An Efficient Feature Subset Selection with Fuzzy Wavelet Neural Network for Data Mining in Big Data Environment

S. Varshavardhini[1] and A. Rajesh[2*]

[1]Research Scholar, Department of Computer Science and Engineering Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India. varshu28@gmail.com, Orcid: https://orcid.org/0009-0001-9553-0121

[2*]Research Supervisor, Department of Computer Science and Engineering Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India. arajesh.se@velsuniv.ac.in, Orcid: https://orcid.org/0000-0001-5435-0629

Received: March 15, 2023; Accepted: April 18, 2023; Published: May 30, 2023

Abstract

Big data refers to the massive quality of data being generated at a drastic speed from various heterogeneous sources namely social media, mobile devices, internet transactions, networked devices, and sensors. Several data mining (DM) and machine learning (ML) models have been presented for the extraction of knowledge from Big Data. Since the big datasets include numerous features, feature selection techniques are essential to eliminate unwanted and unrelated features which degrade the classification efficiency. The adoption of DM tools for big data environments necessitates remodeling the algorithm. In this aspect, this paper presents an intelligent feature subset selection with fuzzy wavelet neural network (FSS-FWNN) for big data classification. The FSS-FWNN technique incorporates Hadoop Ecosystem tool for handling big data in an effectual way. Besides, the FSS-FWNN technique involves three processes namely preprocessing, feature selection, and classification. In addition, quasi-oppositional chicken swarm optimization (QOCSO) technique is employed for the feature selection process and the FWNN technique is applied for the classification process. The design of QOCSO algorithm as an FS technique for big data classification shows the novelty of the work and the feature subset selection process considerably enhances the classification performance. An extensive set of simulations is carried out and the results are reviewed in terms of several evaluation factors in order to analyse the improvement of the FSS-FWNN approach. The experimental findings demonstrated that the FSS-FWNN approach outperformed the most current algorithms.

Keywords: Big Data, Data Mining, Data Classification, Feature Selection, Machine Learning, Metaheuristics.

1 Introduction

Recently, the exploitation of the internet is drastically increased, denoting the capability of processing different Data Mining (DM) processes in several application areas is getting more challenging and important. Big data describes large volumes of data that are hard to work with using traditional methods. Big data can be characterized by five Vs: volume, variety, velocity, veracity, and value. Volume refers

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 2 (May), pp. 233-248.
DOI: *10.58346/JISIS.2023.I2.015*

*Corresponding author: Research Supervisor, Department of Computer Science and Engineering Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India.

to the size and quantity of the data, which determines its potential insight. Variety refers to the types and sources of the data, which may be structured or unstructured. Velocity refers to the speed and frequency of the data generation and processing. Veracity refers to the quality and reliability of the data, which may be affected by noise, inconsistency, or incompleteness. Value refers to the usefulness and relevance of the data for business or decision making. Complex characteristics of data bring about a difficulty in obtaining common feature selection method for big data. A method specific to a background is feasible. To work with a minimum set of modelling features as they significantly reduce algorithm's complexity and computational cost.

The dramatic increase of storage techniques, and other features, like digital society, novel technologies, and appearance of mobile networks has enabled the development of big data (Peralta.D,2015). But the big data is derived from a specified quantity of redundant data. When processing and transmitting the redundant data, the complexity and time get drastically increased. For resolving this problem, redundant data within big data could be removed or mined using DM methods. But, the application of DM in several challenges, like, financial market analysis, medical data analysis, and network traffic monitoring aren't well understood (M. Minelli,2013).

In order to detect a solution for this problem, Machine Learning (ML) methods were introduced for implementing DM in big data, and intelligent analyses in different applications, like image processing, face recognition, medical diagnoses, voice recognition, DNA classification, signal processing, Internet of Things (IoT) or social networks. ML is a group of computer systems which enable computer programs for improving manually via experience for implementing a smart system (A. Fernandez,2014). ML build methods depending on training data in engineering problems for making decisions/predictions without being obviously programmed for doing this. Also, ML is most major sections of artificial intelligence (AI) and is speed up rapid growth in AI. Its main goal is to utilize computer algorithm for extracting data from the gathered data. But conventional ML methods aren't highly efficient in mining beneficial data from big data because of their limitation in managing difficult tasks (J. Bacardit,2013).

The adaption of DM tools for big data problems might need the restructuring of the algorithm and their involvement in similar environment. Amongst the various possibilities, the MapReduce model (J. Dean,2010) and their distributed file system, initially proposed by Google, provide a robust and effective architecture for addressing the analyses of big datasets. This method is now considered in DM, instead of other parallelization systems like MPI (Message Passing Interface), due to its simplicity and fault tolerance method (Kholod, I., 2020). Various current studies were concentrated on the parallelization of ML method utilizing the Map Reduce method (A. Srinivasan,2012). Currently, novel and more flexible tasks have seemed to expand the regular Map Reduce approaches, like Apache Spark (M. Zaharia,2012) that were effectively employed on different DM and ML challenges. Therefore, it can be trying to speed up DM algorithm and enhance their accuracy by removing noise and redundant data. The specialized survey defines 2 major kinds of data reduction methods. Initially, instance generation (J.A. Olvera-Lopez,2010) and instance selection processes are concentrated on the instance level. Then, feature extraction and feature selection (FS) (J.A. Lee,2007) model works at the characteristics level. Amongst the present methods, evolution methods were effectively utilized for FS methods (B.de la Iglesia,2013). In the present survey, there are no methods for tackling the feature space with evolution big data methods.

This paper designs a novel feature subset selection with fuzzy wavelet neural network (FSS-FWNN) for big data classification. The FSS-FWNN technique uses Hadoop Ecosystem tool for handling big data in an effectual way. For feature selection process, the FSS-FWNN technique designs a quasi-oppositional chicken swarm optimization (QOCSO), and the FWNN technique is applied for the

classification process. Moreover, the application of QOCSO algorithm as the feature subset selection process significantly enhances the accuracy of categorization A detailed results analysis is performed, and the findings are reviewed in terms of many areas, in order to inspect the better performance of the FSS-FWNN approach.

2 Literature Review

This paragraph performs a brief review of existing DM techniques for big data classification. (Amazal and Kissi,2021) proposed a distributed method for FS based mutual information (MI) technique, i.e., extensively used in ML and pattern recognition. A shortcoming of MI is that it neglects term frequency in FS process. The scheme presents a distributed FS method, i.e., MTF-MI, for improving the quality of the elected features. The presented method is executed on Hadoop with the help of Map Reduce programming paradigm. (Sleeman and Krawczyk,2021) proposed the primary compound architecture to handle multiclass big data challenges, simultaneously addresses the presence of multiple classes and huge amounts of data. They proposed for analyzing the instance level complexities in all classes, leads to understanding what makes learning complexities. They embedded this data in common resampling algorithm that permits to informative balance of multiple classes.

(Chen et al,2020). utilized 3 common datasets with a huge amount of parameters (Human Activity Recognition Using Smartphones Car Evaluation Database, and Bank Marketing,) for conducting the research. There are 4 major aims why FS is significant. Firstly, for simplifying the method by decreasing the training time, overfilling, amount of variables by improving generalization, and for avoiding the curse of dimension. This study adopts RF approach for selecting significant features in classification. (Zhong and Xiao,2017) with improved fusion node and DL models, we constructed an architecture for improving healthcare predictions. For logical inference and data extraction, the DL technique involves the complex use of ML methodologies such as Bayesian fusion and NN. The DL approach is used in conjunction with data fusion models to provide more comprehensive and trustworthy predictions from large amounts of healthcare data. In (Hernández et al,2020)., 2 novel hybrid neural frameworks integrating morphological neurons and perceptrons are presented. The first framework, referred known as MLNN, consists of a hidden layer of morphological neurons and an output layer of conventional perceptron that can extract features. The following structure, known as LMNN, consists of a feature extractor, an output layer of morphological neuron, and several perceptron layers for nonlinear classification. In (Varatharajan et al.,2018), FIR and IIR filters are first utilized for removing the nonlinear and linear delay presents in the input ECG signal. Additionally, filter is utilized for removing unnecessary frequency elements from the input ECG signal. The LDA is utilized for reducing the feature presents in the input ECG signal. The SVM method is broadly employed for recognizing patterns. But, conventional SVM technique doesn't appropriate for computing distinct features of the dataset. They utilize SVM method with a weighted kernel function approach for classifying further features from the input ECG signal.

(Xing and Bei,2019) presented an enhanced KNN approach and compare it with the conventional KNN approach. Aims at the drawbacks of conventional KNN approach in processing huge datasets, this study proposed an enhanced KNN method on the basis of density cropping and cluster denoising. It carries out denoising process by clustering, and enhances the classification efficacy of KNN approach by accelerating the search speed of KNN when preserving the classification accuracy of KNN method. (Hababeh et al.,2018) proposed a combined method for classifying and securing big data beforehand performing data duplication, analysis, and mobility. The necessity of securing big data

mobility is defined by categorizing the data based on the threat impact levels of their content into 2 classes; public and confidential.

3 The Proposed Model

In this study, a novel FSS-FWNN technique is developed solely for big data classification. The proposed FSS-FWNN technique performs three different processes (as shown in Fig. 1) namely preprocessing, QOCSO based feature subset selection, and FWNN based classification. In addition, for massive data administration, the Hadoop Ecosystem tool is employed. In the next sections, we'll go through the specifics of how each module works.

3.1. Design of Hadoop Ecosystem

The Hadoop Ecosystem and its components are used to handle Big Data. It is a kind of publicly accessible framework on a distributed platform that enables the stakeholder to process and store Big Data on computer clusters utilising a simpler programming style. It is designed to scale out to tens of thousands of nodes from a single server with improved fault tolerance. The 3 major elements of Hadoop consist of Hadoop Distributed File System (HDFS), Hadoop YARN, and MapReduce. Fig. 2 illustrates the structure of Hadoop.

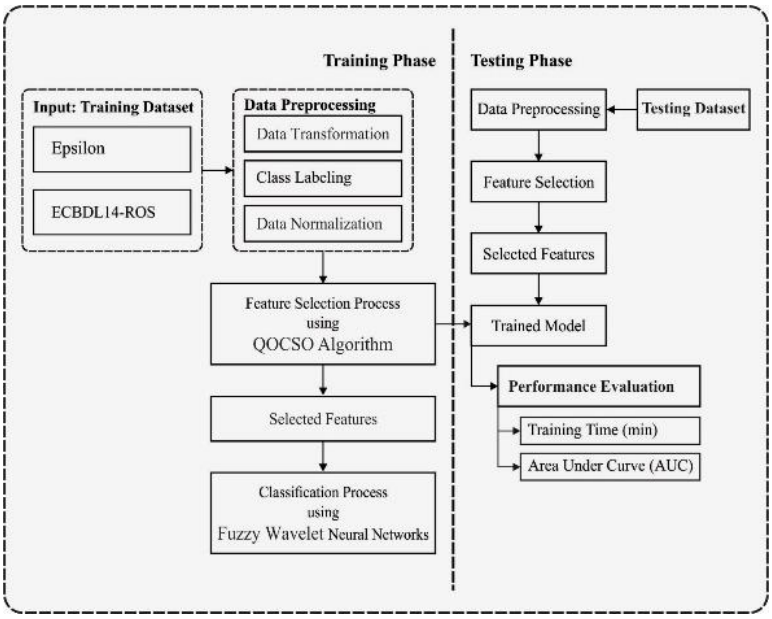

Figure 1: Overall Process of FSS-FWNN Model

Hadoop Distributed File System (HDFS)

The Google File System (GFS) serves as a model for the HDFS. It is modelled as a slave or master type, with the master having several real data nodes (also known as data nodes) and numerous name nodes (also known as metadata).

Hadoop Map Reduce

For providing large scalability on a thousand Hadoop clusters, Hadoop Map Reduce is employed i.e., known as programming structure at Apache Hadoop core. For processing huge data on massive clusters,

MapReduce is utilized. MapReduce task processing is consists of Reduce and Map phases. The framework deals with task scheduling, controlling, and failed task re-execution. The framework of MapReduce consists of single slave node handler and one master resource handler for each cluster node.

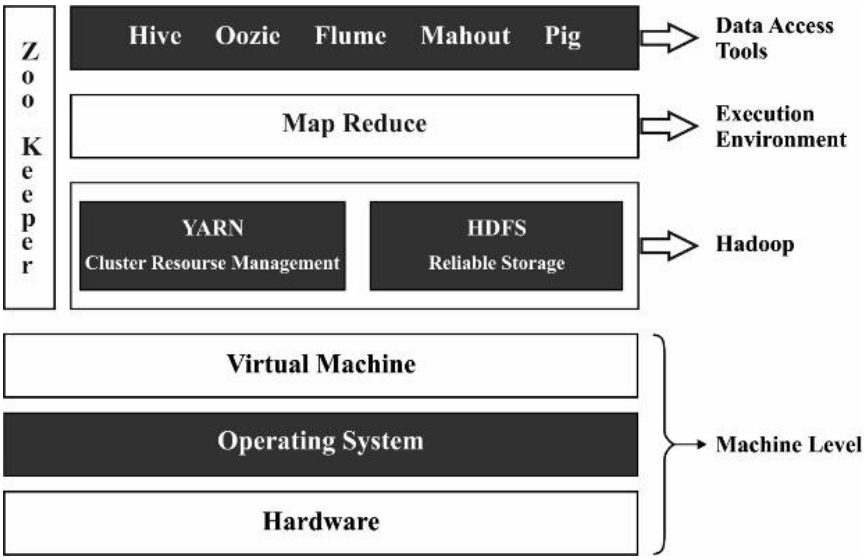

Figure 2: Structure of Hadoop

Hadoop YARN

It's a method for managing clusters. The essential characteristic of the second Hadoop generation may be described using the cumulative knowledge of the first Hadoop generation. YARN acts as a resource manager and core structure for Hadoop clusters, providing data governance tools, dependable operations, and security. (Selvi. R.T, 2021). For managing Big Data, other platform elements and tools could be installed on the Hadoop framework.

Map Reduce Implementation

In this approach, they implement MRODC technique for enhancing the classification efficiency and scalability. The MRODC approach tasks are given below:

- Based on N-gram, calcite each sentence's Polarity score.
- Based on the Polarity score, carry out data classification.
- Based on classified data, calculate term frequency and new words.

By applying different text mining methods, essential data from HDFS could be pre-processed. Using Map function, the concurrent iteration implementation is performed, i.e., called as reduce and Combiner functions respectively.

Map Phase: Each line is now accepted sequentially by the Map task as a number of key-value pairs that make up the input for the Map function. Each data object's value is first computed using the created corpus by the Map function, which then transmits its results to the Combiner function based on the varied gram sizes.

Combine Phase: The whole data object was derived firstly from Combiner function in this stage from Map function and data classification is performed on the basis of an equivalent class. Consequently,

it integrates the whole data with equivalent class values, in an equivalent class, it records the amount of instances and toward Reducer functions, and the result of each cluster is transmitted.

Reduce Phase: From different classes, the Reduce function derive whole data in this stage i.e, the Combiner function output. Later, the amount of each data in different class labels are calculated and final results are stored in HDFS and class labels as well as succeeding iteration will initiate.

3.2. Data Preprocessing

At the initial stage, data preprocessing take place in three levels namely data transformation, class labeling, and data normalization. Firstly, the input data in the .xls format is transformed into the useful .csv format. Then, the class labeling process is performance in which the samples are allotted to respective class labels. Lastly, the data normalization process is carried out by the use of min-max normalization, as specified below.

$$Min - Max.Norm = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{min} and x_{max} denotes the minimum and maximum values of the samples.

3.3. Design of QOCSO Algorithm for Feature Selection

During the feature subset selection process, the QOCSO algorithm is derived to choose an optimal subset of features. The CSO algorithm is stimulated by the hierarchal order of a chicken swarm and the behavior of the individual chickens. In the fundamental CSO approach, there are 3 types of roles, chicks, roosters, and hens, each containing distinct behaviors specification. In the succeeding section, they provide fundamental assumptions for the CSO approach:

(1) The CSO approach separates a chicken swarm into a small number of groups, all of them have a few chicks, single roosters, and numerous hens.
(2) The identities of chicks, hens, and roosters are defined by their fitness values, the optimal one is chosen as rooster, the worst one is the chick, and other individuals represent hens. All the hens arbitrarily select single rooster as her mate and become a member of his group, and all chicks also arbitrarily choose single hen as its mother.
(3) In the entire population, the separate identities, the spouse relations, and the mother children relations aren't changed for G generation (G represents the iterative cycle), and the identities, the spouse relations, and the mother children relations would be upgraded afterward G generation.
(4) For food, every single group of the complete population follows their spouse roosters, and each individual inside a group will arbitrary compete with the other. A person with the best fitness value is quite likely to be able to eat.

Every chicken is defined as its location. Where RN, HN, CN, and MN represents the amount of mother hens, roosters, hens, and chicks, correspondingly, and $x_{i,j}^{t}$ denotes the location of arithmetical i^{th} chicken in the j^{th} dimension space on the t^{th} iteration, whereas $i \in \{1, \dots, N\}, j \in \{1, \dots, D\}$, and $t \in \{1, \dots, T\}$ and N, D, and T represent the whole amount of chickens, the dimensional amount, and the maximal iteration time, correspondingly. Chicks, roosters, and hens have certain location upgrade formulas. Fig. 3 demonstrates the flowchart of CSO. For roosters, recurrent location is determined below:

$$x_{i,j}^{t+1} = x_{i,j}^{t} * \left(1 + Randn(0, \sigma^2)\right), \quad (2)$$

$$\sigma^2 = \begin{cases} 1, & if\ f_i \le f_k, \\ \exp\left(\frac{f_k - f_i}{|f_i| + \varepsilon}\right) & \text{otherwise } k \in [1, RN], k \ne i. \end{cases} \quad (3)$$

Now, Randn $(0, \sigma^2)$ denotes an arbitrary amount follows Gaussian distribution with anticipation of 0 and difference of σ^2, ε represents a smaller constant, k indicates the amount of other roosters i.e., selected arbitrarily, and f_i and f_k denotes the fitness value of i^{th} and the k^{th} rooster, correspondingly.

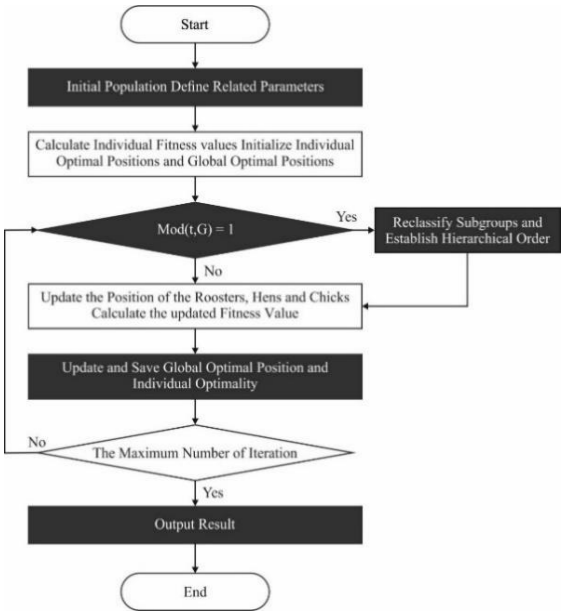

Figure 3: Flowchart of CSO

The recurrent location of a hen is determined below:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + C_1 * Rand\ * \left(x_{r_1,j}^{t} - x_{i,j}^{t}\right) + C_2 * Rand\ * \left(x_{r_2,j}^{t} - x_{i,j}^{t}\right), (4)$$

$$C_1 = \exp\left(\frac{(f_i - f_{r_1})}{(abs(f_i) + \varepsilon)}\right), \quad (5)$$

$$C_2 = \exp\left(f_{r_2} - f_i\right). \quad (6)$$

Now, C_1 & C_2 denotes the learning factor, Rand represents an arbitrary amount that follows uniform distribution in the range of zero and one, r_1 indicates the index of rooster i.e., the spouse of i^{th} hen, r_2 denotes the amount of roosters or hens i.e., chosen arbitrarily, and $r_1 \ne r_2$. The recurrent location of a chick is determined below:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL * \left(x_{m,j}^{t} - x_{i,j}^{t}\right), \quad (7)$$

Whereasx_m^t. denotes the mother hen of chick and FL is an arbitrary f^jactor in the range of [0,2]. Hence, the fundamental CSO approach is displayed in Algorithm 1.

The QOCSO algorithm is aimed to improve the CSO algorithm's convergence rate by using the idea of QOBL. In contrast to arbitrary numbers, (Tizhoosh,2005) established the Oppositional Based Learning (OBL) approach, which incorporates opposing numbers with a high possibility of achieving a solution. Convergence speed and accuracy are improved by combining metaheuristic and OBL techniques. Furthermore, OBL had previously been developed to QOBL, demonstrating that using quasi opposite number instead of opposite number is more successful in discovering global optimal outcomes. The following are arithmetically supplied QOBL descriptions:

Consider χ represents a real number in I-dimension space. Now, the opposite number x^o and the quasi-oppositional number x^{qo} (of x) is determined using Eq. (8) & (9), correspondingly:

$$x^0 = a + b - x \tag{8}$$

Whereas $x \in a$ & $\in b$].

$$x^{qo} = rand\left(\frac{a+b}{2}, x^0\right) \tag{9}$$

Here, $X(x_1, x_2, \dots, x_n)$ denotes a point in n-dimension space. Where the oppositional point, $X^o(x_1^o, x_2^o, \dots, x_n^0)$ is represents in Eq. (10); and the quasi-oppositional point, $X^{qo}(x_1^{qo}, x_2^{qo}, \dots, x_n^{qo})$, is determined in Eq. (11):

$$x_i^o = a_i + b_i - x_i \tag{10}$$

In which $x_i \in \mathbb{R}$ & $x_i \in [a_i, b_i] \forall i \in 1,2, \dots, n$.

$$x_i^{qo} = rand\left(\frac{a_i + b_i}{2}, x_i^o\right) \tag{11}$$

QOBL is applied in QOCSO for generation jumping and population initialisation (Truong,2019). QOBL based generation jumping assist the jump for a novel candidate solution that carries an optimal fitness value. The variable, j_r (jumping rate), determined either to keep a present solution or jump to a quasi-opposite solution. The proposed technique makes use of QOCSO algorithms to identify the set of features that optimises classification accuracy while requiring the least amount of FS. Because to the size of the feature space—which has each feature represented by a single dimension with a range of zero to one—a clever searching approach is required to locate the optimal place in the search space that maximises the given FF. The FF for the QOCSO is to maximize classification efficiency on the validation set provided the training data, as displayed in Eq. (12) when retaining minimal amount of FS.

$$f_\theta = \omega * E + (1 - \omega)\frac{\sum_i \theta_i}{N}, \tag{12}$$

Whereas f_θ denotes the FF provided a vector θ sized N with zero or one component signifying selected or unselected features, N denotes the overall amount of features in dataset, E indicates the classification error rate and ω represents a constant managing the significance of classification efficiency to the amount of FS (Hafez,2015). The utilized parameters are similar to the amount of features in the provided dataset. Each parameter is constrained in the range zero and one, in which the parameter values approach one; its receptive features are candidates to be chosen in classification. In single fitness estimation, the parameter is threshold for deciding the accurate features to be calculated as in Eq. (13).

$$f_{ij} = \begin{cases} 1 \; if \; X_{ij} > 0.5 \\ 0 \; otherwise, \end{cases} \tag{13}$$

whereas X_{ij} denotes the dimensional value for search agent i at dimension j. When upgrading the chicken swarm location; solution, at few dimensional the upgraded value could violate the limitations; [0,1], and therefore they utilized simple truncation rule for ensuring parameter limit. By the training procedure, every chicken location denotes one attribute subset. The training set is utilized for evaluating the FWNN on the validation set through the optimization for guiding the FS method. The classification is estimated on a validation set inside the FF. Additionally, the utilized FF integrates reduction size and classification accuracy.

3.4. Data Classification using FWNN Technique

Finally, the FWNN technique receives the chosen feature subset as input and performs classification process where the input data instances are allotted to respective class labels. In this FWNN architecture,

the resultant part of its fuzzy rules can be defined using the wavelet functions. The FWNN model includes five distinct layers as defined in the following:

Layer-1. The input layer only transfers to the succeeding layer the input signal vector $x = \{x_1, x_2, \dots, x_n\}$

Layer-2. At the layer of fuzzification, the MFs (MFs) are under parameterization for matching with the particle needs for several applications. For example, a Gaussian MF is defined as follows.

$$A_{qr}(x_q) = \exp\left[-\frac{1}{2}\left(\frac{x_q - a_{qr}}{b_{qr}}\right)^2\right], \quad (14)$$

where for $q = 1,2, \dots, n$ and $r = 1,2, \dots, k_q$, A_{qr} can be integrated to the rth MF occurs in the provided rule and validated for the qth element of the input vector. The modifiable variables are a_{qr} and b_{qr}, denoting the MF's center and width correspondingly.

Layer-3. It denotes the inference layer. Consider a set of m rules, where R_i is a provided rule and $i = 1,2, \dots, m$, all rules actually generate an outcome μ_i by the aggregation A_{qr} by the use of T-norm. The outcome of the pth rule in the layer can be represented as follows:

$$\mu_p = \prod_{q=1}^{n} A_{qr}(x_q), p = r_1, r_2, \dots, r_n, \quad (15)$$

where $r_1 = 1, \dots, k_1, r_2 = 1, \dots, k_2, r_n = 1, \dots, k_n$.

The outcomes of the layer are summed to the summation node positioned amongst Layer-3 and Layer-4. The node's output is subsequently used in the normalisation process level (Linhares,2015)

Layer-5. It designates the layer of normalisation where the outcome's normalisation factor is located. of the ith rule, $\overline{\mu}$ can be represented as follows:

$$\overline{\mu} = \frac{\mu_i}{\beta} = \frac{\mu_i}{\sum_{i=1}^{m} \mu_i}, i = 1,2, \dots, m. \quad (16)$$

Layer-6. This is the FWNN's resulting layer. The Mexican Hat family of wavelets is seen here can be employed and is mathematically expressed as follows.

$$\psi(x) = \frac{1}{\sqrt{d}}\left(1 - \left(\frac{x-t}{d}\right)^2\right) \exp\left[-0.5\left(\frac{x-t}{d}\right)^2\right]. \quad (17)$$

The input of the wavelet layer denotes the normalized weights $\overline{\mu}$ and the input vector $x = \{x_1, x_2, \dots, x_n\}$, whereas the output of this layer is denoted as f_j can be defined as follows.

$$f_j = \overline{\mu}\Psi_j, j = 1,2, \dots, m,$$

$$f_j = \overline{\mu}_j \sum_{i=1}^{n} (1 - T_{ij}^2) \exp\left[-0.5T_{ij}^2 \frac{1}{\sqrt{d_{ij}}}\right], \quad (18)$$

where the term $T_{ij} = \frac{x_i - t_{ij}}{d_{ij}}, d_{ij} > 0$ is used for simplifying the mathematical equation and n denotes the wavelet function count in the Layer-5.

Layer-6. At the output layer, every signal from the wavelet neuron is summed using Eq. (19):

$$y = \sum_{j=1}^{m} f_j. \quad (19)$$

It is noticed that the FWNN related parameters are available in layers 2 and 5. In addition, the MFs and wavelet functions are modified using learning algorithm like back propagation (BP) approach.

4 Performance Validation

This section examines the big data classification outcome of the proposed technique. The proposed model is tested using Epsilon and ECBDL14-ROS dataset. The details related to the dataset are shown in Table 1.

Table 1: Dataset Descriptions

Dataset	**Epsilon**	**ECBDL14-ROS**
Training Instances	400000	65003913
Test Instances	100000	289917
Number of Features	2000	631
Splits	512	32768

Table 2 investigates the classification results analysis of the proposed FSS-FWNN technique on the Epsilon dataset in terms of Training and Testing AUC. Fig. 4 showcases the training AUC analysis of the FSS-FWNN technique on the applied Epsilon dataset under varying number of selected features. The figure demonstrated that the FSS-FWNN technique has showcased effective outcomes with the maximum training AUC on the applied training set. For instance, with the chosen features of 721, the FSS-FWNN technique has offered a higher training AUC of 73.56% whereas the LR, NB, and SVM models have accomplished a lower training AUC of 69.85%, 71.54%, and 68.55% respectively. Similarly, with the chosen features of 110, the FSS-FWNN technique has obtained a maximum training AUC of 70.42% whereas the LR, NB, and SVM models have accomplished a lower training AUC of 64.96%, 68.03%, and 64.92% respectively.

Table 2: Results Analysis of Various Methods with Proposed FSS-FWNN Method on Epsilon Dataset in terms of Training and Testing AUC (%)

Selected Features	**Logistic Regression**	**Naive Bayes**	**SVM**	**FSS-FWNN**
Training Set				
2000	67.86	70.38	64.40	72.85
721	69.85	71.54	68.55	73.56
337	68.73	70.54	68.05	72.05
110	64.96	68.03	64.92	70.42
Testing Set				
2000	67.84	70.08	64.33	72.85
721	70.00	71.27	68.65	73.46
337	68.67	70.30	67.99	71.29
110	64.97	67.94	64.93	69.51

Fig. 5 illustrates the testing AUC analysis of the FSS-FWNN method on the applied Epsilon dataset under various number of selected features. The figure exhibited that the FSS-FWNN approach has demonstrated effectual results with the higher testing AUC on the applied testing set. For sample, with the chosen features of 721, the FSS-FWNN algorithm has offered a higher testing AUC of 73.46% whereas the LR, NB, and SVM approaches have accomplished a lower testing AUC of 70.00%, 71.27%, and 68.65% correspondingly. Followed by, with the chosen features of 110, the FSS-FWNN technique has obtained a maximum testing AUC of 69.51% whereas the LR, NB, and SVM methodologies have accomplished a minimal testing AUC of 64.97%, 67.94%, and 64.93% correspondingly.

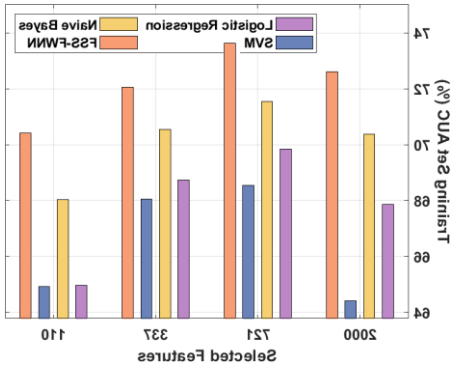

Figure 4: Training Set AUC Analysis of FSS-FWNN Model on Epsilon Dataset

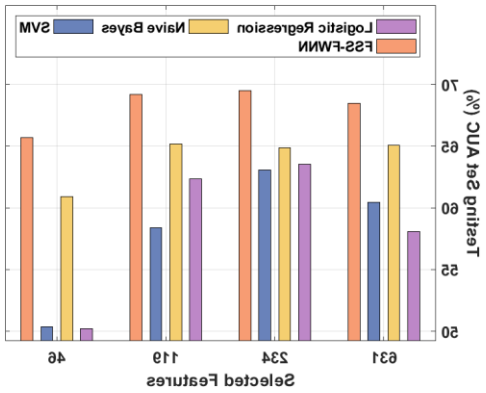

Figure 5: Testing Set AUC Analysis of FSS-FWNN Model on Epsilon Dataset

Table 3 examines the classification outcomes analysis of the presented FSS-FWNN algorithm on the ECBDL14-ROS dataset with respect to Training and Testing AUC. Fig. 6 demonstrates the training AUC analysis of the FSS-FWNN method on the applied ECBDL14-ROS dataset under different number of selected features. The figure exhibited that the FSS-FWNN approach has portrayed effective outcome with the maximal training AUC on the applied training set. For instance, with the chosen features of 631, the FSS-FWNN algorithm has offered an increased training AUC of 71.06% whereas the LR, NB, and SVM approaches have accomplished a reduced training AUC of 58.21%, 67.14%, and 59.66% correspondingly. At the same time, with the chosen features of 046, the FSS-FWNN methodology has gained a superior training AUC of 69.48% whereas the LR, NB, and SVM methods have accomplished the least training AUC of 50.17%, 61.36%, and 50.32% correspondingly.

Table 3: Results Analysis of Various Methods with Proposed FSS-FWNN Method on ECBDL14-ROS Dataset in terms of Training and Testing AUC (%)

Selected Features	Logistic Regression	Naive Bayes	SVM	FSS-FWNN
Training Set				
631	58.21	67.14	59.66	71.06
234	64.16	66.73	63.69	70.32
119	63.09	67.32	58.84	75.25
046	50.17	61.36	50.32	69.48
Testing Set				
631	58.08	65.06	60.46	68.44
234	63.52	64.89	63.07	69.46
119	62.35	65.16	58.41	69.19
046	50.22	60.93	50.39	65.67

Fig. 7 illustrates the testing AUC analysis of the FSS-FWNN approach on the applied ECBDL14-ROS dataset under distinct number of selected features. The figure showcased that the FSS-FWNN method has outperformed effectual outcome with the maximal testing AUC on the applied testing set. For sample, with the chosen features of 234, the FSS-FWNN method has offered an increased testing AUC of 69.46% whereas the LR, NB, and SVM techniques have accomplished a minimum testing AUC of 63.52%, 64.89%, and 63.07% respectively. Likewise, with the chosen features of 046, the FSS-FWNN approach has reached a maximum testing AUC of 65.67% whereas the LR, NB, and SVM algorithms have accomplished a lower testing AUC of 50.22%, 60.93%, and 50.39% correspondingly.

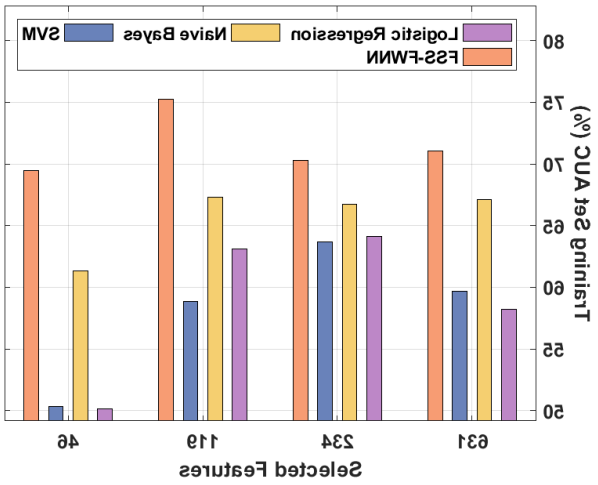

Figure 6: Training Set AUC Analysis of FSS-FWNN Model on ECBDL14-ROS Dataset

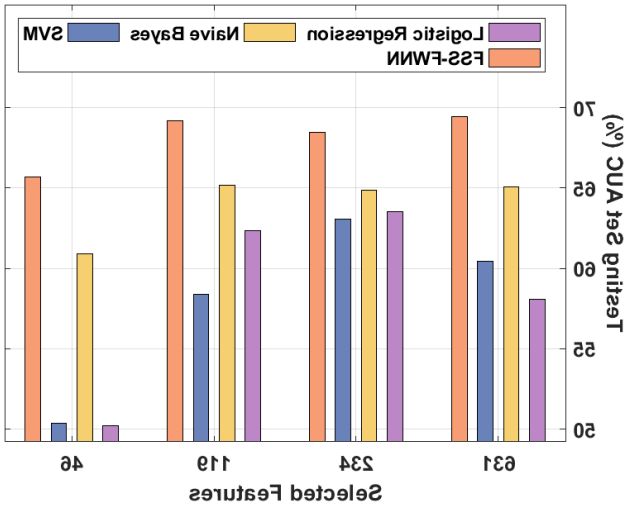

Figure 7: Testing Set AUC Analysis of FSS-FWNN Model on ECBDL14-ROS Dataset

A brief training time analysis of the FSS-FWNN technique takes place on the applied Epsilon dataset in Table 4 and Fig. 8. The experimental results stated that the FSS-FWNN technique has exhibited better performance with the lower training time under distinct number of chosen features. For instance, with 2000 features, the least training time of 5.002m has been required by the FSS-FWNN technique whereas an increased training time of 6.122m, 10.086m, and 5.570m have been provided by the LR, NB, and SVM techniques. Eventually, with 110 features, a reduced training time of 4.001m has been offered by the FSS-FWNN technique whereas a higher training time of 8.364m, 4.404m, and 7.791m has been provided by the LR, NB, and SVM techniques.

Table 4: Results Analysis of Various Methods with Proposed FSS-FWNN Method on Epsilon Dataset in Terms of Training Time (m)

Selected Features	Logistic Regression	Naive Bayes	SVM	FSS-FWNN
2000	6.122	10.086	5.570	5.002
721	6.823	5.674	6.828	5.167
337	8.136	5.122	8.424	5.003
110	8.364	4.404	7.791	4.001

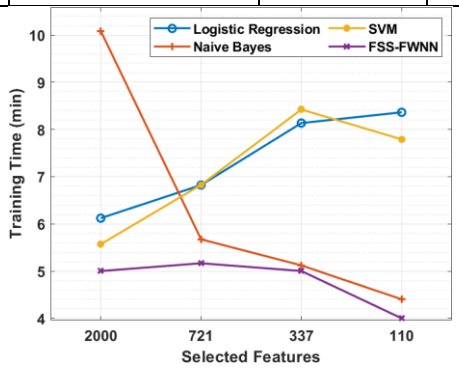

Figure 8: Training Time Analysis of FSS-FWNN Model on Epsilon Dataset

A detailed training time analysis of the FSS-FWNN approach takes place on the applied ECBDL14-ROS dataset in Table 5 and Fig. 9. The experimental outcomes referred that the FSS-FWNN approach has demonstrated optimum efficiency with the lesser training time under different number of chosen features. For sample, with 631 features, a minimum training time of 23.9558m has been necessary by the FSS-FWNN method whereas an improved training time of 77.4865m, 26.3587m, and 88.0647m have been given by the LR, NB, and SVM algorithms. Finally, with 46 features, a minimum training time of 3.3335m has been obtainable by the FSS-FWNN methodology whereas a superior training time of 16.3063m, 3.5848m, and 15.2387m have been providing by the LR, NB, and SVM methods.

Table 5: Results Analysis of Various Methods with Proposed FSS-FWNN Method on ECBDL14-ROS Dataset in Terms of Training Time (m)

Selected Features	Logistic Regression	Naive Bayes	SVM	FSS-FWNN
631	77.4865	26.3587	88.0647	23.9558
234	35.1277	10.2250	38.6870	09.4770
119	19.3830	05.3677	22.5475	05.0035
46	16.3063	03.5848	15.2387	03.3335

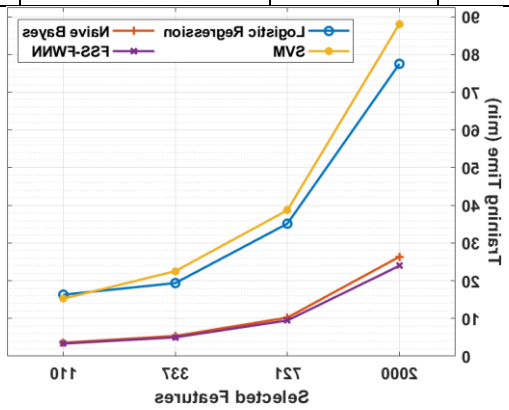

Figure 9: Training Time Analysis of FSS-FWNN Model on ECBDL14-ROS Dataset

By looking into the above-mentioned tables and figures, it is apparent that the FSS-FWNN technique has outperformed the existing techniques in terms of different measures. Therefore, it can be employed as an appropriate tool for big data classification.

5 Conclusion

This paper has presented a new FSS-FWNN technique for data classification in big data environment. For handling big data, Hadoop Ecosystem tool is used. The proposed FSS-FWNN technique performs three different processes namely preprocessing, QOCSO based feature subset selection, and FWNN based classification. The QOCSO algorithm is derived by the combination of QOBL concept to the classical CSO algorithm to boost its convergence rate. In addition, the inclusion of QOCSO based feature selection process helps to accomplish improved classification performance. For inspecting the superior performance of the FSS-FWNN technique, a comprehensive results analysis is made and the results are inspected in terms of various aspects. The experimental results showcased the supremacy of the FSS-FWNN technique over the recent state of art techniques. Finally, with 46 features, a minimum training time of 3.3335m has been obtainable by the FSS-FWNN methodology whereas a superior training time of 16.3063m, 3.5848m, and 15.2387m have been providing by the LR, NB, and SVM methods. In future, we will investigate the performance of the DM tools in big data environment using the data wrangling techniques.

References

[1] Amazal, H., & Kissi, M. (2021). A new big data features selection approach for text classification. *Scientific Programming*, *2021*, 1-10.

[2] Bacardit, J., & Llorà, X. (2009). Large scale data mining using genetics-based machine learning. *In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, 3381-3412.

[3] Chen, R.C., Dewi, C., Huang, S.W., & Caraka, R.E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(1), 1-26.

[4] De La Iglesia, B. (2013). Evolutionary computation for feature selection in classification problems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*(6), 381-407.

[5] Dean, J., & Ghemawat, S. (2010). MapReduce: a flexible data processing tool. *Communications of the ACM*, *53*(1), 72-77.

[6] Fernández, A., del Río, S., López, V., Bawakid, A., del Jesus, M.J., Benítez, J.M., & Herrera, F. (2014). Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(5), 380-409.

[7] Hababeh, I., Gharaibeh, A., Nofal, S., & Khalil, I. (2018). An integrated methodology for big data classification and security for improving cloud systems data mobility. *IEEE Access*, *7*, 9153-9163.

[8] Hafez, A.I., Zawbaa, H.M., Emary, E., Mahmoud, H.A., & Hassanien, A.E. (2015). An innovative approach for feature selection based on chicken swarm optimization. *In IEEE 7th international conference of soft computing and pattern recognition (SoCPaR)*, 19-24.

[9] Hernández, G., Zamora, E., Sossa, H., Téllez, G., & Furlán, F. (2020). Hybrid neural networks for big data classification. *Neurocomputing*, *390*, 327-340.

[10] Kholod, I., Shorov, A., & Gorlatch, S. (2020). Efficient Distribution and Processing of Data for Parallelizing Data Mining in Mobile Clouds. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *11*(1), 2-17.

[11] Lee, J.A., & Verleysen, M. (2007). *Nonlinear dimensionality reduction*, *1*. New York: Springer.
[12] Linhares, L.L., Fontes, A.I., Martins, A.M., Araújo, F.M., & Silveira, L.F. (2015). Fuzzy wavelet neural network using a correntropy criterion for nonlinear system identification. *Mathematical problems in engineering*, *2015*.
[13] Minelli, M., Chambers, M., & Dhiraj, A. (2013). *Big data, big analytics: emerging business intelligence and analytic trends for today's businesses*, *578*. John Wiley & Sons.
[14] Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., & Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, *34*, 133-143.
[15] Peralta, D., Del Río, S., Ramírez-Gallego, S., Triguero, I., Benitez, J.M., & Herrera, F. (2015). Evolutionary feature selection for big data classification: A mapreduce approach. *Mathematical Problems in Engineering*, *2015*, 1-11.
[16] Selvi, R.T., & Muthulakshmi, I. (2021). Modelling the map reduce based optimal gradient boosted tree classification algorithm for diabetes mellitus diagnosis system. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2), 1717-1730.
[17] Sleeman IV, W.C., & Krawczyk, B. (2021). Multi-class imbalanced big data classification on spark. *Knowledge-Based Systems*, *212*.
[18] Srinivasan, A., Faruquie, T.A., & Joshi, S. (2012). Data and task parallelism in ILP using MapReduce. *Machine learning*, *86*, 141-168.
[19] Tizhoosh, H.R. (2005). Opposition-based learning: a new scheme for machine intelligence. *In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, *1*, 695-701.
[20] Truong, K.H., Nallagownden, P., Baharudin, Z., & Vo, D.N. (2019). A quasi-oppositional-chaotic symbiotic organisms search algorithm for global optimization problems. *Applied Soft Computing*, *77*, 567-583.
[21] Varatharajan, R., Manogaran, G., & Priyan, M.K. (2018). A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing. *Multimedia Tools and Applications*, *77*, 10195-10215.
[22] Wang, Z., Qin, C., Wan, B., Song, W.W., & Yang, G. (2021). An adaptive fuzzy chicken swarm optimization algorithm. *Mathematical Problems in Engineering*, *2021*, 1-17.
[23] Xing, W., & Bei, Y. (2019). Medical health big data classification based on KNN classification algorithm. *IEEE Access*, *8*, 28808-28819.
[24] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *In Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, 15-28.
[25] Zhong, H., & Xiao, J. (2017). Enhancing health risk prediction with deep learning on big data and revised fusion node paradigm. *Scientific Programming*, *2017*.

Authors Biography

Varshavardhini S is now a Research Scholar received B.E., degree from New Prince Shri Bhavani College of Engineering and Technology in 2017 and the M.E., degree from Vels Institute of Science, Technology and Advanced Studies in 2019 and currently research scholar From Vels Institute of Science, Technology and Advanced Studies. Research interests includes Data Mining, Big Data. E-mail: varshu28@gmail.com. Orcid: https://orcid.org/0009-0001-9553-0121

Dr.A. Rajesh is now a Associate professor received the M. Tech. in Computer science and Engineering, from the VIT University, Vellore Tamil Nadu, India (2004) and his Ph.D. from Anna University, Chennai, Tamil Nadu, India (2017). He is presently the Associate Professor of Computer Science Engineering at the School of Engineering of VISTAS University, Chennai, Tamil Nadu, India, where he has established an advanced research Virtual Reality laboratory, emphasizing AR-VR visual hybrid tracking approach and the guiding part of a reliable Indoor navigation requests for 3D model of the environment. His research interest includes technology and applications of Machine Learning, AR-VR Technology and Trusted Network telecommunication. E-mail: arajesh.se@velsuniv.ac.in
Orcid: https://orcid.org/0000-0001-5435-0629