

Chitrakant Banchhor\* and N. Srinivasu

# FCNB: Fuzzy Correlative Naive Bayes Classifier with MapReduce Framework for Big Data Classification

<https://doi.org/10.1515/jisys-2018-0020>

Received December 6, 2017.

**Abstract:** The term “big data” means a large amount of data, and big data management refers to the efficient handling, organization, or use of large volumes of structured and unstructured data belonging to an organization. Due to the gradual availability of plenty of raw data, the knowledge extraction process from big data is a very difficult task for most of the classical data mining and machine learning tools. In a previous paper, the correlative naive Bayes (CNB) classifier was developed for big data classification. This work incorporates the fuzzy theory along with the CNB classifier to develop the fuzzy CNB (FCNB) classifier. The proposed FCNB classifier solves the big data classification problem by using the MapReduce framework and thus achieves improved classification results. Initially, the database is converted to the probabilistic index table, in which data and attributes are presented in rows and columns, respectively. Then, the membership degree of the unique symbols present in each attribute of data is found. Finally, the proposed FCNB classifier finds the class of data based on training information. The simulation of the proposed FCNB classifier uses the localization and skin segmentation datasets for the purpose of experimentation. The results of the proposed FCNB classifier are analyzed based on the metrics, such as sensitivity, specificity, and accuracy, and compared with the various existing works.

**Keywords:** Big data, classification, correlative naive Bayes classifier, fuzzy theory, MapReduce.

## 1 Introduction

Data mining [6] has become a prevailing technique for the discovery of valuable information available on network platforms. Big data [6] significantly promotes the traditional industries to achieve better progressiveness, and hence data retrieval from the big data environment is necessary. The term “big data” is derived from the phrase “a large amount of data,” usually in zettabytes processed in a year. Hence, data management options should be openly available to each organization for better handling of big data [15, 22]. The data can be concluded as big data based on the factors of volume, velocity, variety, and veracity. Also, big data from Internet sources arrives in a continuous pattern, and thus the processing of data is more difficult [5].

Data mining schemes come under two major categories: clustering and classification. Various classifiers, such as support vector machine [13], naive Bayes (NB) [24], and extreme learning machine (ELM) [12] primarily contribute toward big data classification [9, 11]. The ELM [5] algorithm provides multiclassification of data rather than binary classification [6]. While processing data with high volume, the computational complexity of algorithms is increased [1]. The supervised classification approaches classify big data through a learning algorithm, and thus finds the suitable classes for the database [7]. The prevailing problem by the large size of data can be solved by introducing MapReduce schemes. Google introduces the MapReduce [1, 3, 14, 25, 29]

---

\*Corresponding author: Chitrakant Banchhor, Research Scholar, Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India, e-mail: banchhorchitrakant@gmail.com

N. Srinivasu: Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

framework for mining data of size larger than petabytes. MapReduce contains the mapper and reducer for the parallel processing of the datasets [4].

Big data include the collection of data from different fields, and the employment of the classification algorithm solves the data mining issues in big data. The main idea behind the classification task is to build a model (classifier) that classifies the data with the goal to accurately predict the target class for each item in the data [14]. There are many techniques, such as decision trees, Bayes networks, genetic algorithms, genetic programming, and so on, to comply with the classification of big data [1]. The properties, such as continuity and distributed blocks, present in the big data pose additional challenges to the ELM algorithms [5]. The big data also has imbalanced datasets and the fuzzy rule-based classification systems (FRBCS) [17, 18], denoted as Chi-FRBCS-Big Data CS, have achieved significant results during the classification of the imbalanced big data [18]. Literature work has also discussed the MapReduce-based fuzzy c-means clustering [19], k-nearest neighbor algorithm [20], fuzzy associative classifier [23], machine learning tools [28], and Chi-FRBCS-BigData algorithm [8] for big data classification.

The primary contribution of this research is the development of the fuzzy correlative NB classifier (FCNB) for big data classification. The proposed model permits the MapReduce framework to deal with the big data.

The paper organization is done in the following manner: Section 1 presents the introduction to the big data classification model. Section 2 presents the proposed FCNB algorithm along with the MapReduce framework for the big data classification. Section 3 presents various simulation results obtained for the proposed FCNB classifier based on the evaluation metrics. Section 4 concludes the research work.

## 2 Proposed Method: Proposed FCNB Classifier with the MapReduce Framework for Big Data Classification

This research work deals with big data classification with the proposed FCNB classifier. The proposed FCNB classifier is the extensive work of the correlative NB (CNB) classifier defined in Ref. [2]. The FCNB classifier is developed by integrating the CNB classifier and the fuzzy theory [9]. Also, this work includes the MapReduce framework for dealing with the big data. In data mining and the cloud environment, there is a continuous flow of data. The existing fuzzy NB (FNB) classifier has various merits, such as dealing with the missing attributes of the data sample, incremental learning, and performing the training with low data samples. In the proposed work, the FNB classifier is modified by adding the correlation between the data samples. This makes the proposed FCNB algorithm a dependent hypothesis. As the research allows the classification of big data, the inclusion of the MapReduce framework is necessary. The MapReduce framework eliminates the problem of the classification of a large dataset and the storage problems.

### 2.1 Algorithmic Description of the Proposed FCNB Classifier

The proposed FCNB classifier gets the training data from various sources as the input. The training data needs to be represented as the probability index table. The probability index table represents the data samples as the data matrix. The rows and columns of the probability index table represent the data and their respective attributes. The training sample for the proposed FCNB classifier is represented as follows:

$$T = \left\{ \begin{array}{l} T_{p,q}; 1 \leq p \leq d \\ 1 \leq q \leq a \end{array} \right\}, \quad (1)$$

where the term  $T_{p,q}$  represents the  $p^{\text{th}}$  data sample in the  $q^{\text{th}}$  attribute of the probability index table. The terms  $d$  and  $a$  represent the total data samples and the attributes present in the training dataset, respectively. The proposed model aims at classifying the data samples into various classes. Equation (2) expresses the classes indicated in the vector form:

$$G = \{g_p; 1 \leq p \leq d\}, \quad (2)$$

where the term  $g_p$  represents the class of the  $p^{\text{th}}$  data sample. The attributes present in the data sample contribute more toward the data classification. Consider the training data sample having  $a$  number of attributes; hence, the attributes of the data sample are represented as follows:

$$H = \{h_q; 1 \leq q \leq a\}, \quad (3)$$

where the term  $h_q$  represents the  $q^{\text{th}}$  attribute of the data sample. The data samples categorized under each attribute have unique symbols. The proposed FCNB classifier calculates the fuzzy membership degree depending on the unique data symbols within the attribute. Consider that there is  $S$  number of unique data symbols within the attribute. For the calculation of the membership degree of the proposed FCNB classifier, consider the  $q^{\text{th}}$  attribute in the training sample that contains  $s$  number of unique symbols. The symbols in the  $q^{\text{th}}$  attribute are indicated by  $h_q \in m^s$ , and the value of  $s$  varies in the range  $1 \leq s \leq S$ . The expression for the membership degree of training samples provided to the proposed FCNB classifier is represented by the following expression:

$$\mu_q^s = \frac{|m_q^s|}{d}, \quad (4)$$

where the term  $\mu_q^s$  shows the membership degree of the  $s^{\text{th}}$  symbol present in the  $q^{\text{th}}$  attribute of the training sample. Also, the term  $|m_q^s|$  represents the total occurrence of the  $s^{\text{th}}$  symbol in the  $q^{\text{th}}$  attribute and  $d$  indicates the data sample in the attribute. The proposed FCNB classifier classifies the data samples into  $K$  number of classes. The variation of the total number of classes is represented as  $G_k$ , and the value of  $k$  is in the range  $1 \leq k \leq K$ . The proposed FCNB classifier also calculates the membership degree of each class for the ground truth information. The membership degree for the  $k^{\text{th}}$  class provided with the ground truth information is represented as follows:

$$\mu_c^k = \frac{|m^k|}{d}, \quad (5)$$

where the term  $|m^k|$  represents the total occurrence of the  $k^{\text{th}}$  class in the ground truth information. The membership degree acts as a prime factor in the data classification. The model size of both the membership degrees derived in Eqs. (4) and (5) is expressed as  $[(a * S) + K]$ , where  $K$  is the total number of classes,  $S$  represents the number of unique data symbols, and  $a$  is the number of attributes.

### 2.1.1 Adapting the FNB Classifier with the Correlation Function

The existing FNB classifier utilizes the NB and the fuzzy-based approaches for the data classification. In this work, the proposed FCNB classifier adapts the FNB classifier with the virtual correlation function to make the proposed algorithm dependent on the hypothesis. Also, the correlation function makes the proposed algorithm an incremental learner. The proposed FCNB classifier finds the virtual correlation factor for each attribute present in the training database. Equation (6) expresses the virtual correlation between each attribute of the training data:

$$C^k = f(h_1, h_2, h_3, \dots, h_q, \dots, h_a), \quad (6)$$

where the term  $C^k$  represents the virtual correlation of the attributes in the  $k^{\text{th}}$  class. The term  $f(\cdot)$  represents the correlation function. The correlation function between the attributes of the data samples is constructed by representing the attributes and the symbols of the training sample as the diagonal matrix. Equation (7) represents the correlation function between the attributes of the training data:

$$f(h_1, h_2, h_3, \dots, h_q, \dots, h_a) = \frac{1}{1 + 2 + \dots + (a - 1)} \sum_{q=1}^a \sum_{s=q+1}^a r(h_e, h_q), \quad (7)$$

where the term  $r(h_e, h_q)$  represents the correlation between the  $e^{\text{th}}$  and the  $q^{\text{th}}$  attributes. The term  $1 + 2 + \dots + (a - 1)$  in Eq. (7) can be expressed as  $\frac{a(a-1)}{2}$  based on the triangular number series [10]. Now, Eq. (7) can be rewritten as

$$f(h_1, h_2, h_3, \dots, h_q, \dots, h_a) = \frac{1}{\frac{a(a-1)}{2}} \sum_{q=1}^a \sum_{s=q+1}^a r(h_s, h_q). \quad (8)$$

$$f(h_1, h_2, h_3, \dots, h_q, \dots, h_a) = \frac{2}{a(a-1)} \sum_{q=1}^a \sum_{s=q+1}^a r(h_s, h_q). \quad (9)$$

In this research work, the proposed FCNB classifier considers the correlation factor for finding the relation between the data samples present in the training data. The proposed FCNB classifier finds the correlation of the independent data sample present in the training set. The correlation factor for finding the relation between the unique symbols present in the attributes is represented as follows:

$$r(h_s, h_q) = \left[ \frac{\text{correlative}(h_s, h_q) + 1}{2} \right], \quad (10)$$

where the function  $\text{correlative}(h_s, h_q)$  indicates the Pearson's correlation coefficient [16]. The function  $\text{correlative}(h_s, h_q)$  finds the linear correlation between the data samples. The general expression for the Pearson's correlation coefficient is expressed by the following equation:

$$\text{correlative}(h_s, h_q) = \frac{\sum_{p=1}^d (t_{pq} - \bar{t}_q)(t_{ps} - \bar{t}_s)}{\sqrt{\sum_{p=1}^d (t_{pq} - \bar{t}_q)^2} \sqrt{\sum_{p=1}^d (t_{ps} - \bar{t}_s)^2}}, \quad (11)$$

where the term  $\bar{t}_q$  indicates the average of the data samples present in the  $q^{\text{th}}$  attribute and the term  $\bar{t}_s$  represents the average of the unique data symbols in the  $q^{\text{th}}$  attribute. The final output from the training of the proposed FCNB classifier contains the membership degree from the attribute, membership degree from the ground truth information, and the correlation factor. The output of the proposed FCNB classifier is expressed as follows:

$$\text{FCNB} = \{ \mu_q^k, \mu_c^k, C \}. \quad (12)$$

The membership degree for the attributes has the size of  $(d * S)$ , while the membership degree for the ground truth information has the size of  $(1 * K)$ . The correlation factor between the unique symbols of the attributes represented by each class has the size of  $(1 * K)$ . The results of the training of the proposed FCNB classifier have the total size of  $(d * S + 2K)$ .

## 2.2 Testing of the Proposed FCNB Classifier

This section presents the testing phase of the proposed FCNB classifier. The proposed FCNB algorithm utilizes the posterior probability of the NB classifier, the fuzzy membership degree, and the correlation function to classify the test data. For the training phase, the proposed FCNB classifier is provided with the test data represented as  $X$ . The proposed FCNB algorithm tries to classify the test data into  $K$  number of classes. The output of the proposed FCNB classifier is represented as follows:

$$G = \arg \text{Max}_{k=1 \text{ to } K} P(g_k|X) * C^k, \quad (13)$$

where the term  $P(g_k|X)$  defines the posterior probability based on the test data  $X$  for the class  $g_k$  and the term  $C^k$  represents the correlation for the class  $k$ . The value of  $P(g_k|X)$  is represented based on the following expression:

$$P(g_k|X) = P(g_k) \prod_{q=1}^a \left[ \frac{P(h_q|g_k)}{P(h_q)} * \mu_q \right], \quad (14)$$

where  $P(h_q|g_k)$  and  $P(h_q)$  represent the posterior probability for the attribute  $h_q$  based on the class  $k$  and the probability of occurrence of the attribute  $h_q$  in the class. The proposed FCNB model uses the Laplacian correction [2] in the above expression for avoiding the missing of the attributes during the training phase. The adjustment is done based on the following expressions:

$$P(X|g_k) = \frac{\left[ \sum_{X \in d} \mu_q^X \cdot \mu_X^k \right] + 1}{\left[ \sum_{X \in d} \mu_X^k \right] + |dom(h_q)|}, \quad (15)$$

$$P(g_k) = \frac{\left[ \sum_{X \in d} \mu_X^k \right] + 1}{d + |dom(G)|}, \quad (16)$$

where the term  $dom(G)$  represents the total number of classes and the term  $dom(h_q)$  represents the total number of data symbols present in the  $q^{\text{th}}$  attribute.

### 2.3 Pseudo Code of the Proposed FCNB Classifier

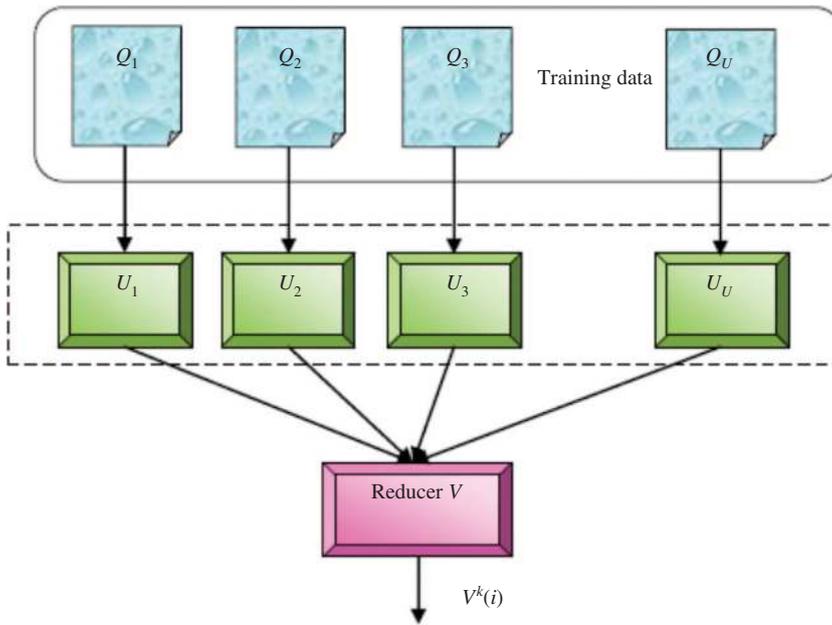
This section presents the pseudo code of the proposed FCNB classifier. As shown in Algorithm 1, the proposed FCNB classifier classifies the data into  $K$  number of classes. In the training phase, the proposed FCNB classifier gets the training data  $T$  as the input. For the training data, the membership degree and the correlative function are calculated. In the testing phase, the probability index of the test data is calculated and based on the classification output in Eq. (13), the class of the test data is found by the proposed FCNB classifier.

### 2.4 Adapting the FCNB Classifier in the MapReduce Framework

The application of the proposed FCNB classifier to the concept of the big data classification can be achieved by introducing the MapReduce framework in the proposed FCNB classifier. The MapReduce framework has the mapper and the reducer that allow the simultaneous functioning of the large dataset. This research performs big data classification through the training and testing phases of the proposed FCNB classifier.

**Algorithm 1:** Pseudo Code of the Proposed FCNB Classifier.

Algorithm: FCNB classifier	
Serial no.	<b>Steps</b>
1	<b>Input:</b> Data sample: $T$
2	<b>Output:</b> class: $G = \{G_1, G_2, \dots, G_K\}$
3	<b>Begin</b>
4	<b>// Training phase</b>
5	Read the training data $T$
6	<b>For</b> ( $p = 1$ to $d$ )
7	Read the data samples
8	<b>For</b> ( $q = 1$ to $a$ )
9	Read the attributes
10	Calculate the membership degree of the $q^{\text{th}}$ attribute
11	Calculate the membership degree of the ground value
12	Calculate the correlation factor $C^k$
13	<b>End for</b>
14	<b>End for</b>
15	<b>//Testing phase</b>
16	Read the test input $X$
17	Calculate the probability index of the class $P(g^k X)$
18	Find the class $G$ for the test data using Eq. (13)
19	<b>Return</b> the class $G$
20	<b>End</b>

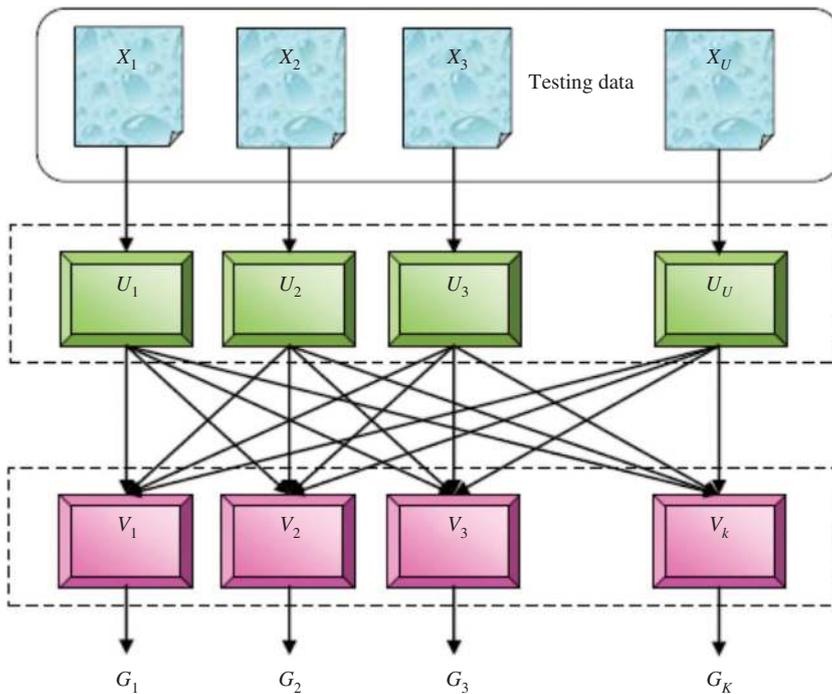


**Figure 1:** Training of the FCNB-Based MapReduce Framework.

In the training phase, the training data  $T$  is fed to the MapReduce function. Figure 1 presents the architecture of the proposed FCNB classifier enabled with the MapReduce for the training phase.

#### 2.4.1 Training Phase

**Training of the mapper:** The mapper present in Figure 2 gets the training data as the input. The training data is represented in the matrix, with the rows indicating the data and the columns indicating the attributes.



**Figure 2:** Testing of the FCNB-Based MapReduce Framework.

The training data to the mapper of the proposed FCNB classifier is represented as given in Eq. (17). As the data that arrives at the proposed classifier is a continuous data, the size of the data is very large. Hence, the data requires partitioning. In this work, the training data sample  $T$  is partitioned into  $U$  parts. Each part of the training sample is represented by the following expression:

$$T = \{Q_i; 1 \leq i \leq U\}, \quad (17)$$

where the term  $Q_i$  represents the  $i^{\text{th}}$  part of the data matrix. Each partitioned data is provided to the mapper of the proposed model. Hence, the number of mappers in the model equals the number of data sample parts. Consider that the proposed model has  $U$  mappers and  $V$  reducers. The data present in the mapper is represented by the following expression:

$$Q_i = \left\{ \begin{array}{l} n_{b,q}; 1 \leq b \leq A_i \\ 1 \leq q \leq a \end{array} \right\}, \quad (18)$$

where the term  $n_{b,q}$  represents the part of the data provided to the  $i^{\text{th}}$  mapper. The value of  $b$  varies based on the data present in the mapper  $A_i$ . The mappers present in the proposed classification model find the classes from the training data. Each mapper provides the data to the reducer of size  $(d * S) + 2K$ . The mapper generates the probability index table for the training sample, and it is represented as follows:

$$U_i = \{\mu_q(i), \mu_c(i), C(i), A_i\}, \quad (19)$$

where  $\mu_q(i)$  represents the membership degree of the  $q^{\text{th}}$  attribute for the data sample  $i$ ,  $\mu_c(i)$  shows the membership degree for the ground value,  $C(i)$  represents the correlation factor, and the term  $A_i$  indicates the number of data present in the mapper  $i$ .

**Training of the reducer:** For the training of the reducer, the outputs of each mapper are provided to the  $V$  reducers in the proposed model. The probability index table obtained from the mapper is of size  $(d * S) + 2K$ . The reducer tries to merge the probability index table from the mapper to a single function of size  $(d * S) + 2K$ . The reducer uses the aggregation mechanism to merge the outputs of the mapper. The membership degree present in the mapper output is reduced at the reducer phase based on the following expressions:

$$\mu_q^k = \frac{\sum_{i=1}^U \mu_q^k(i)}{U}, \quad (20)$$

$$\mu_c^k = \frac{\sum_{i=1}^U \mu_c^k(i)}{U}, \quad (21)$$

where the terms  $\mu_q^k$  and  $\mu_c^k$  represent the membership degrees of the attribute and the ground information of the data part  $i$ , respectively. The classified information from each mapper is merged in the reducer and is expressed as follows:

$$V^k = \frac{\sum_{i=1}^U V^k(i)}{U}, \quad (22)$$

where the term  $V^k(i)$  represents the classified output of the data part  $i$ .

#### 2.4.2 Testing Phase

The testing phase of the proposed FCNB classifier with the MapReduce is explained in this section. Figure 2 presents the MapReduce framework with the proposed FCNB classifier during the testing phase. For the testing, the test data  $X$  is provided to the MapReduce framework.

**Testing of the mapper:** The test data provided to the mapper is represented as  $X$ . Initially, the test data  $X$  is subjected for the partitioning and is expressed as follows:

$$X = \{X_x; 1 \leq x \leq U\}, \quad (23)$$

where the term  $X_x$  represents the  $x^{\text{th}}$  part of the test data  $X$ . The test data contains  $d$  number of data samples and  $a$  number of attributes. For the test data, the membership degree, the correlative function, and the number of data for each mapper are calculated. Finally, the mapper provides the information to the reducer.

**Testing of the reducer:** In the testing phase, the output of the mapper is fed to the reducer. The reducer merges the information and provides the information about the class variable of each part of the test data sample. The reducer provides  $K$  number of classes and is represented as follows:

$$G = \{G_1, G_2, \dots, G_K\}. \quad (24)$$

### 3 Results and Discussion

The simulation results achieved by the proposed FCNB classifier are presented in this section, which also contains the results of the comparative discussion achieved by analyzing the results of various comparative models.

#### 3.1 Experimental Setups

**Experimental setup 1:** The experimentation setup 1 contains a set of four mappers for analyzing the performance of different algorithms.

**Experimental setup 2:** The experimentation setup 2 contains a set of five mappers for the simulation purpose.

The entire experimentation is done on the Java platform installed in a personal computer with the following configurations: Windows 10 OS, 4 GB Ram, and Intel I3 processor.

#### 3.2 Dataset Description

The experimentation of the proposed FCNB classifier is done with the standard dataset localization dataset [16] and the skin segmentation dataset [26] utilized from the University of California, Irvine (UCI) machine repository, and their descriptions are provided as follows:

**Localization dataset** [16]: The localization dataset from the UCI machine repository contains information about the activities of the person, and is obtained through the observation of various persons wearing the data tag on their ankle, leg, etc. The localization dataset contains a large set of instances (total of 164,860) under eight attributes. Table 1 shows the description of the localization database.

**Table 1:** Description of Localization Dataset.

Dataset characteristics	Univariate, sequential, time series
Number of instances	164,860
Area	Life
Attribute characteristics	Real
Number of attributes	8
Associated tasks	Classification
Missing values	N/A

**Table 2:** Description of Skin Segmentation Dataset.

Dataset characteristics	Univariate
Number of instances	245,057
Area	Computer
Attribute characteristics	Real
Number of attributes	4
Associated tasks	Classification
Missing values	N/A

**Skin segmentation dataset** [26]: This dataset is collected from people of various age groups with different skin colors. The learning sample taken for the experimentation varies as 245,057, and they are grouped under four attributes. This dataset is more suitable for big data classification as the simulation results can be categorized under the skin set and the non-skin set. Table 2 shows the description of the skin segmentation dataset.

### 3.3 Comparative Models

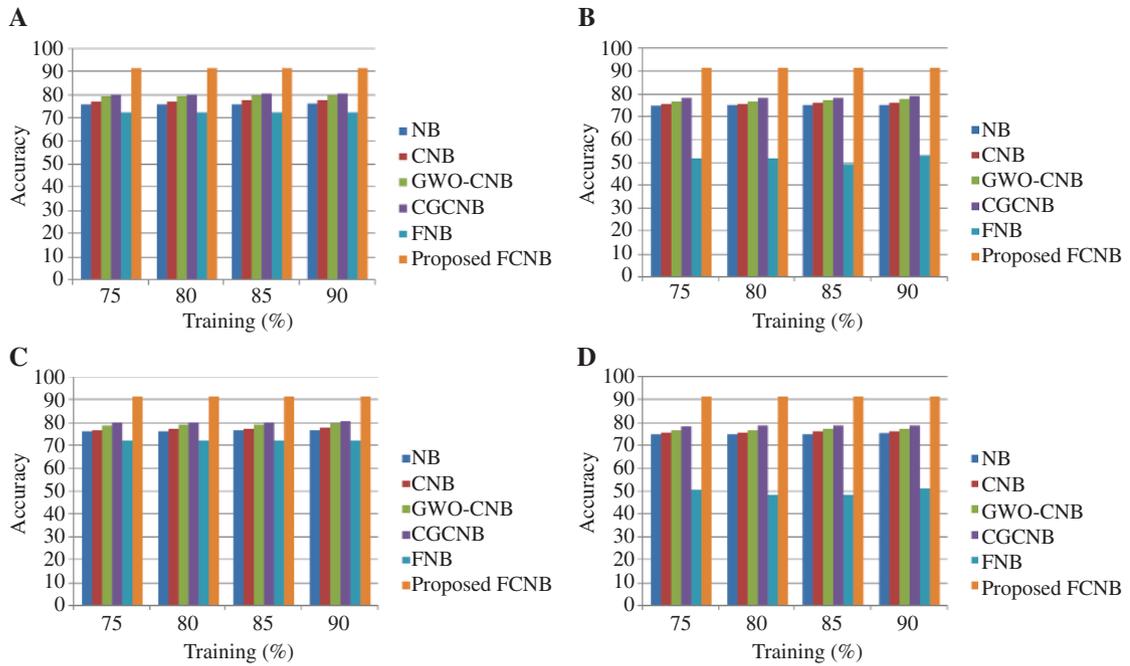
The performance of the proposed FCNB classifier for big data classification is compared with various methods, such as NB [27], CNB [2], gray wolf optimization-based CNB (GWO-CNB), cuckoo gray wolf-based CNB (CGCNB), and FNB classifier [27]. The NB classifier performs data classification by defining the probabilistic definition, and the CNB classifier uses the correlative function along with the NB for making the suitable decision. Incorporating the GWO [21] with the CNB leads to the formation of the GWO-CNB classifier, and the optimization scheme is used for defining the class. The CGCNB classifier is designed with the integration of the cuckoo search (CS) algorithm and the GWO algorithm with the CNB. The FNB classifier uses the fuzzy theory along with the NB for the classification purpose.

### 3.4 Comparative Analysis

Comparative analysis is done by varying the training percentage of the localization and the skin datasets for the various numbers of mappers, and the performance of each model is measured by the sensitivity, specificity, and accuracy.

#### 3.4.1 Analysis Based on Accuracy

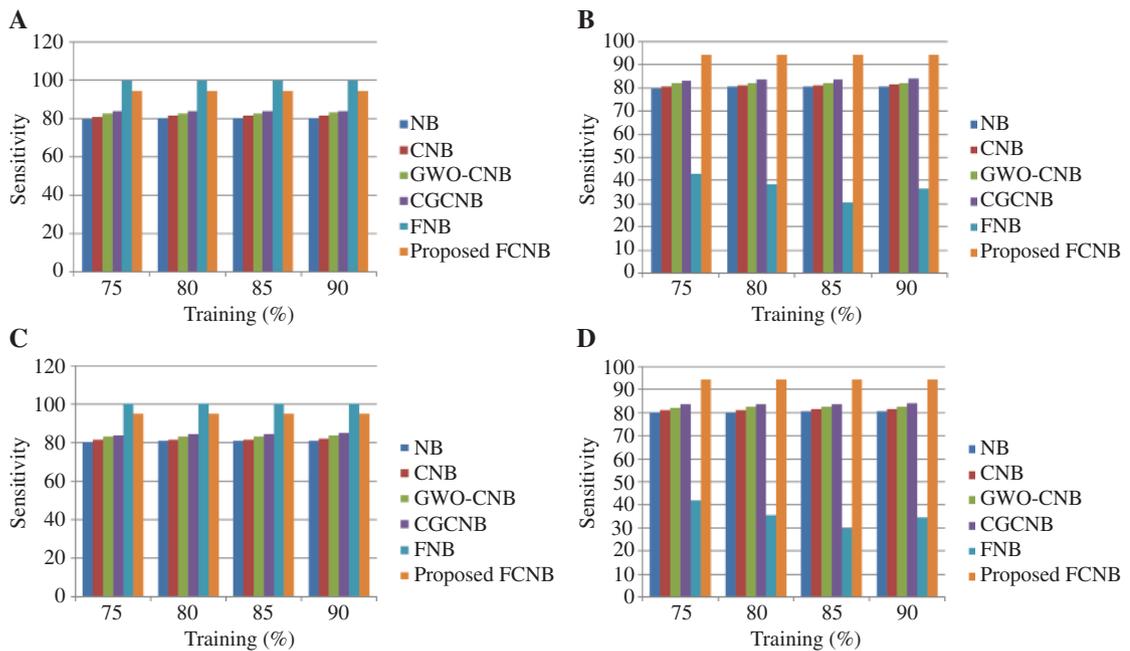
Figure 3 presents the comparative analysis of the proposed FCNB classifier based on the accuracy metric for the varying training percentages of the dataset and the mapper. Figure 3A presents the performance of the classifiers with the mapper size of 4 and varying training percentages of the localization dataset. For 90% training of the localization dataset, the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved accuracy values of 76.504%, 77.9505%, 79.862%, 80.8977%, and 72.33%, respectively, while the proposed FCNB classifier had an improved accuracy value of 91.7816%. Figure 3B presents the performance analysis of the classifiers in the skin dataset with mapper = 4. Here, the comparative models NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved the accuracy values of 75.723%, 76.636%, 77.770%, 79.327%, and 53.45%, respectively; however, the proposed FCNB classifier achieved the accuracy value of 91.7817%. Figure 3C and D present the performance analysis of the classifiers in the localization and skin datasets for mapper = 5. For 90% training of the localization dataset, the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers with five mappers achieved the accuracy values of 76.996%, 77.808%, 79.7638%, 80.6568%, and 72.3310%, respectively, while the proposed FCNB achieved the high accuracy value of 91.78164%. For the skin dataset, the proposed FCNB with mapper = 5 achieved an accuracy value of 91.78165%.



**Figure 3:** Comparative Analysis of the FCNB Classifier Based on Accuracy. For mapper = 4 in (A) localization dataset and (B) skin dataset. For mapper = 5 in (C) localization dataset and (D) skin dataset.

### 3.4.2 Analysis Based on Sensitivity

Figure 4 presents the comparative analysis of the proposed FCNB classifier based on the sensitivity metric for the varying training percentages of the dataset and the mapper. Figure 4A presents the analysis of the classifiers based on sensitivity with the mapper size of 4 and varying training percentages of the localization dataset. For 90% training of the localization dataset, the existing NB, CNB, GWO-CNB,

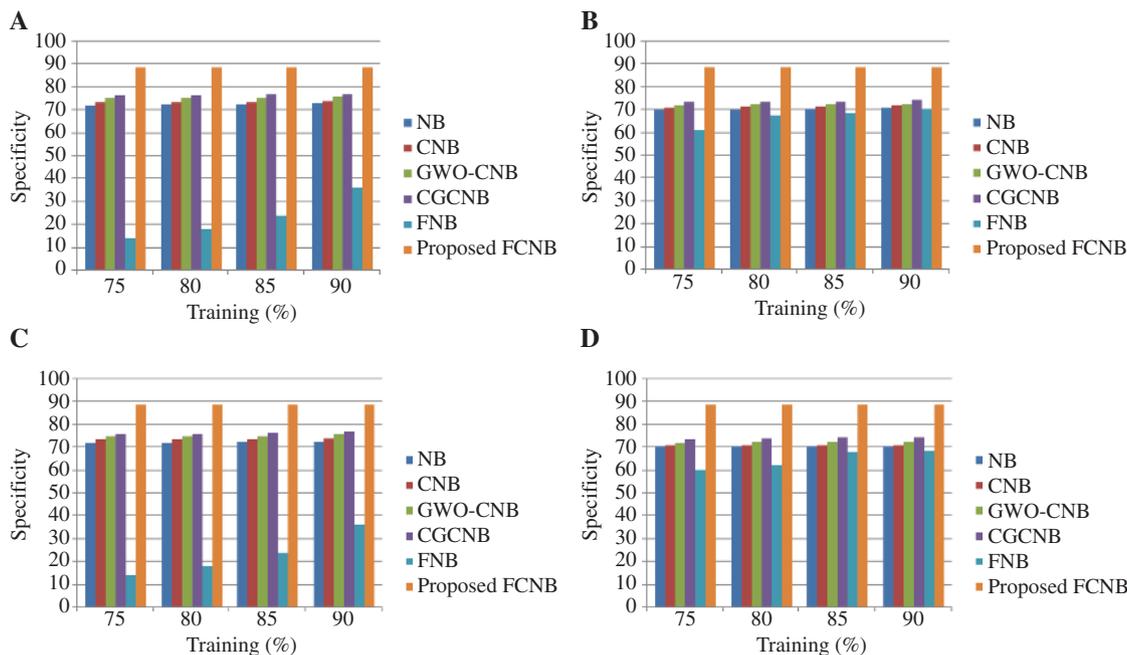


**Figure 4:** Comparative Analysis of the FCNB Classifier Based on Sensitivity. For mapper = 4 in (A) localization dataset and (B) skin dataset. For mapper = 5 in (C) localization dataset and (D) skin dataset.

CGCNB, and FNB classifiers achieved a sensitivity of 80.699%, 81.899%, 83.2474%, 84.399%, and 99.986%, respectively, while the proposed FCNB classifier achieved a sensitivity value of 94.79%, which was less than that of the FNB classifier. This is due to the factor that the training data taken for the classification is classified toward the same class. Figure 4B presents the performance of the classifiers in the skin dataset along with mapper = 4 based on sensitivity. Here, the comparative models NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved the sensitivity value of 80.845%, 81.845%, 82.466%, 84.2254%, and 36.811%, respectively; however, the proposed FCNB classifier achieved the sensitivity value of 94.79%. Figure 4C and D present the performance of the classifiers in the localization and skin datasets for mapper = 5 based on the sensitivity metric. For 90% training of the localization dataset, the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers with five mappers achieved the sensitivity value of 80.879%, 81.4741%, 83.447%, 84.478%, and 99.98%, respectively, while the proposed FCNB achieved the sensitivity value of 94.79%. Similarly, for the 90% training of the skin dataset, the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved the sensitivity values of 80.960%, 81.986%, 82.918%, 84.316%, and 34.76%, respectively. For the same dataset, the proposed FCNB with mapper = 5 achieved an accuracy value of 94.79%. From the analysis, the proposed FCNB seems to have the sensitivity value of 94.79% for both the localization and skin datasets.

### 3.4.3 Analysis Based on Specificity

Figure 5 presents the comparative analysis of the proposed FCNB classifier based on the specificity metric for the varying training percentages of the dataset and the mapper. Figure 5A presents the performance of the classifiers with the mapper size of 4 and varying training percentages of the localization dataset based on the specificity metric. For 90% training of the localization dataset, the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved specificity values of 72.884%, 73.953%, 75.9060%, 76.9617%, and 36.12%, respectively, while the proposed FCNB classifier had improved specificity value of 88.891%. Figure 5B presents the performance of the classifiers in the skin dataset along with mapper = 4 based on specificity. Here, the comparative models NB, CNB, GWO-CNB, CGCNB, and FNB classifiers achieved the specificity value of 70.8791%, 71.8203%, 72.7240%, 74.2836%, and 70.67%, respectively; however, the proposed FCNB classifier



**Figure 5:** Comparative Analysis of the FCNB Classifier Based on Specificity. For mapper = 4 in (A) localization dataset and (B) skin dataset. For mapper = 5 in (C) localization dataset and (D) skin dataset.

**Table 3:** Time Complexity of the Comparative Methods.

Methods	Time complexity (s)
NB	7.4
CNB	8.2
GWO-CNB	7
CGCNB	6.8
FNB	6.3
Proposed FCNB	5

had the specificity value of 88.891%. Figure 5C presents the performance of the classifiers in the localization for mapper = 5 based on the specificity metric. For 90% training of the localization dataset, the NB, CNB, GWO-CNB, CGCNB, and FNB classifiers with five mappers achieved the specificity value of 72.7007%, 73.8631%, 75.8758%, 76.991%, and 36.127%, respectively, while the proposed FCNB had a high specificity value of 88.89%. Figure 5D presents the performance of the classifiers in the skin dataset for mapper = 5 based on the specificity metric. For the skin dataset, the proposed FCNB with mapper = 5 achieved a specificity value of 88.8912%.

#### 3.4.4 Analysis Based on Time Complexity

Table 3 shows the time complexity of the comparative methods. The time complexity of the proposed FCNB is 5 s; on the other hand, the time complexity of the existing methods, such as NB, CNB, GWO-CNB, CGCNB, and FNB, is 7.4, 8.2, 7, 6.8, and 6.3 s, respectively. The time complexity of the proposed method is less when compared to the existing methods, which shows the effectiveness of the proposed method.

## 4 Conclusion

This work introduces the classification algorithm based on the fuzzy network, called FCNB, for data classification in the big data framework. The proposed FCNB classifier is designed through the integration of the correlation and the fuzzy theory, along with the MapReduce framework. As the proposed FCNB combines the fuzzy theory and the NB, it has improved classification performance in the large data framework. The proposed FCNB classifier is used along with the MapReduce framework for dealing with the large data environment. The simulation of the proposed FCNB classifier is done by considering the localization and skin segmentation datasets from the UCI repository. Also, the performance of the proposed FCNB classifier is compared against the existing NB, CNB, GWO-CNB, CGCNB, and FNB classifiers. From the simulation results, the proposed FCNB classifier shows improved performance in both the localization and skin segmentation datasets under the conditions of mapper = 4 and mapper = 5. For the skin segmentation, the FCNB classifier has high values of 91.78166%, 94.79%, and 88.8912% for accuracy, sensitivity, and specificity, respectively.

## Bibliography

- [1] Á. Arnaiz-González, A. González-Rogel, J. F. Díez-Pastor and C. López-Nozal, MR-DIS: democratic instance selection for big data by MapReduce, *Progr. Artif. Intell.* **6** (2017), 211–219.
- [2] C. Banchhor and N. Srinivasu, CNB-MRF: adapting correlative naive Bayes classifier and MapReduce framework for big data classification, *Int. Rev. Comput. Softw. (IRECOS)* **11** (2016).
- [3] A. Bechini, F. Marcelloni and A. Segatori, A MapReduce solution for associative classification of big data, *Inform. Sci.* **332** (2016), 33–55.
- [4] R. Bhukya and J. Gyani, Fuzzy associative classification algorithm based on MapReduce framework, in: *Proceedings of the International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 357–360, Davangere, 2015.

- [5] J. Chen, H. Chen, X. Wan and G. Zheng, MR-ELM: a MapReduce-based framework for large-scale ELM training in big data era, *Neural Comput. Appl.* **27** (2016), 101–110.
- [6] M. Duan, K. Li, X. Liao and K. Li, A parallel multiclassification algorithm for big data using an extreme learning machine, *IEEE Trans. Neural Netw. Learn. Syst.* **29** (2017), 2337–2351.
- [7] M. Elkano, M. Galar, J. Sanz and H. Bustince, CHI-BD: a fuzzy rule-based classification system for big data classification problems, *Fuzzy Sets Syst.* **348** (2018), 75–101.
- [8] A. Fernández, S. del Río, A. Bawakid and F. Herrera, Fuzzy rule based classification systems for big data with MapReduce: granularity analysis, *Adv. Data Anal. Classif.* **11** (2017), 711–730.
- [9] A. Haque, B. Parker, L. Khan and B. Thuraisingham, Evolving big data stream classification with MapReduce, in: *Proceedings of IEEE 7th International Conference on Cloud Computing*, pp. 570–577, Anchorage, AK, 2014.
- [10] M. Hazewinkel, Arithmetic series, in: *Encyclopedia of Mathematics*, Springer, Netherlands, 2001.
- [11] O. Hegazy, S. Safwat and M. E. Bakry, A MapReduce fuzzy techniques of big data classification, in: *Proceedings of the SAI Computing Conference (SAI)*, pp. 118–128, London, 2016.
- [12] G. B. Huang, Q. Y. Zhu and C. K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* **70** (2006), 489–501.
- [13] X. Huang, L. Shi and J. A. K. Suykens, Support vector machine classifier with pinball loss, *IEEE Trans. Pattern Anal. Mach. Intell.* **36** (2014), 984–997.
- [14] M. S. Kamal, S. Parvin, A. S. Ashour, F. Shi and N. Dey, De-Bruijn graph with MapReduce framework towards metagenomic data classification, *Int. J. Inform. Technol.* **9** (2017), 59–75.
- [15] W. Lin, Z. Wu, L. Lin, A. Wen and J. Li, An ensemble random forest algorithm for insurance big data analysis, *IEEE Access* **5** (2017), 16568–16575.
- [16] Localization dataset from <https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity>, Accessed on October 2017.
- [17] V. Lopez, S. del Rio, J. M. Benitez and F. Herrera, On the use of MapReduce to build linguistic fuzzy rule based classification systems for big data, in: *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1905–1912, Beijing, 2014.
- [18] V. López, S. del Río, J. M. Benítez and F. Herrera, Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data, *Fuzzy Sets Syst.* **258** (2015), 5–38.
- [19] S. A. Ludwig, MapReduce-based fuzzy c-means clustering algorithm: implementation and scalability, *Int. J. Mach. Learn. Cybernet.* **6** (2015), 923–934.
- [20] J. Maillou, I. Triguero and F. Herrera, A MapReduce-based k-nearest neighbor approach for big data classification, in: *IEEE Trustcom/BigDataSE/ISPA*, pp. 167–172, Helsinki, 2015.
- [21] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* **69** (2014), 46–61.
- [22] S. S. Patil and S. P. Sonavane, Enriched over\_sampling techniques for improving classification of imbalanced big data, in: *Proceedings of IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 1–10, San Francisco, CA, 2017.
- [23] B. Pei, F. Wang and X. Wang, Research on MapReduce-based fuzzy associative classifier for big probabilistic numerical data, in: *Proceedings of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 903–906, Chengdu, 2016.
- [24] G. Santafe, J. A. Lozano and P. Larranaga, Bayesian model averaging of naive Bayes for clustering, *IEEE Trans. Syst. Man Cybernet. Pt. B (Cybernetics)* **36** (2006), 1149–1161.
- [25] A. Segatori, F. Marcelloni and W. Pedrycz, On distributed fuzzy decision trees for big data, *IEEE Trans. Fuzzy Syst.* **26** (2018), 174–192.
- [26] Skin segmentation dataset from <https://archive.ics.uci.edu/ml/datasets/skin+segmentation>, Accessed on October 2017.
- [27] H. Storr, A compact fuzzy extension of the naive Bayesian classification algorithm, in: *Intelligent Systems in e-Commerce (ISeC)*, 2002.
- [28] I. Triguero, D. Peralta, J. Bacardit, S. García and F. Herrera, MRPR: a MapReduce solution for prototype reduction in big data classification, *Neurocomputing* **150** (2015), 331–345.
- [29] J. Zhai, S. Zhang and C. Wang, The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers, *Int. J. Mach. Learn. Cybernet.* **8** (2017), 1009–1017.