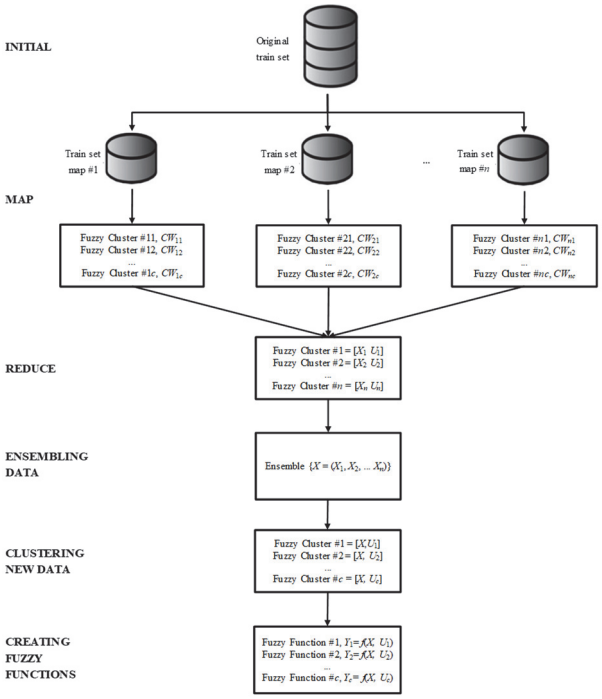

Figure 2 The procedure to build the fuzzy functions following a MapReduce scheme in the FFFCM-BigData algorithm

This procedure consists of the following phases:

Phase 1 (Initial): In the initial stage, each feature A_i is determined for the entire training dataset. Then, the original training dataset is automatically divided into independent data chunks sent to different processing units.

Phase 2 (Map): In the second stage, each processing unit independently processes the data it holds and clusters this data using the original FCM algorithm. Specifically, the optimal fuzziness level m^* and the number of clusters c^* pair have been determined through an iterative search process using standard FCM algorithms. Based on observational data, membership matrices were created for different numbers of clusters ($c \in \{2, ..., 10\}$) and degrees of fuzziness ($m \in \{1.3, ..., 3\}$) for FCM-based membership matrices have been created. These matrices were used to train linear regression (LSE) and support vector machine (SVM) models, and were evaluated on the training and test datasets using R^2, adjusted R^2, and RMSE metrics. The pair with the highest accuracy (c^*, m^*) was selected as the optimal combination for the proposed model.

Then, the clusters of training data segments were weighted using an intuitive method called the Penalized Certainty Factor [39]. At this stage, each distributed data segment is processed independently, and the representative cluster is determined by selecting only the cluster with the highest weight from each segment according to the weights obtained using the PCF method. In this way, each processing unit produces only the cluster with the locally strongest structure. Afterwards, a key is assigned to each processor, and the processing is done according to this key. The input elements of the set with the highest weight are produced as output according to the specified key.

$$CW_j = PCF_j = \frac{\sum_{x_p \in C_j} \mu_{A_j}(x_p) - \sum_{x_p \notin C_j} \mu_{A_j}(x_p)}{\sum_{j=1}^{n} \mu_{A_j}(x_p)} \quad (18)$$

here, $\mu_{Aj}(x_p)$ represents the membership degree of x_p to the j-th set, and C_j denotes the class determined by the j-th set. Algorithm 1 represents the Map function.

Algorithm 1. Map () Function

FunctionmapFCM (data, info, intermKVStore)
Input Data, max(c) = 10, max(m) = 3
Output c^*, m^*, V, U, max{$CW_j(X_i)$}
Start
For c 2, 3, …, max(c) # Try the number of clusters
For m 1.25, 1.50, 1.75, …, max(m) # Try the blur base

$$U_{ij} = \left[\sum_{k=1}^{c} \left(\frac{X_j - V_i}{X_j - V_k} \right)^{\frac{2}{m-1}} \right]^{-1}, \forall i, j$$

$$V_i = \frac{\sum_{j=1}^{n} \left(U_{ij}\right)^m X_j}{\sum_{j=1}^{n} \left(U_{ij}\right)^m}, \forall i$$

score = # It can be adjusted according to the highest R^2 or the lowest RMSE
if score < best_score:
best_score = score
$c^* = c$, $m^* = m$
End For
Max $1 \le k \le c$ $\{||V_k - V_{k-1}||^2\} > \varepsilon$
End For
The FCM algorithm is run according to optimum m^* and c^*, clustering is performed, and weighting is carried out.

$$CW_j = \frac{\sum_{X_i \in C_j} U_{ij} - \sum_{X_i \notin C_j} U_{ij}}{\sum_{j=1}^{n} U_{ij}}$$

$$\text{Value} = \max\left\{CW_j(X_i)\right\}$$

Add (weighted data, 'key', value)
End

Phase 3 (Reduce-Ensembling Data): In the fourth stage, the data produced in the Map stage and associated with a specific key are combined to obtain the final reduced dataset. Algorithm 2 represents the Reduce function.

Algorithm 2. Reduce () Function

Function reduceFCM (intermKey, intermValIter, outKVStore)
Input weighted data
Output Reduced Data
Start
Read (hasnext(intermValIter))
Reduced_Data ← getnext(intermValIter);
End
  Add (outKVStore, intermKey, Reduced_Data);
End

Phase 4 (Clustering New Data - Creating Fuzzy Function): In the fourth stage, the reduced data is clustered using the FCM algorithm with iterative searches for different numbers of clusters ($c \in \{2, ..., 10\}$) and degrees of fuzziness ($m \in \{1.3, ..., 3\}$) for the FCM algorithm, the optimum c^* and m* values were determined through an iterative search, and the reduced dataset was clustered using the FCM algorithm with the determined optimum parameters.

For each cluster of data clustered using the FCM and IFCM algorithms, linear and polynomial fuzzy functions were created using the LSE (FF-LSE) and SVM (FF-SVM) model types. Algorithm 3 demonstrates the modeling of the reduced dataset using fuzzy functions (FF-LSE, FF-SVM) with the FCM and IFCM algorithms.

Algorithm 3. Ensemble-Data-FF-LSE -SVM-Fuzzy Function ()

Function FF-LSE-SVM (Reduced_Data)
Input Max(c), Max(m), c_reg, ε, Function_Type, Model_Type
Output Y_i,
Start
Weight Matrix $W = [w_1, w_2, ..., w_n]$ / Predefined weights assigned to data points
For c = 1, 2, 3 …, Max(c) # Try the number of clusters
For m = 1.25, 1.50, 1.75, …, Max(m) # Try the blur base
If FCM_Type = FCM

$$\Gamma_i = \left(\psi_{ij} \mid i = 1, ..., c^*; j = 1, ..., \ p\right)$$

$$X_{ij}^{'} = \left[1, \Gamma_i, X\right]$$

$$U_{ij} = \left[\sum_{k=1}^{c}\left(\frac{X_j - V_i}{X_j - V_k}\right)^{\frac{2}{m-1}}\right]^{-1}, \forall i, j$$

$$V_i = \frac{\sum_{j=1}^{n}\left(U_{ij}\right)^m X_j}{\sum_{j=1}^{n}\left(U_{ij}\right)^m}, \ \forall i$$

score = # It can be adjusted according to the highest R^2 or the lowest RMSE
if score < best_score:
best_score = score
$c^* = c, m^* = m$
If FCM_Type = IFCM

$$\Gamma_i = \left(\psi_{ij}\right)$$

$$\Phi_i\left(X, \Gamma_i\right) = \left[1, \Gamma_i, X\right]$$

$$U_{ij}^{imp} = \left[\sum_{k=1}^{c}\left(\frac{X_j - V_i}{X_j - V_k}\right)^{\frac{2}{(m-1)}} \cdot w_j\right]^{-1}, \ \forall i, j$$

$$V_i = \frac{\sum_{j=1}^{n}\left(U_{ij}^{imp}\right)^m \cdot w_i \cdot X_j}{\sum_{j=1}^{n}\left(U_{ij}^{imp}\right)^m \cdot w_i}, \ \forall i$$

If score < best_rmset:
score = best_score
$c^* = c, m^* = m$
End For
Max $1 \le k \le c$ $\{\|V_k - V_{k-1}\|^2\} > \varepsilon$
End For
The algorithm is run according to the optimum m^* and c^*, and fuzzy functions are produced.
If Model_Type = LSE

$$Y_i^* = \beta_{i0}^* + \beta_{i1}^* \Gamma_i + \beta_{i2}^* X_{ij}$$

$$Y^* = \frac{\sum_{i=1}^{c^*} \psi_i Y_i^*}{\sum_{i=1}^{c^*} \psi_i}$$

End
If Model_Type = SVM

$$Y_i = \omega_i \cdot \phi\left(\Phi_i\right) + b_i$$

$$Y^* = \hat{y}_k = \frac{\sum_{i=1}^{c^*} \psi_{ik} \hat{y}_{ik}}{\sum_{i=1}^{c^*} \psi_{ik}}$$

End

6 EXPERIMENTAL ANALYSIS

In this section, the experimental analysis conducted on the datasets obtained from https://www.kaggle.com/datasets is presented. As shown in Tab. 1, these datasets are as follows:

- EPMD: Elevator Predictive Maintenance Dataset
- EMT: Electric Motor Temperature
- PG: Production Quality

The datasets summarized in Tab. 1 were divided into smaller parts and distributed to various processing units using the Map Function (Algorithm 1) within the MapReduce framework on the TRUBA infrastructure, employing the Adaptive Random Forest (ARF) method, which was set up at the MODSİMMER Data Center. Thanks to this approach, the scalability of large datasets has been increased and the processing time has been shortened.

Data segments were clustered according to the parameters and values specified in Tab. 2; subsequently, the parameter values providing the best performance were

determined. With these determined optimal parameter values, the data segments were re-clustered, and each cluster was weighted using the PCF method. The cluster with the highest weight has been selected as representative of the corresponding data piece.

These representative clusters obtained from each training data piece are combined (ensemble) to create a reduced dataset representing the entire dataset (Algorithm 2). Tab. 3 presents the reduced datasets obtained for each dataset.

The reduced datasets were re-clustered using the classical FCM algorithm, and the optimal c^* and m^* values were determined. Later, fuzzy functions were created for the FF-LSE and FF-SVM algorithms based on performance metrics (RMSE, R^2, and adjusted R^2) (Algorithm 3).

Table 1 Characteristics of the datasets used in the experiment

Data Sets	The Number of Observations	The Number of Variables	Training Data Sets	Validation Data Sets	Testing Data Sets
EPMD	112002	6	84002	28000	28000
EET	1330816	7	998112	332704	332704
PG	25302	18	18977	6325	6325

Table 2 The Parameters of models

c value	m value	c_reg	Epsilon
[2, 3, …, 9, 10]	[1.3, 1.5, …, 3.0]	[2^0, 2^1, …, 2^7]	[0.1, 0.2, …, 0.5]

Table 3 The reduced datasets obtained from the MapReduce algorithm

Data Sets	The Number of Observations	The Number of Variables	Training Data Sets	Validation Data Sets	Testing Data Sets
EPMD	23833	6	17874	5959	5959
EET	273651	7	205238	68413	68413
PG	3419	18	2565	854	854

Table 4 EPMD: elevator predictive maintenance dataset solution

Classical								
LSE			SVM					
R^2	Adjusted R^2	RMSE	R^2	Adjusted R^2	RMSE			
0.7291	0.7288	2.3587	0.7399	0.7397	2.3111			
FF-LSE								
Fuzzy Functions	FCM				IFCM			
	Optimum Parameters	R^2	Adjusted R^2	RMSE	Optimum Parameters	R^2	Adjusted R^2	RMSE
$f(u, x)$	$c^* = 6$; $m^* = 1.7$	0.7298	0.7296	2.3555	$c^* = 5$; $m^* = 2.0$	0.8084	0.8083	1.9834
$f(u, u^2, x)$	$c^* = 5$; $m^* = 2.3$	0.7285	0.7282	2.3613	$c^* = 9$; $m^* = 1.5$	0.8376	0.8375	1.8260
$f(u, e^u, x)$	$c^* = 9$; $m^* = 1.5$	0.7286	0.7284	2.3608	$c^* = 2$; $m^* = 1.5$	0.8280	0.8279	1.8793
FF-SVM								
$f(u, x)$	c_reg = 64; $\varepsilon = 0.1$; $c^* = 7$; $m^* = 1.3$	0.9183	0.9183	1.2949	c_reg = 128; $\varepsilon = 0.1$; $c^* = 4$; $m^* = 1.3$	0.9497	0.9496	1.0165
$f(u, u^2, x)$	c_reg = 64; $\varepsilon = 0.1$; $c^* = 7$; $m^* = 1.3$	0.9184	0.9183	1.2948	c_reg = 128; $\varepsilon = 0.1$; $c^* = 3$; $m^* = 1.5$	0.9559	0.9559	0.9517
$f(u, e^u, x)$	c_reg = 128; $\varepsilon = 0.1$; $c^* = 5$; $m^* = 1.7$	0.9185	0.9184	1.2938	c_reg = 128; $\varepsilon = 0.2$; $c^* = 5$; $m^* = 1.5$	0.9591	0.9590	0.9169

Table 5 EMT: electric motor temperature dataset solution

Classical								
LSE			SVM					
R^2	Adjusted R^2	RMSE	R^2	Adjusted R^2	RMSE			
0.8854	0.8851	8.12	0.8015	0.8011	10.6855			
FF-LSE								
Fuzzy Functions	FCM				IFCM			
	Optimum Parameters	R^2	Adjusted R^2	RMSE	Optimum Parameters	R^2	Adjusted R^2	RMSE
$f(u, x)$	$c^* = 9$, $m^* = 1.5$	0.886	0.886	8.0898	$c^* = 8$, $m^* = 1.7$	0.946	0.9461	5.565
$f(u, u^2, x)$	$c^* = 10$, $m^* = 1.5$	0.892	0.8918	7.8814	$c^* = 7$, $m^* = 1.5$	0.95	0.9494	5.387
$f(u, e^u, x)$	$c^* = 10$, $m^* = 1.5$	0.908	0.9077	7.2781	$c^* = 10$, $m^* = 1.5$	0.947	0.9465	5.542
FF-SVM								
$f(u, x)$	c_reg = 128; $\varepsilon = 0.3$; $c^* = 5$; $m^* = 1.5$	0.984	0.9839	3.0407	c_reg = 128; $\varepsilon = 0.4$; $c^* = 3$; $m^* = 1.5$	0.993	0.9933	1.955
$f(u, u^2, x)$	c_reg = 128; $\varepsilon = 0.4$; $c^* = 8$; $m^* = 1.5$	0.984	0.9839	3.0402	c_reg = 128; $\varepsilon = 0.3$; $c^* = 3$; $m^* = 2.7$	0.994	0.9936	1.92
$f(u, e^u, x)$	c_reg = 128; $\varepsilon = 0.3$; $c^* = 8$; $m^* = 2$	0.984	0.9842	3.0153	c_reg = 128; $\varepsilon = 0.3$; $c^* = 4$; $m^* = 2$	0.994	0.9941	1.838

Tab. 4 presents the performance results of the models developed using FF-LSE and FF-SVM fuzzy function types on the "EPMD: Elevator Predictive Maintenance Dataset". In the classical models, it has been observed that

the classical SVM model performs better than the classical LSE model. In the FF-LSE models, the model developed using the IFCM algorithm with the function structure "$f(u, u^2, x)$" has yielded better results compared to the other models. In contrast, among the FF-SVM models developed using the IFCM algorithm, the model with the function structure "$f(u, e^u, x)$" has shown better performance. Fig. 3 is a line graph showing the actual and predicted output values of the FF-SVM model developed using the IFCM algorithm and the "$f(u, e^u, x)$" function structure.

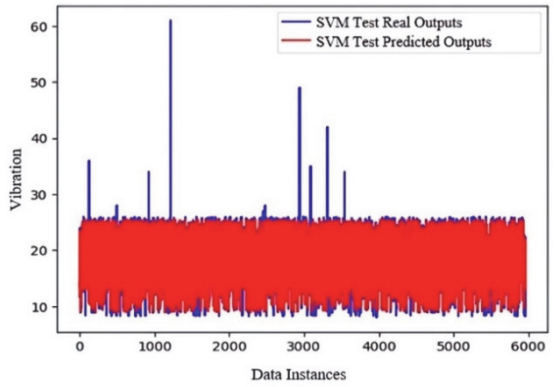

Figure 3 Graph of actual and predicted output values of the FF-SVM model developed based on the IFCM algorithm for the "EPMD: elevator predictive maintenance dataset"

Tab. 5 presents the performance results of the models developed using FF-LSE and FF-SVM fuzzy function types on the "EMT: Electric Motor Temperature Dataset". In the classical models, it has been observed that the classical LSE model performs better than the classical SVM model. In the FF-LSE models, it has been observed that among the models developed using the IFCM algorithm, the model with the function structure "$f(u, u^2, x)$" performed better. In the FF-SVM models, the model with the function structure "$f\,u, e^u, x)$" developed using the IFCM algorithm has yielded the best results among the models. Fig. 4 is a line graph showing the actual and predicted output values of the FF_SVM model developed using the IFCM algorithm and the "$f(u, e^u, x)$" function structure.

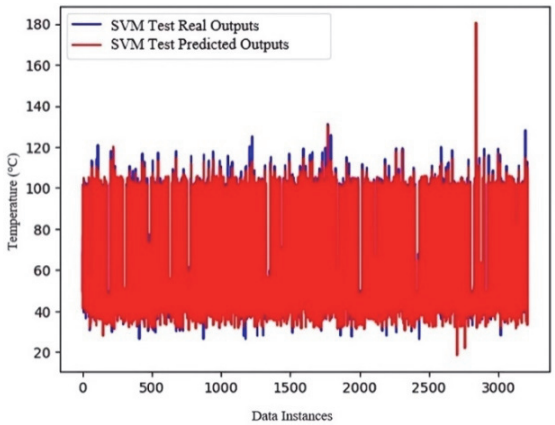

Figure 4 Graph of actual and predicted output values of the FF-SVM model developed based on the IFCM algorithm for the "EMT: electric motor temperature dataset"

Table 6 PG: production quality dataset solution

Classical								
LSE			SVM					
R^2	Adjusted R^2	RMSE	R^2	Adjusted R^2	RMSE			
0.7844	0.78	15.01	0.8051	0.8011	14.275			
FF-LSE								
Fuzzy Functions	FCM				IFCM			
	Optimum Parameters	R^2	Adjusted R^2	RMSE	Optimum Parameters	R^2	Adjusted R^2	RMSE
$f(u, x)$	$c^* = 5$, $m^* = 1.5$	0.787	0.7824	14.93	$c^* = 6$, $m^* = 1.3$	0.789	0.7842	14.87
$f(u, u^2, x)$	$c^* = 6$, $m^* = 1.5$	0.788	0.7834	14.896	$c^* = 6$, $m^* = 1.3$	0.788	0.7835	14.89
$f(u, e^u, x)$	$c^* = 4$, $m^* = 1.3$	0.787	0.7821	14.941	$c^* = 6$, $m^* = 1.3$	0.791	0.7865	14.79
FF-SVM								
$f(u, x)$	c_reg=2; $\varepsilon = 0.1$; $c^* = 7$; $m^* = 3$	0.804	0.8001	14.31	c_reg=2; $\varepsilon = 0.1$; $c^* = 6$; $m^* = 1.5$	0.809	0.8049	14.14
$f(u, u^2, x)$	c_reg=2; $\varepsilon = 0.1$; $c^* = 7$; $m^* = 3$	0.804	0.8001	14.31	c_reg=2; $\varepsilon = 0.1$; $c^* = 2$; $m^* = 1.3$	0.809	0.805	14.14
$f(u, e^u, x)$	c_reg=2; $\varepsilon = 0.1$; $c^* = 5$; $m^* = 2.3$	0.805	0.8007	14.291	c_reg=2; $\varepsilon = 0.1$; $c^* = 8$; $m^* = 1.5$	0.811	0.807	14.06

Tab. 6 presents the performance results of the models developed using FF-LSE and FF-SVM fuzzy function types on the "PG: Production Quality Dataset". In classical models, it has been observed that the classical SVM model performs better than the classical LSE model. In FF-LSE models, it has been observed that among the models developed using the IFCM algorithm, the model with the function structure "$f(u, e^u, x)$" exhibited better performance. In the FF-SVM models, among the models developed using the IFCM algorithm, the model with the function structure "$f(u, e^u, x)$" has yielded the best results. Fig. 5 is a line graph showing the actual and predicted output values of the FF-SVM model developed using the IFCM algorithm and the "$f(u, e^u, x)$" function structure.

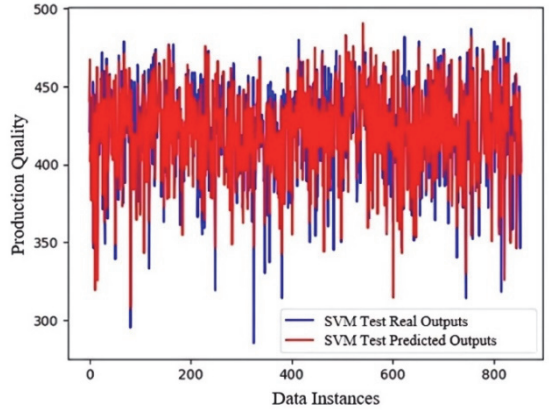

Figure 5 Graph of actual and predicted output values of the FF-SVM model developed based on the IFCM algorithm for the "PG: production quality dataset"

7 CONCLUSION

When big data is evaluated effectively, data-driven decisions can be made and positive developments can be achieved in relevant areas. However, in order for data scientists and analysts to obtain consistent and meaningful results from big data analytics, they need to have detailed knowledge of the existing data structure. Due to the volumetric density, high processing speed, and structural complexity of big data, analyzing such data with traditional methods is quite difficult.

The need to transition to scalable parallel architectures has emerged for existing clustering algorithms to work effectively on big data. In response to this need, a parallelized structure was created on big data using the MapReduce framework, thereby reducing processing time and increasing system scalability.

In the study, the performance of the proposed FFFCM-BigData algorithm is demonstrated through the parallel implementations of the original FCM and IFCM algorithms via the MapReduce paradigm on the ARF high-performance computing infrastructure. FCM and IFCM-based big data clustering algorithms perform better than classical clustering algorithms. However, the implementation of these methods requires more computational power and consists of more technically complex structures.

In order to evaluate the validity and effectiveness of the proposed FFFCM-BigData algorithm, comprehensive comparisons were made in your study with fuzzy clustering approaches based on FCM and IFCM, as well as different fuzzy function models such as FF-LSE and FF-SVM; the results obtained from these methods supported the validity of the proposed model.

In the experimental results, comparisons between the FF-LSE and FF-SVM models showed that models with an extended feature space matrix based on membership values and transformations yielded better results than models with an extended linear prediction matrix based on membership values and original input variables. However, it has been observed that these results could be further improved by including various transformations of the membership values in the augmented matrix. FF-SVM models are more powerful than classical SVM models because they have the capacity to operate without directly using membership values as predictors.

As a result, the FFFCM-BigData algorithm proposed in this study, along with FCM and IFCM-based fuzzy function models, has been effectively parallelized within the MapReduce paradigm and successfully applied in big data environments. In commonly used approaches in the literature, a separate model is created for each sub-dataset as a result of parallel processing, and the final results are obtained by statistically combining the outputs of these models. In this study, a single global model representing the entire system has been created. With the FFFCM-BigData algorithm, the model's generalization capability has been preserved, providing a faster, more efficient, and scalable solution for big data analysis. The main limitation of this study is that the datasets used are mostly from engineering fields, which may restrict the generalizability of the results to different sectors. In future studies, to increase the generalizability and application diversity of the proposed method, comprehensive applications can be conducted by selecting datasets from different sectors such as healthcare and finance.

8 REFERENCES

[1] Bataineh, K. M., Naji, M., & Saqer, M. (2011). A Comparison Study between Various Fuzzy Clustering Algorithms. *Jordan Journal of Mechanical & Industrial Engineering*, *5*(4), 335-343.

[2] Ludwig, S. A. (2015). MapReduce-based fuzzy c-means clustering algorithm: Implementation and scalability. *International Journal of Machine Learning and Cybernetics*, *6*(6), 923-934. https://doi.org/10.1007/s13042-015-0367-0

[3] Türkşen, İ. B. (2010). Bulanık Sistem Modellerinin Gelişimi. *İstatistik Araştırma Dergisi*, *7*(2), 11-24.

[4] Boudjerida, F., Akhtar, Z., Lahoulou, A., & Chettibi, S. (2024). Integrating fuzzy C-means clustering and fuzzy inference system for audiovisual quality of experience. *International Journal of Information Technology*, *16*(4), 2549-2562. https://doi.org/10.1007/s41870-023-01562-7

[5] Sugeno, M. & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, *1*(1), 7-31. https://doi.org/10.1109/TFUZZ.1993.390281

[6] Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, *15*(1), 116-132. https://doi.org/10.1109/TSMC.1985.6313399

[7] Ouellet, V., Mocq, J., El Adlouni, S., & Krause, S. (2021). Improve performance and robustness of knowledge-based fuzzy logic habitat models. *Environmental Modelling & Software*, 144, 105138. https://doi.org/10.1016/j.envsoft.2021.105138

[8] Türkşen, B. (2008). Fuzzy functions with LSE. *Applied Soft Computing*, *8*(3), 1178-1188. https://doi.org/10.1016/j.asoc.2007.12.004

[9] Göleç, A. M., Tokat, E., & Türkşen, I. B. (2012). Forecasting model of Shanghai and CRB commodity indexes. *Expert Systems with Applications*, *39*(10), 9275-9281. https://doi.org/10.1016/j.eswa.2012.02.077

[10] Morán, J., Bertolino, A., De La Riva, C., & Tuya, J. (2024). Automatic debugging of design faults in MapReduce applications. *IEEE Transactions on Software Engineering*, *50*(4), 1234-1247. https://doi.org/10.1109/TSE.2024.336976

[11] Wasikowski, M. A. (2010). Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1388-1400. https://doi.org/10.1109/TKDE.2009.187

[12] Del Río, S., López, V., Benítez, J. M., & Herrera, F. (2015). A MapReduce approach to address big data classification problems based on the fusion of linguistic fuzzy rules. *International Journal of Computational Intelligence Systems*, *8*(3), 422-437. https://doi.org/10.1080/18756891.2015.1017377

[13] Havens, T. C., Bezdek, J. C., Leckie, C., Hall, L. O., & Palaniswami, M. (2012). Fuzzy c-means algorithms for very large data. *IEEE Transactions on Fuzzy Systems*, *20*(6), 1130-1146. https://doi.org/10.1109/TFUZZ.2012.2201485

[14] Ben Ayed, A., Ben Halima, M., & Alimi, A. M. (2014). Survey on clustering methods: Towards fuzzy clustering for big data. *2014 6th International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, 79-84. https://doi.org/10.1109/SOCPAR.2014.7008028

[15] López, V., Del Río, S., Benítez, J. M., & Herrera, F. (2015). Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258, 5-38. https://doi.org/10.1016/j.fss.2014.01.015

[16] Azar, T. & Hassanien, A. E. (2015). Dimensionality reduction of medical big data using neural-fuzzy classifier. *Soft Computing*, *19*(4), 1115-1127. https://doi.org/10.1007/s00500-014-1327-4

[17] Fernández, A., Carmona, C. J., del Jesus, M. J., & Herrera, F. (2016). A view on fuzzy systems for big data: Progress and opportunities. *International Journal of Computational Intelligence Systems*, *9*(1), 69-80. https://doi.org/10.1080/18756891.2016.1180820

[18] Labib, S. S. (2016). A comparative study to classify big data using fuzzy techniques. *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, 1-5. https://doi.org/10.1109/ICEDSA.2016.7818508

[19] Wang, H., Xu, Z., & Pedrycz, W. (2017). An overview on the roles of fuzzy set techniques in big data processing: Trends, challenges and opportunities. *Knowledge-Based Systems*, *118*, 15-30. https://doi.org/10.1016/j.knosys.2016.11.008

[20] Fernández, A., del Río, S., Bawakid, A., & Herrera, F. (2017). Fuzzy rule based classification systems for big data with MapReduce: Granularity analysis. *Advances in Data Analysis and Classification*, *11*(4), 711-730. https://doi.org/10.1007/s11634-016-0260-z

[21] Elkano, M., Galar, M., Sanz, J., & Bustince, H. (2018). CHI-BD: A fuzzy rule-based classification system for Big Data classification problems. *Fuzzy Sets and Systems*, *348*, 75-101. https://doi.org/10.1016/j.fss.2017.07.003

[22] Chi, Z., Yan, H., & Pham, T. (1996). Fuzzy algorithms: With applications to image processing and pattern recognition. *World Scientific*, *10*, 225. https://doi.org/10.1142/3132

[23] Dahiphale, D. (2023). MapReduce for graphs processing: New big data algorithm for 2-edge connected components and future ideas. *IEEE Access*, *11*, 54986-55001. https://doi.org/10.1109/ACCESS.2023.3281266

[24] Gahi, Y., Mouftah, H. T., & Guennoun, M. (2016). Big Data Analytics: Security and Privacy Challenges. *Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC)*, 1016-1021. https://doi.org/10.1109/ISCC.2016.7543859

[25] Aktan, E. (2018). Büyük Veri: Uygulama Alanları, Analitiği ve Güvenlik Boyutu. *Bilgi Yönetimi*, *1*(1), 1-22. https://doi.org/10.33721/by.403010

[26] Hashem, I. A. T., Anuar, N. B., Gani, A., Yaqoob, I., Xia, F., & Khan, S. U. (2016). MapReduce: Review and open challenges. *Scientometrics*, *109*(1), 389-422. https://doi.org/10.1007/s11192-016-1945-y

[27] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-Part I. *Information Sciences*, *8*(3), 199-249. https://doi.org/10.1016/0020-0255(75)90036-5

[28] Grinder, J. & Bandler, R. (1976). *Patterns of the hypnotic techniques of Milton H. Erickson, M.D. (Vol. 1)*. Meta Publications.

[29] Sasaki, M. (1993). Fuzzy functions. *Fuzzy Sets and Systems*, *55*(3), 295-301. https://doi.org/10.1016/0165-0114(93)90255-G

[30] Demirci, M. (1999). Fuzzy functions and their fundamental properties. *Fuzzy Sets and Systems*, *106*(2), 239-246. https://doi.org/10.1016/S0165-0114(97)00280-7

[31] Demirci, M. & Recasens, J. (2004). Fuzzy groups, fuzzy functions and fuzzy equivalence relations. *Fuzzy Sets and Systems*, *144*(3), 441-458. https://doi.org/10.1016/S0165-0114(03)00301-4

[32] Hathaway, R. J. & Bezdek, J. C. (1993). Switching regression models and fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, *1*(3), 195-204. https://doi.org/10.1109/91.236552

[33] Hoppner, F. & Klawonn, F. (2003). A contribution to convergence theory of fuzzy c-means and derivatives. *IEEE Transactions on Fuzzy Systems*, *11*(5), 682-694. https://doi.org/10.1109/TFUZZ.2003.817858

[34] Celikyilmaz, F. (2005). Fuzzy functions with support vector machines. *Information Sciences*, 5163-5177. https://doi.org/10.1016/j.ins.2007.06.022

[35] Çelikyılmaz, A. & Türkşen, I. B. (2008). Enhanced fuzzy system models with improved fuzzy clustering algorithm. *IEEE Transactions on Fuzzy Systems*, *16*(3), 779-794. https://doi.org/10.1109/TFUZZ.2007.905919

[36] Çelikyılmaz, A. & Türkşen, I. B. (2008). Uncertainty modeling of improved fuzzy functions with evolutionary systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(4), 1098-1110. https://doi.org/10.1109/TSMCB.2008.924587

[37] Celikyilmaz, A. & Türksen, I. B. (2009). *Modeling uncertainty with improved fuzzy functions*. Modeling uncertainty with fuzzy logic: With recent theory and applications, Springer. https://doi.org/10.1007/978-3-540-89924-2_5

[38] Brereton, R. G. & Lloyd, G. R. (2010). Support vector machines for classification and regression. *Analyst*, *135*(2), 230-267. https://doi.org/10.1039/B918972F

[39] Ishibuchi, H. & Yamamoto, T. (2005). Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, *13*(4), 428-435. https://doi.org/10.1109/TFUZZ.2004.841738

Contact information:

Ahmet ARTUT
(Corresponding author)
1) Cumhuriyet University, Department of Computer Technologies,
Gürün Vocational School, Sivas Cumhuriyet University, Sivas, Türkiye
2) Erciyes University, Graduate School of Natural and Applied Sciences, Department of Industrial Engineering, 38030 Kayseri, Türkiye
E-mail: aartut@cumhuriyet.edu.tr

Adem GÖLEÇ
Erciyes University, Department of Industrial Engineering,
Erciyes University, Kayseri, Türkiye
E-mail: ademgolec@erciyes.edu.tr