**European Journal of Molecular & Clinical Medicine**
**ISSN 2515-8260 Volume 07, Issue 3, 2020**

# STACKING ENSEMBLE LEARNING AND FEATURE SELECTION METHODS FOR DATA CLASSIFICATION

**[1]Dr. N. Tajunisha, [2]A. SHANMUGAPRIYA**

[1]Associate Professor, HOD COMPUTER SCIENCE DEPARTMENT, Sri Ramakrishna College of Arts and Science for women, Coimbatore.

[2]Research scholar Sri Ramakrishna College of Arts and Science for women, Coimbatore.

**ABSTRACT:**

In data mining approaches, classification is a major task. Various modern applications have adopted this. Recently Cascaded Fuzzy Relevance Vector Machine (FRVM) are developed to classify datasets. However single classifier will not give higher accuracy rather than the multiple classifiers. For solving this issue, ensemble learning is introduced in this work for the classification of samples. It is a way of combining various classifiers such as Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS) and Modified Convolutional Neural Network (MCNN) from which a novel classifier is formulated which performs better than any constituent classifier. This work consists of four major steps. First step consists of parallel operation data samples for classification. Second feature selection is performed by using Filter based functions are chi-squared filter, Euclidean Distance, Pearson correlation coefficient, Correlation Based Feature Selection (CFS), Fast Correlation Based Filter (FCBF), and Information Gain (IG). Thirdly outliers are removed by Fuzzy C means (FCM) clustering algorithm. Further in classification model, virtual pair is selected automatically by using Ant Colony Optimization (ACO) method and then Stacking Ensemble Learning (SEL) has been developed for classification reducing error rate and for improving accuracy. SEL is a technique in which two classifiers are trained using a single training dataset. Using k-fold validation, further divided the training set and formed the resultant model. Performance comparison results of various classifiers under two benchmark datasets such as Wisconsin Diagnostic Breast Cancer (WDBC) and PD600.

**Keywords:**Monotonicity, optimal virtual pair selection, knowledge learning, Feature Selection, Fuzzy C means (FCM) clustering, Stacking Ensemble Learning (SEL),Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS), Ant Colony Optimization (ACO), Modified Convolutional Neural Network (MCNN).

## I. Introduction

Everywhere across the globe, huge amount of data are collected as well as stored in databases in recent days. This will get increased in every day. Research facilities an enterprises are using Terabytes of data [1]. Such databases contains hidden knowledge and invaluable information. It is hard or impossible to mine those databases without automatic extraction techniques.

Huge amount of data are analysed automatically using data mining techniques and interesting relationship and knowledge are computed which are implicit in huge data volumes. Sifting process are automated using data mining via historical data for discovering new information. This is a major difference between statistics and data mining, where, statistician devised a

model for dealing with a specific analysis problem [2].From expert systems, it is also distinguishes data mining, where knowledge engineer built a model from rules extracted from expert experience. It is interesting to incorporate data mining with prior knowledge but it is a challenging problem.

On other side, development, demonstration and pushing specific algorithms and models use are focused mainly in traditional data mining research according to data collected from observations. However, while solving real-world problems with data mining there is a prior knowledge which is included with collected data [3].

For instance, if loan applicants A and B have the same attribute values in loan-approval applications, except that applicant A has greater income than applicant B, then applicant A is having high chance of getting loan. In other words, 'a higher income increases chance of loan approval' is an example of prior domain knowledge [4].

This issue has been as of late tended to by Cao's hypothetical system of space driven information mining, which features the significance of area insight, started up as far as angles, for example, space information, foundation data, earlier information, master information, etc [5]. One significant kind of earlier information in this setting depends on the monotonic connections among information and yield factors.

In numerous information arrangement applications one could have from the earlier information to the degree that, every single other thing being equivalent, an expansion in an information variable (trait) ought not prompt a diminishing (or increment) in class marks, with the previously mentioned credit endorsement case being an agent case of this [6]. It considers the way that in numerous applications, each information point may not be actually marked as one specific class, and in this way, it applies a to each information point.

It likewise uses master information concerning the monotonic relations between the reaction and indicator factors, which is spoken to as monotonicity imperatives [7]. Instances of other application areas in which this kind of information exists incorporate medication (for example smoking expands the likelihood of vascular sicknesses) and financial aspects (for example house costs increment or diminishing dependent on the area).

The earlier information on monotonicity, which can be obtained from area specialists, past viable experience, and the writing, gives helpful data about the central issue, notwithstanding the preparation information [8]. When considering this earlier information about the information, one needs to include some monotonicity requirements into the order model.

It has been indicated that an order method that consolidates monotonicity requirements can extricate information that is both progressively sensible and conceivable. Arrangement with monotonicity imperatives, otherwise called monotonic order, is an ordinal grouping issue where a monotonic limitation is available: a higher estimation of a trait in a model, fixing different qualities, ought not diminish its class task [9]. The monotonicity of relations between the reliant and logical factors is normal as an earlier information structure in information order.

The estimation of the information about monotonicity in learning models is of an extraordinary enthusiasm for two principle contentions. Initially, monotonicity forces imperatives on the forecast work. This reductions the size of the speculation space and furthermore the intricacy of the model [10]. Besides, the area specialists choose the acknowledgment or dismissal of the

models yielded in the event that they are steady with the space information, paying little mind to their exactness.

Numerous information learning calculations have been adjusted to have the option to deal with monotonicity limitations in a few styles. There are two stages to treat with monotonic characterization issues [11]. The first is to preprocess the information so as to "monotonizes" the informational collection, dismissing the models that abuse the monotonic limitations or choosing highlights to improve order execution and abstain from over fitting; and the subsequent one is to compel learning just monotone characterization capacities. This research work focusing the approach to learning with monotonicity constraints, proposed an Stacking Ensemble Learning (SEL) for the classification of samples. It is a way of combining various classifiers such as Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS) and Modified Convolutional Neural Network (MCNN) from which a novel classifier is formulated which performs better than any constituent classifier.

This research work is organized as follows. Section 2 reviews advantages and disadvantages in some of methods in monotonic classification. In Section 3, proposed approach are described. In Section 4 experimentation and results analysis are presented. In Section 5, conclusions are highlighted.

## 1. Literature Review

Monotonic classification with description of commonly used monotonic classification's brief review is presented in this section.

García et al [12] used evolutionary algorithms to implement a selection technique of highly effective hyper rectangles for tackling monotonic classification. An exhaustive experimental analysis is used for comparing proposed model using a huge amount of datasets derived from real regression and classification problems. With respect to mean absolute error and accuracy, other instance-based and rule learning models are outperformed by this evolutionary proposal as reported in results and it needs only few hyper rectangles.

Doumpos et al [13] considered the requirements of monotonicity in developing a Non-linear SVM credit rating models having linear programming. Models predictive ability is enhanced by introducing monotonicity hints as indicated in results.

Gonzalez et al [14] used resulting trees monotonicity degree for proposing a simple pruning mechanism. On monotonic data sets, various experimentation are conducted to study about performance of different decision trees. While holding monotonicity restriction, trees produced in Random forest are minimized in this work and it provided better performance in prediction when compared with standard algorithms.

Stiglic et al [15] estimated disease risk in hospital discharge record data in an optimum way using an implemented Support Vector Machine - Recursive Feature Elimination (SVM-RFE) technique. Prior knowledge from human disease networks extracted from hospital discharge historical data are incorporated in this technique and difficulties in constructing classifiers are minimized using this data.

From complex system, knowledge representations and feature selection methods of bioinformatics are adopted for predicting future risk in hospitalization according to highly imbalanced and 11,170 dimensional hospital discharge data with 7 million records which are collected in year 2008.

Li et al [16] incorporated prior knowledge with data mining using a presented fuzzy support vector machine (SVM) model. It includes the fact in various applications that, every input point will not be properly labelled as one specific class and so fuzzy membership is applied to every input point in this. By considering monotonic relations among predictor variables and response, expert knowledge is also utilized by this and in monotonicity constraints form they are represented.

Monotonically constrained fuzzy SVM classification problem is formulated here and termed as FSVM. Dual optimization problem of this is derived and its monotonic properties are analysed therotically. For ensuring bounded and unique solution, Tikhonov regularization technique is applied. In order to evaluate model's ability in retaining monotonicity, proposed a new measure called frequency monotonicity rate.

On synthetic and real world datasets, the experimentation was conducted and it indicates that, various contributions from every data are considered in this model and monotonicity's prior knowledge is used. In terms of retaining monotonicity and predictive ability, it has huge advantages over original FSVM and SVM models in classification problems.

Hu et al [17] used large margin principle for introducing a feature selection with monotonicity constraint. For monotonic classification, two new evaluation algorithms are designed by introducing the monotonicity constraint into existing feature selection algorithms based on margins. Some real and artificial datasets are used for testing proposed algorithms and its effectiveness is shown in experimental results.

Rus et al [18]implemented an intelligent tutoring system for teaching self-regulatory processes to students in complex science topics learning. In specific, student-generated paragraphs in prior knowledge activation, a self-regulatory process based students' mental models detection is focused. Two major classes of techniques are described here and different machine learning algorithms are combined with every technique.

Also provided a detailed comparison between techniques and algorithms. Prediction of techniques and human judgements are compared to evaluate the performance of proposed techniques with set of prior knowledge activation paragraphs collected from previous experiment in MetaTutor on college students. Based on experimentation, highly accurate results are produced using a content based technique with Bayes Nets algorithm and word-weighting.

Chen et al [19] used support vector machine having monotonicity constraints derived from financial experts prior knowledge for implementing a novel rating model. On real world data sets, experimentations are conducted and it indicates that, proposed technique is a domain knowledge oriented and data driven technique. During collection process, in data occurring, monotonicity loss can be corrected using this and when compared with conventional counterpart, better performance is exhibited.

Pan et al [20] considered monotonic and non- monotonic features separately for proposing a feature selection algorithm in ordinal classification. Hybrid monotonic classification consistency assumption is introduced first and relevance between features and ordinal classification decision are done using a defined feature evaluation function.

Then optimum feature subset is searched by combing genetic algorithm (GA) and reported measure. Feature size are reduced effectively and classification performance is enhanced using a proposed technique as shown implemented numerical experimentation.

Cano et al [21] introduced the utilization of preparing set choice to pick the best cases which lead the monotonic classifiers to get increasingly exact and proficient models, satisfying the monotonic imperatives. To show the advantages of the proposed preparing set choice calculation, called MonTSS, do an experimentation more than 30 informational indexes identified with ordinal grouping issues.

Bartley et al [22] presented a novel strategy for joining monotone information into Random Forest classifiers. As opposed to monotonising the trees in the gathering, consider Random Forest as a type of weighted neighborhood plot and figure an advancement issue to negligibly transform the model to expand monotonicity. This methodology has the upside of clearly consolidating fractional monotonicity (in a few, instead of all, highlights).

What's more, apply the new strategy to genuine datasets and research the effect of monotonicity and test size on prescient precision. Here locate that Random Forest is frequently truly adept at perceiving monotonicity without alteration. Furthermore, the expansion in monotonicity is altogether decidedly related to increments in exactness.

From the above audit these strategies for picking the best occasions which lead the monotonic classifiers to acquire progressively exact and effective models. Furthermore to the issue of building those classifier in unequal settings, it is likewise imperative to appropriately quantify the grouping execution in such settings.

The old style grouping precision ought to be supplanted with a measure that will put more concentrate on the arrangement execution for uncommon positive examples. So the single classifier won't give higher exactness instead of the numerous classifiers. So as to unravel this issue, outfit learning has been presented in this work for the order of tests.

## 2. Proposed Methodology

This work consists of four major steps. First step consists of parallel operation data samples for classification. And database parallelization done by map reduce method for time consumption for large scale data. Second feature selection is performed by using Filter based functions are chi-squared filter, Euclidean Distance, Correlation Based Feature Selection (CFS), Pearson correlation coefficient, Fast Correlation Based Filter (FCBF), and Information Gain (IG).

Thirdly outliers are removed by Fuzzy C means (FCM) clustering algorithm. Further in classification model, virtual pair is selected automatically by using Ant Colony Optimization (ACO) method and then Stacking Ensemble Learning (SEL) has been developed for classification reducing error rate and for improving accuracy.SEL is a technique in which two classifiers are trained using a single training dataset.

Using k-fold validation, further divided the training set and formed the resultant model. It is a way of combining various classifiers such as Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS) and Modified Convolutional Neural Network (MCNN) from which a novel classifier is formed which produces better classifier performance. Proposed methodology's overall process is illustrated in figure 1.

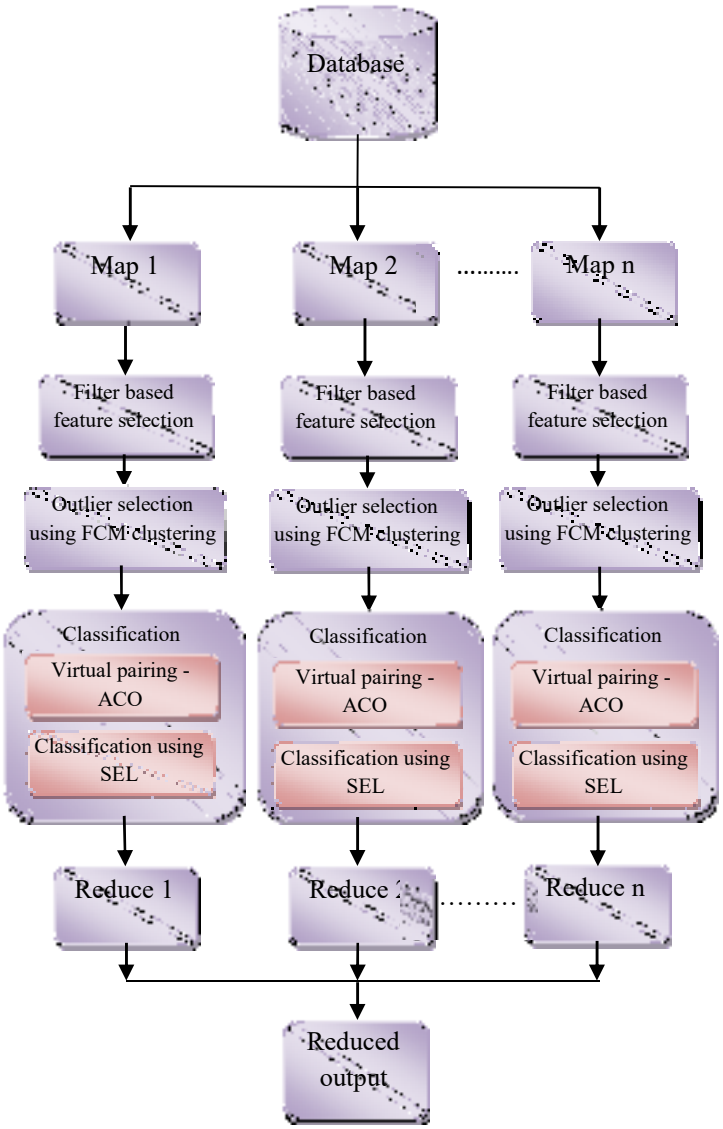


**Figure .1. The overall process of the proposed methodology**

### 3.1. Definition of Monotonicity

In most of the data mining applications, as assumption is made that, higher or better outcome classes can be obtained with greater or better observation evaluation on specific attributes. It indicates, between output and input variables, there exist a monotonic relationship, which shows that, increase in inut value will leads to either increase or decrease in output value.

**Definition 1(monotonicity):** Let$f(x)$: $R^n \rightarrow R$. On this input space$R^n$, defined a partial ordering≤. On the space R, defined a linear ordering≤. Therefore, following statement needs to be satisfied to make f as a monotonic one:

$$x \leq x' \Rightarrow f(x) \leq f(x') for any x and x' \qquad (1)$$

Where, on input space $R^n$, partial order is defined in an intuitive manner, so that, $x = (x_1, x_2, \ldots, x_n)$ and $x' = (x_1', x_2', \ldots, x_n')$, say $x \leq x'$ if and only if $x_i \leq x'$i for $i = 1, \ldots, n$. A monotonic function will have monotonicity property. In classficiation, if a function is observed as a monotonic by an experts, then it will have monotonicity property.

**Definition 2 (monotonicity of dataset):**For a dataset $D = \{(x_i, y_i) \mid i = 1, 2, \ldots, N\}$,suppose that a instances pair $(x_i, y_i)$ and $(x_j, y_j)$ does not breach monotonicity, when$x_i \leq x_j \Rightarrow y_i \leq y_j$. Dataset D's monotonicity is measured using a The Frequency Monotonicity Rate (FMR). Ratio of data pairs available in a dataset, which do not transgress monotonicity condition defined FMR. The Frequency Monotonicity Rate (FMR) is expressed as,

$$FMR = FM/P \tag{2}$$

Where, observed pairs count is referred as P and pair count, which are not violating monotonicity condition is referred as FM. The error which is having association with monotonic prior knowledge based on this pair is expressed as,

$$e(x, x') = \begin{cases} f(x) - f(x'), & if f(x) > f(x') \\ 0, & if f(x) \leq f(x') \end{cases} \tag{3}$$

There must be a small value of error which is having association with monotonic prior knowledge as per (2), which is needed for reducing $(x, x')$, $\square = 1, \ldots, M$, at the same time. A general method to achieve this is to minimize penalty function,

$$E_M = \sum_{i=1}^{M} \beta_i e(x, x') \tag{4}$$

Where, a non-negative number is specified by every $\beta_i$, it may be a constant or a variable.

### 3.2. Parallel Operation by Map Reduce

There is a difference between previously analysed parallel computation models and MapReduce. Sequential and parallel computation are interleaved in it.

**MapReduce Basics:** Information about support vector $SV_i$ are considered in basic information unit in MapReduce programming paradigm and their data are binary strings. A set of $\langle RV_i, data \rangle$pairs is given as input to any MapReduce algorithm. In three stages, non-support vectors are eliminated. They are, reduce stage, shuffle stage and map stage.

A single $\langle RV_i, data \rangle$pairs I given as an input to mapper $\mu$ in map stage and any number of new $\langle RV_i, data \rangle$pairs are produced at the output. At a time, map operation is performed only on one pair. Easy parallelization is allowed by this as different $\langle RV_i, data \rangle$for map are processed using various machines. In shuffle stage, underlying system, where Map reduce is implemented, sends all $\langle RV_i, data \rangle$values having association with an individual $\langle RV, data \rangle$ to same machine.

All values $\langle RV, data \rangle$ having association with s single key k are taken as a input in reducer $\rho$ in reduce stage and multiset of $\langle RV, data \rangle$pairs having same key are produced at output. MapReduce computation's sequential aspects are highlighted using this. All maps should be finished before reduce stage initialization. All values can be accessed using same key by reducer, on these values, sequential computations can be performed using this.

Reducers operating on different keys, which can be simultaneously executed are observed to exploit parallelism in reduce step. There are huge rounds of reduce functions and different map in MapReduce paradigm program, which are performed one after another.

There is a reducers and mappers sequence in $\{\mu_1, \rho_1, \mu_2, \rho_2, \ldots, \mu_R, \rho_R\}$ map reduce program. A multiset of $\langle RV_i, data \rangle$pairs is given as an input and represented as $U_0$. For executing program on input $U_0$: For $r = 1, 2, \ldots, R$, do:

- **Execute Map:**Mapper $\mu r$ is feed with every pair $U_0$ in $U_{r-1}$ and execute it. Support vector sequence will be generated by mapper. Assume $U_r'$ as multiset of key$\langle RV_i, data \rangle$ pairs output of $\mu_r$.

- **Shuffle:** For every k, assume $V_{k,r}$ as multiset of values vi so that $\langle RV_i, data \rangle \in U_r'$ . From $U_r'$ , multisets $V_{k,r}$ is constructed using underlying MapReduce implementation.
- **Execute Reduce:** Separate instance of reducer $\rho_r$ are feed with k and some $V_{k,r}$'s arbitrary permutation for every k and execute it. Tuples sequence $\langle k, data_1' \rangle, \ldots \langle k, data_n' \rangle, \ldots$ Is generated by a reducer. Assume Ur as multiset of $\langle RV_i, data \rangle$ pairs output of $\rho_r$, that is, $U_r = \cup_K \rho_r \ (k; data_{k,r})$. With different initialization points, $SVM_3$ optimization can be initialized by merging $RVM_1$ and $RVM_2$ results. In every stage, optimization is tried to advance as much as possible for producing better results. Initial data splitting, results merging and how well optimization is initialized from partial results given by previous stages defines this. Global optimum convergence is guaranteed by this technique.

### 3.3. Feature Selection

Samples are parallelized using mapping concept. Then they are allowed for selection of features for reducing attributes. Feature selection process is done using filter based feature selection. In Filter technique, ranking methods are used as a principle parameter. Using a suitable ranking criterion, a score is assigned to variables and removed the variables having a score less then threshold value. These techniques are cheaper in computation wise and over fitting is avoided but, dependencies between features are avoided in Filter techniques. So, selected subset may not be optimum and there is a need to find redundant subset. Following gives the basic filter selection algorithms [23].

- **Chi-Square Test**

Independence between two events are checked using chi-squared filter technique. If P(XY) = P(X)P(Y) or equivalently P(X/Y) = P(X) and P(Y/X) = P(Y), then two events X, Y are defined as independent.

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \qquad (5)$$

Need of elimination of null hypothesis (H0) of independence is indicated using High scores on $X^2$ in expression (1). There is a dependency between class and term occurrence.

- **Euclidean Distance**

With respect to Euclidean distance, computed the correlation between features in this technique of feature selection. For example with n features in a sampled feature called 'a', these 'n' number of features are compared with other 'n-1' features via computing distance between them using following expression (6). Inclusion of new features will not affect the distance between features.

$$d(a,b) = \{\Sigma_i (a_i - b_i)^2\}^{1/2} \qquad (6)$$

- **Correlation Criteria**

A simple criteria called Pearson correlation coefficient is used and is defined as in expression (7). Where, $i^{th}$ variable is represented as $x_i$, output class is represented as Y, variance is represented as var() and covariance is represented as cov(). Linear dependencies between target and variable can only be detected using this correlation ranking. This is a major disadvantage of it.

$$R(i) = \frac{cov(x_i, Y)}{\sqrt{var(x_i) * var(Y)}} \qquad (7)$$

- **Information Gain**

Importance of specified feature vector attributes are indicated using information gain. Terms with the highest information gain scores are selected by IG feature selection techniques. About class prediction, information amount is measured using information gain, if there exist only feature presence information and its class distribution. Expected minimization in entropy- uncertainty associated with a random feature- is measured as,

$$\text{Entropy} = \sum_{i=1}^{n} -p_i \log_2 p_i \quad (8)$$

Where, classes count is represented as 'n' and S's probability belonging to class 'i' is represented as Pi. The A and S's gain is computed as,

$$\text{Gain (A)} = \text{Entropy(S)} - \sum_{k=1}^{n_1} \frac{S_k}{S} * Entropy(S_k) \quad (9)$$

Where, subset of S is given by Sk.

- **Mutual Information (MI)**

Dependency measure between two variables is used in an information theoretic ranking criterion. Shannon's definition of entropy is specified first for describing MI.

$$H(X) = -\sum_i P(y) \log P(y) \quad (10)$$

In output Y, uncertainty which is information content is represented using above expression. If a variable X is observed by it, then conditional entropy is expressed as,

$$H(Y/X) = -\sum_i \sum_y P(x, y) \log P\left(\frac{y}{x}\right) \quad (11)$$

In output Y, uncertainty can be reduced by observing a variable X as indicated in above expression. In uncertainty, decrease is expressed as,

$$I(Y, X) = H(Y) - H(Y|X) \quad (12)$$

Between Y and X, MI is specified using this and if X and Y are independent, then MI is zero and if they are dependent, it will be higher than zero. It indicates that, information about one variable can be provided using another variable, which proves dependency. For discrete variables, given the above defined definitions and for continuous variable, the same can be obtained using replacement of summation by integration.

- **Correlation Based Feature Selection (CFS)**

A heuristic is used for selecting attributes in Correlation-based Feature Selection algorithm. Individual features usefulness are measured using this to predict class label along with inter-correlation level between them. It avoids highly irrelevant an correlated features. Expression used for filtering out redundant, irrelevant features which produces poor class prediction is given by,

$$F_s = \frac{N * r_{ci}}{N + N(N-1) r_{ii}} \quad (13)$$

- **Fast Correlation Based Feature Selection**

A class of multivariate feature selection technique is FCBF (Fast Correlation Based Filter) [24]. With entire features set, it is initialized and feature dependences are computed using a symmetrical uncertainty and with sequential search strategy, backward selection approach is used for finding finest subset. There are two stages in FCBF algorithm: Relevance analysis is done in first stage, where, based on relevance score, input variables are ordered. It calculated as symmetric uncertainty based on required output.

Irrelevant variables having ranking score less than predefined threshold are discarded using this stage. From relevant set derived from first stage, predominant features are selected using

redundancy analysis in second stage. In an iterative manner, this selection process is done, which removes those variables for forming approximate Markov blanket.

A normalized information theoretic measure is Symmetrical Uncertainty (SU), where feature dependencies are computed using entropy and conditional entropy.

Total independency between two features are indicated using a value 0 and dependency is indicated using value 1in Symmetrical Uncertainty. Using one feature value, another one can be predicted totally.

### 3.4. Outlier Selection by Fuzzy C-Means Clustering (FCM)

Computation of outliers as fuzzy c-means clustering's side-product is focused in this work. There are two stages in this proposed strategy. Purely fuzzy c-means process is done is first stage and exceptional objects are identified in second stage based on novel metric according to membership values entropy [25]. From hard c-means clustering, is separated form a fuzzy c-means clustering technique, where hard partitioning is employed in hard c-means.

Fuzzy partitioning is employed in FCM so that a data point of a every patter does not belongs to some cluster definitely. But is a member of all clusters with a membership value between 0 and 1. FCM is an iterative algorithm. Computation of cluster centres for minimization of dissimilarity function is focused in FCM. For accommodating fuzzy partitioning introduction, based on expression (14), randomly initialized the membership matrix (U).

$$\sum_{i=1}^{c} u_{ij} = 1, \forall_j = 1, \ldots\ldots, n \qquad (14)$$

The dissimilarity function utilized in FCM is expressed as,

$$J(U, c_1, c_2, \ldots\ldots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} \qquad (15)$$

Where, $u_{ij}$ is lies between 0 and 1; cluster i's centroid is represented as $c_i$, between $i^{th}$ centroid($c_i$) and $j^{th}$ data point, Euclidian distance is given by $d_{ij}$, weighting exponent is represented as $m \in [1, \infty]$. Two conditions needs to satisfied for reaching a minimized dissimilarity function. They are expressed in expression (16) and (17).

$$c_1 = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \qquad (16)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \qquad (17)$$

Following steps are determined in this algorithm

**Step 1.** Randomly initialize the membership matrix (U) that has constraints in Equation (14).

**Step 2.** Calculate centroids(ci) by using Equation (16)

**Step 3.** Compute dissimilarity between centroids and data points using equation (15). Stop if its improvement over previous iteration is less than threshold.

**Step 4.** Move to step 2. Within a dataset, cluster centres are moved to "right" location by FCM iteratively via cluster centres iterative update and membership grades for every data point.

Major objective of this proposed fuzzy c-means based outlier detection(FCM-OD) algorithm is producing outliers via FCM output post-processing. There are two consecutive stages. In first stage, until convergence fuzzy c-means is performed purely and in second stage, for every vector, outlier factor is assigned. Outlier factor based on its membership grades of every clusters.

By intuition, there will be equal outlier's membership grades as it does not belongs to some specified cluster definitely. In can also be stated as, membership variable's uncertainty of this vector is very high. From information theory viewpoint, entropy is an information measure and random variable uncertainty. In general, random variable is *X*, set values that are taken by *X* is represented as *S* (*X*) and X's probability function is represented as *p* (*x*), entropy E (X) is defined as,

$$E(X) = -\sum_{x \in S(X)} p(x) \log (p(x)) \qquad (18)$$

Hence vector $x_j$'s outlier function is defined as follows,

$$OF(x_j) = \sum_{k=1}^{c} |-u_{ij} log u_{ij}) / \log c \qquad (19)$$

Where, cluster count are represented as c, $x_j$'s membership grade to $i^{th}$ cluster is represented as $u_{ij}$. As shown in expression (19), large values will make vector as outlier. In addition, normalization of outlier factors to scale [0,1] is done, because maximum value so called log c is attained by $(\sum_{k=1}^{c} |-u_{ij} log u_{ij})$, if all $u_{ij}$ are same like 1/c through a classical theorem.

From implementation viewpoint, $(\sum_{k=1}^{c} |-u_{ij} log u_{ij})$ needs to be computed for computing vector's outlier factor because for a specified running process, there is a constant value of log c. That is concerned with membership variable entropy computation.

Algorithm 1 shows FCM-OD algorithm according to above concepts. The FCM algorithm's computational complexity is given by *O* (*tcnd*) operations, where, iterations count is given by t, clusters count is given by c, features count is given by d and objects count is given by n. A *O* (*nlogn*) operations are needed to sort *n* outlier factors using quick sort. So, FCM-OD algorithm is fast, as its complexity is *O* (*tcnd*+ *nlogn*).

**Algorithm 1. FCM-OD**

**Input:** Features
**Output :** Top p outliers
**Steps:**
1. Run FCM on D until converge to get the membership matrix;
2. For each vector $x_i$ in D do;
3. Compute the outlier factor $x_i$;
4. End for;
5. Sort the outlier factors and return top p outliers

### 3.5. Virtual pair selection by ACO

- **ANT Colony Optimization**

Ant colony optimization algorithm is used for selecting virtual pair automatically for solution optimization. Lot of attention have been attracted by techniques based on Ant Colony Optimization (ACO). Previous iteration knowledge are applied in these techniques for producing better solutions.

A type of a metaheuristic is ant colony optimization, where for a difficult discrete optimization problems, optimum solutions are computed using a cooperation between colony of artificial ants. For a specified combinatorial optimization problem (COP), adequate model is defined in ACO's application to its solution. The COP model (S, Ʊ, f) can be formulated as a problem having search space of S, constraints set Ʊ and objective function f, which is used for minimizing computation cost in computing optimum solution from a set of possible solution.

Parameters are set in ant colony optimization metaheuristic. Pheromone trails are initialized for constructing Ant Solutions Apply Local Search [optional] Update Pheromones. Meta-heuristic iterates over three phases, if termination condition is not satisfied after initialization. Ants construct number of solutions in every iteration. Then using a local search, these solutions are improved and this is not a mandate step. At last updated the pheromones.

- **ACO Algorithm**

A solution for a specified problem can be computed using single real or an artificial ant. But, good solution can be computed using cooperation among different individuals using stimergy as computed in real ants world. While walking along problem space, pheromone are deposited by ants as stated already. In artificial ants which lives in virtual world, modification of numeric values which are the pheromone having association with every problem state are done.

Implemented a mechanism as like in physical pheromone evaporation in real ant colonies. This makes the artificial ants to focus on novel promising directions for search and allows ants to forget about its past history. Agents early convergence as a sub-optimum paths are avoided using this. There are major differences between artificial and real ants. They are,

1) In a real world which is characterized using environment where most times are stochastic and probabilistic, natural ants lives and in a discrete world, artificial ants lives. Through a finite set of problem state, they used to move sequentially.

2) Major difference lies in update of pheromone evaporation and deposition. On the ground, all ants deposits pheromone as they walk along paths in a real world problem space. Pheromone value update is not done by all ants in every case in a computational problem space (graph). After constructing a solution, update is done.

3) For achieving system efficiency, additional mechanism are used in some artificial ants implementation and this won't be available in real ants like backtracking [26].

Following are the major (ACO) algorithms steps [27]:

1. Initialization of Pheromone trail.

2. Pheromone trail based solution construction: Based on probabilistic model, an entire solution is constructed for a problem by every ant.

3. Evaluation of solution: According to problem specific fitness function, solution's quality are evaluated.

4. Update of pheromone trail: In two phases, this is applied. They are evaporation and reinforcement. A quantity of pheromone is deposited by every ant based on its solution's fitness

in reinforcement phase. For avoiding stagnation, fraction of pheromone are evaporated in evaporation phase.

Following aspects are specified in ant system design,

- An environment for representing problem domain as a graph is specified as mentioned in figure 1. For the problem, it is suitable for ants for constructing and navigating a solution using this.
- A heuristic evaluation function (η) based problem is specified, which specifies a quality factor of various solution construction steps.
- A pheromone updating (τ) rule, which considers trails evaporation and reinforcement.
- Heuristic function (η) and pheromone trail (τ) strength based probabilistic transition rule that is used for iteratively construct a solution. Constructed solution is evaluated using a fitness function.

***Rank-Based Ant System***

A variant of the ACO algorithm called the $AS_{elitist}$ [21](Ant System with elitist strategy) achieved improved results compared to the ones achieved by the AS. In elitist strategy, more emphasis is given to the best solutions constructed to guide successive solution constructions. The fittest (best) solutions are preserved is the hallmark of the elitist strategy, this gives rise the problem of placing more importance on local search aspects over the global search aspects; a massive disadvantage. The Rank-Based ($AS_{Rank}$) technique therefore exploits $AS_{elitist}$ strategy's success over the conventional AS algorithm to further improve computational performance. After solutions have been constructed by each ant, based on length of ants completed tours, they are sorted. Trail update is now done using rank (μ) of every ant; thereafter only a number of elitist ants are considered.

The $AS_{Rank}$ algorithm's pheromone update is given by:

$$T(x,y) \leftarrow (1-\rho).\tau(x,y)+\sum_{\mu=1}\Delta\tau^{\mu}(x,y)+\Delta\tau^{best}(x,y) \quad (20)$$

$$\Delta\tau^{\mu}(x,y) \leftarrow (\sigma-\mu)Q/L\mu \quad (21)$$

$$\Delta\tau^{best}(x,y) \leftarrow \sigma(Q/L_{best}) \quad (22)$$

Where

σ represents elitist ants count

μ represents ranking index

On edge (i, j), trail level increase used by μ-th best ant in its tour is represented as $\Delta\mu^{\tau}(x, y)$. If edge (x,y) is not passed by μ-th best ant, then its trail level increase will be zero.

On edge (i, j), pheromone quantity laid by elite ants in its tour is represented as $\Delta\tau^{best}(x, y)$. If edge (x,y) is not a part of computed best solution, then its value evaluated to zero.

$L_{\mu}$ represents tour length completed by μ-th best ant

$L_{best}$ represents tour length completed by "best" ant.

A cost effective technique is local search, which is used in problems of optimization. After extracting iteration's best rule, performed a local search on it. In rule premises every attribute is removed for producing highly general accurate rule and its quality is tested for evaluation.

If quality of original rule is less than new rule, new rule is considered instead on original one and until modified rule producing less quantity than original rule, attributes are removed from it.

B Rule=Iteration Best Generated Rule

For each Attribute Att in Brule

N Rule=B Rule. Remove (Att)

IF N Rule. Quality>=B Rule. Quality

B Rule=N Rule

End IF

End For each

A clear specification of algorithm's convergence to a solution. Let see the comparison result on various classifiers in terms of accuracy, recall and precision in detailed manner from the following section.

### 3.6. Classification by Stacking Ensemble Learning (SEL)

Classifiers parallel combination produces stacking, where parallel execution of all the classifiers are done and at meta level, learning is performed. For a specified problem, which algorithm or model produces better performance at Meta level are computed [28]. Valuable information given by other classifiers are not considered, if best classifier is selected from base level classifier.

On a single dataset, multiple classifiers generated using various learning algorithms L1…Ln are combined using a process called stacking. In initial stage, generated a set of base level classifiers C1, C2…Cn. By combining base level classifier, developed a meta level classifier in second phase.

- **Mathematical insight into stacking ensemble**

With independent base model errors and with ensemble having M base models of an error rate $e < ½$, probability that ensemble makes an error is probability that more than M/2 base models misclassifying example. Basic idea of stacking is that, if input–output pair (x, y) is left out in hi's training set, after completing hi's training, model's error can be computed using output y.

If (x,y) is not included in hi's training set, then hi(x) will deviate from desired output y. So, for estimating this discrepancy, novel classifier can be trained and specified as y – hi(x). Error madein first classifier are learned using training of second classifier in addition. An enhanced final classification decision are computed by including this estimated errors to firsy classifier's output.

- **Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS)**

A new class of Adaptive Neuro-fuzzy System [29] is introduced in this work, which is represented as IANFIS - Improved Adaptive Neuro-fuszzy Inference System. ANFIS training error are inserted in this system's third layer for realizing this structure. ANFIS robustness and convergence capability are enhanced using training error recurrence.

Non-linear functions are identified by applying this proposed EANFIS system and comparison of results are made between proposed system and usual ANFIS system for validating proposed adaptive neurofuzzy system's effectiveness.

Layers of ANFIS and EANFIS are similar. The major difference is that the inclusion of training error $e_y$ in third layer. First order Sugeno system's fuzzy system parameters are optimized using a hybrid learning rule by ANFIS. For a two-input first-order Sugeno fuzzy model, proposed IANFIS system architecture has two rules.

In every respective layer, nodes output is represented as $O_i$, where, layer l's $i^{th}$ node is represented as i. A two inputs first-order Sugeno system's layer by layer description is given below. Input variables are fuzzified using first layer. Membership grades are generated using this.

$$O_i^1 = g(x) \tag{23}$$

Where, neuro-fuzzy system's membership function is represented as g, trapezoidal function is selected in this work. Firing strengths are generated using second layer.

$$O_i^2 = w_i = \pi_{j=1}^{m} g(x) \tag{24}$$

Firing strengths are normalized using third layer.

$$O_i^3 = \overline{w_i}' = e_y + \frac{w_i}{w_1 + w_2} \tag{25}$$

In ANFIS architecture, modification is done in this layer. Sum of normalized weight $w_i$ including training error $e_y$ is assumed as a new normalized weight, $\overline{w_i}'$. Difference between desired output $y_d$ and EANFIS overall output y defines training error.

$$e_y = y - y_d \tag{26}$$

According to consequent parameters, rule outputs are computed in layer 4.

$$O_i^4 = y_i = \overline{w_i}'.f_i = \overline{w_i}'(p_i.x + q_i.y + r_i) \tag{27}$$

All the inputs from layer 4 are added in layer 5. This is a EANFIS system's overall output.

$$y = O_i^5 = \sum_i y_i = \sum_i \overline{w_i}'.f_i = \sum_i \overline{w_i}'(p_i x + q_i.y + r_i) \tag{28}$$

$$= \sum_i (e_y + \overline{w_i}')(p_i x + q_i.y + r_i) \tag{29}$$

- **Error optimization using Gaussian Divergence function**

As in usual ANFIS, there are two steps in learning procedure. In first step, propagation of input pattern are done and using iterative least mean square procedure, estimated the optimum consequent parameters. In second step, propagation of patterns are done again and local parameters are modified using back-propagation. Until satisfying error criterion, this procedure is iterated. Under fixed local parameters, identified consequent parameters are optimum. The last layer's structure is given by,

$$Y = X * W \tag{30}$$

Where, predictors vector is represented as X and regression parameter vector is represented as W, which needs to be computed.

Output error $e_y$'s gradient sum is used in EANFIS system for making EANFIS system's local parameters correction ( values having every membership functions $A_1$, $A_2$, $B_1$, $B_2$). Gaussian Divergence function technique is used for updating local parameters and back-propagating the error signal. As in ANFIS system, it has expression for first membership function's first local parameter's modification.

$$a_i(t+1) = a_i(t) - \frac{h}{p} \cdot \frac{\partial \varepsilon_y}{a_i} \quad (31)$$

Where, for local parameter $a_i$, h is a learning rate. Partial derivatives are computed using following rule, which are used to update membership function parameters.

$$\frac{\partial E}{\partial a_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_i} \cdot \frac{\partial w_i}{\partial g} \cdot \frac{\partial g}{\partial a_i} \quad (32)$$

- **Modified Convolutional Neural Network (MCNN)**

A fully connected layer, pooled layer and convolution layers are there in CNNs. In CNN, key part is convolutional layer [30]. From input feature or image maps, features are extracted using this layer function. There will be multiple convolution kernels in every convolutional layer, multiple feature maps are computed using this. In the following manner, convolution layer is computed.

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (33)$$

Where, previous layer output's characteristic map is represented as $x_i^{l-1}$, jth convolution layer's ith channel output is represented as $x_j^l$ and activation function is represented as f(.) . Input feature maps subset is represented as $M_j$ and it is used for computing $u_j^l$, convolution kernel is represented as $k_{ij}^l$ and respective offset is represented as $b_j^l$ .

Between two convolutional layers, pooling layer is sandwiched in general. Feature map dimension minimization is a major function of this layer and to some extent, features scale invariance are maintained. There are two major pooling techniques, namely, max pooling and mean pooling. Convolution and pooling process are similar, where sliding window is involved in pooling process with simple computation and a filter is used in convolution process.

In an area, average value is used as pooled area value in mean pooling. Data background is preserved well using this technique. Data's maximum value is assumed as an area's pooled value in max pooling and data texture's are preserved well using this. Multiple data computed after passing data through different pooling and convolution layers are integrated using fully connected layer function for computing high-layer semantic features to use in classification.

Local feature filtering and parameter sharing are used for characterizing CNNs. Fully-connected parameter matrices are not used. Along the frequency bands, locality are captured using local filters. For normalizing spectral variations, a max pooling layer is added on top of convolution layer. This makes CNN hidden activations are invariant to various data types and produces better representations of features.

The CNN architecture which is having slight difference from existing proposals are shown in Figure 2. Along frequency, filters are considered in this convolution layer. Modelling of time variability is assumed. Acoustic feature like log filter bank's N neighbouring frames are given as an input to CNNs, where every frame $v_i$ is a 1D feature map.

From this layer, hidden outputs has J vectors ([h1, h2, h3 …]). Output feature map $h_j$ and input feature map $v_i$ are connected using a trainable 1D filter $r_{ji}$ and along $v_i$, across frequency axis, it is shared. From this convolution layer, output is computed as,

$$h_j = \sigma(\sum_{i=1}^{N} r_{ij} * v_i + b_j) \qquad (34)$$

Where, 1D discrete convolution operator is represented as *, trainable bias attached to hj is represented as bj. Logistic sigmoid activation function σ is used in this work.

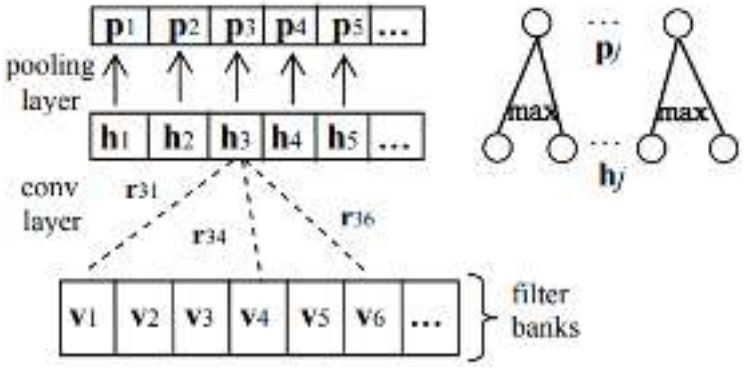


Figure 2. One stage of CNNs consisting of the convolution and pooling layers.

Then, on top of convolution layer, added a max-pooling layer. In a vector-wise mode, max-pooling is performed. In general, for every vector hj, non-overlapping groups are formed by dividing its units and within every group, maximum activation is produced as an output. With k pooling size, every feature maps pj's size after pooling is 1/k of its before-pooling hj's size. Convolution stage is formed by combining pooling and convolution layers.

In this model, two such stages are stacked in CNN, where output of lower pooling layer is propagated to higher convolution layer. Over these two stages, added the softmax layer and multiple fully-connected DNN layers. Invariant features are extracted by training pooling and convolution layers in this structure as from feature learning perspective, where, these high-level features are used by fully-connected layers.

- **Scarce population based Feature Extraction for MCNN**

Traditional CNN's linear projection using Scarce population are proposed in this work as a replacement for highly complex feature extractor and it is a simple extractor. Figure 3(a) shows the maxout layer example, where, size of a group which is a units count in every group equals 3. At every hidden layer, groups are formed by partitioning units and on every group, max-pooling is imposed. From any maxout layers, generated the Scarce representations using a non-maximum masking operation as shown in Figure 3(b).

In specific, for specified input frame, within every group, all units have its own outputs. They are not pooled together as a one output. However, in this group retained only highest value in this group, where, other outputs are made to 0. In feature extraction stage, non-maximum masking is only performed.

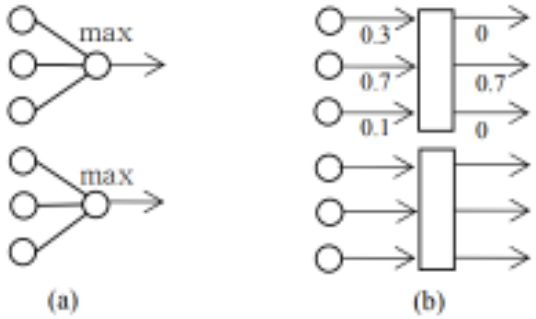


Figure 3. An example for (a) maxout layer and (b) Scarce feature generation with non-maximum masking.

The training stage always applies max-pooling. First, compute Scarce population of every feature type as a quantitative indicator. If feature vector fm is there in m-th frame, then Scarce population.

$$Scar\square e_p = \left\| \frac{f_m}{f_{m_2}} \right\| 1 \qquad (35)$$

Over entire target-language training set, this metric value's average is reported as an average. Using this work, Scarce population is shortened as $Scarce_p$. Higher features scarce can be represented using lower $Scarce_p$. So, from a proposed extractor, there will be a natural production of high- scarce features due to enforcement of 0 value to most of the hidden outputs. The major difference is that, maxout layers are used for replacing fully-connected layers in CNNs. Better classification can be achieved using this combined feature extractor as proven experimentally.

## 3. Results and Discussion

Diagnostic Breast Cancer (WDBC) dataset [31] is used for validating proposed framework's effectiveness. All stacking ensemble classifiers incorporate a modification in order to perform a fair comparison and all techniques are implemented using same MATLAB quadratic programming solver. On a 3.40-GHz Intel Core i7–3770 CPU having 16-GB RAM running Windows Server 2008, in MATLAB R2011a, executed these codes.

- **Datasets**

Two different datasets are considered in this work. As mentioned above, dataset 1 is assumed as WDBC in proposed system. After removing missing values, there are 683 instances. There are nine attributes in every instance. They are, mitoses, normal nucleoli, bland chromatin, bare nuclei, single epithelial cell size, marginal adhesion, uniformity of cell shape, uniformity of cell size and clump thickness.

Every attributes are numeric and normalized to a value between 0 to 1. Prediction of tumor as a benign (1) or malignant (2) is done using classification task. Consultation from a senior medical doctor is used for computing monotonic attributes.

The 600 loan applications given by a local bank in Taiwan is assumed as a PD600 dataset. Data collection duration is 2001 to 2002. There are 17 variables in every applications, which includes, operations, credit ($NT), use of revolving interest, number of times record checked, record of payment, monthly balance ($NT), real income, family knows about the loan

application, monthly salary ($NT), real estate, company type, education, years in the current job, work location, marital status, applicant's sex, age.

According to literature review and materials related to Joint Credit Information Center Certification, selected the following variables after interviewing bank management: tax withholding voucher, household registration certification, wage transfer accounts, borrower's bankbook and data on a personal credit application form. Identification of loan applicants who will fulfil their loan obligation (non-default: 2) and those who fail to do so (default: 1) is a major task.

- **Performance Measures**

Proposed MP-SEL algorithm and other support vector classifiers are compared in this study. With respect to F-measure, precision, recall and accuracy, their performances are examined.

Actual positives proportion that are identified correctly defined recall measures and is defined as,

$$Recall = \frac{TP}{TP + FN} \tag{36}$$

Test instances proportion having positive predictive outcomes which are predicted correctly defines precision rate. It plays major role in measuring predictive techniques and probability that a positive test reflects the underlying condition being tested for is reflected by this. It is given by,

$$Precision = \frac{TP}{TP + FP} \tag{37}$$

Recall-sensitivity- and precision [positive predictive value (PPV)]'s harmonic mean defines F-measure and is given by,

$$F - measure = 2 \cdot \frac{P.R}{P + R} \tag{38}$$

For avoiding a condition having low recall and high precision or vice versa, both recall and precision are considered in F-measure. Various algorithm's performance are compared using proposed system using this way. High intuitive measurement parameter is accuracy, where predictive ability is directly defined by this according to tested data proportion which are classified correctly and is defined as,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{39}$$

Where, in data mining, commonly used values for computing technique's predictive power are FN (False Negatives), TN (True Negatives), FP (False Positives), TP (True Positives) and are defined as,

True Positives (TP) = instances count having positive outcomes that are classified correctly.

False Positives (FP) = instances count having positive outcomes that are classified wrongly.

True Negative (TN) = instances count having negative outcomes that are classified correctly.

False Negative (FN) = instances count having negative outcomes that are classified wrongly.

***For Dataset 1:***

Table 1: Comparison of Various Techniques for Datamining Consists of 200 Samples for Dataset 1

| Techniques/Matrices | MRRMC-FSVM | HMRRMC-FSVM-ABC | MR-CFRVM-ACO | MR-SEL |
|---|---|---|---|---|
| Accuracy | 94.7 | 95.30 | 96.45 | 97.89 |
| Recall | 89.93 | 92.70 | 95.48 | 96.24 |
| Precision | 93.5 | 95.70 | 97.64 | 98.55 |

Table 1 shows comparison of various techniques for 200 samples of dataset 1 (WDBC). The classifiers used for the comparison are MRRMC-FSVM, HMRRMC-FSVM-ABC, and MR-CFRVM-ACO. It shows that the proposed MR-SEL has higher accuracy, recall and precision as 97.89%, 96.24% and 98.55% respectively at the range of 200 samples.

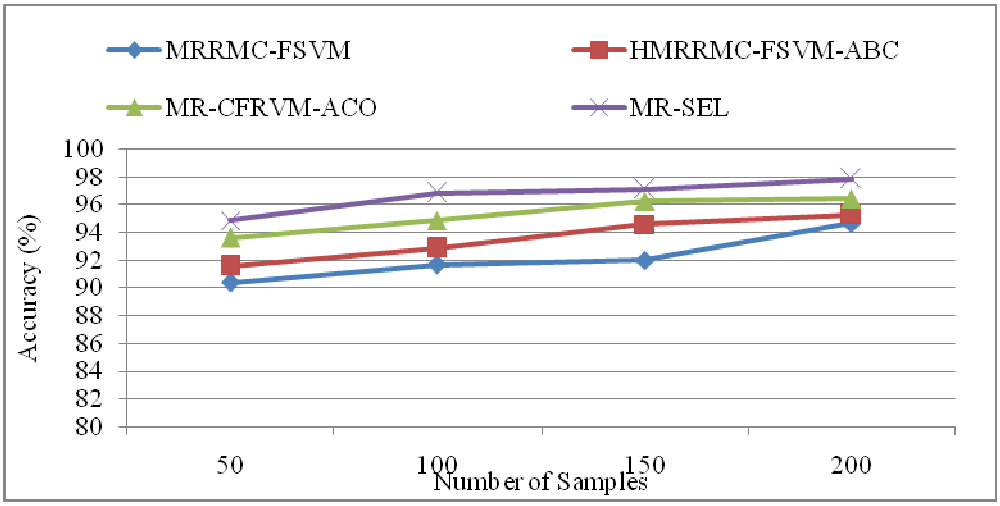


Figure 4: Accuracy Comparisons on Various Classifiers

Figure 4 shows accuracy comparison on various techniques and have proven that the proposed method has high accuracy value as 94.87% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC and MR-CFRVM-ACO have accuracy as 90.4%, 91.60%, and 93.60% respectively.

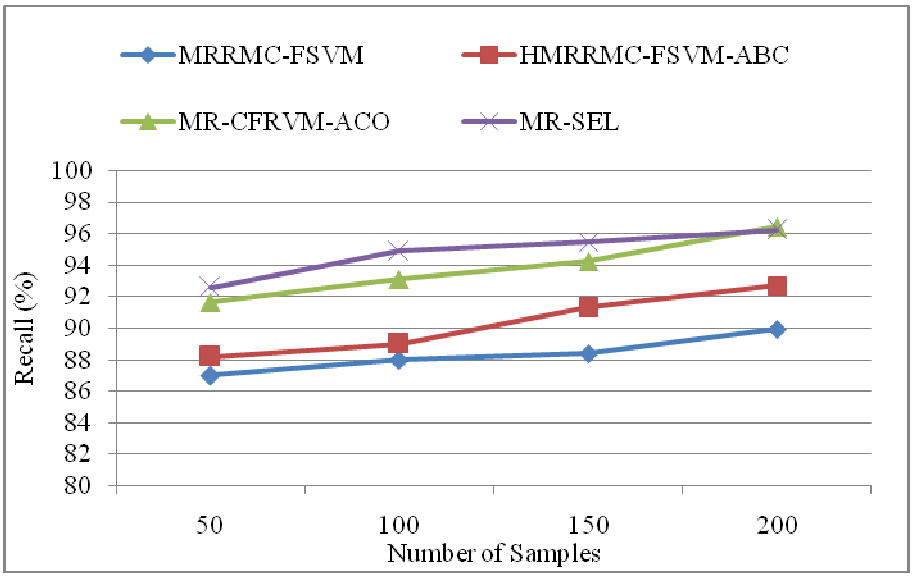


Figure 5: Recall Comparisons on Various Classifiers

Figure 5 shows recall comparison on various techniques and have proven that the proposed method has high recall value as 92.58% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC and MR-CFRVM-ACO have recall as 87%, 88.20%, and 91.64% respectively.

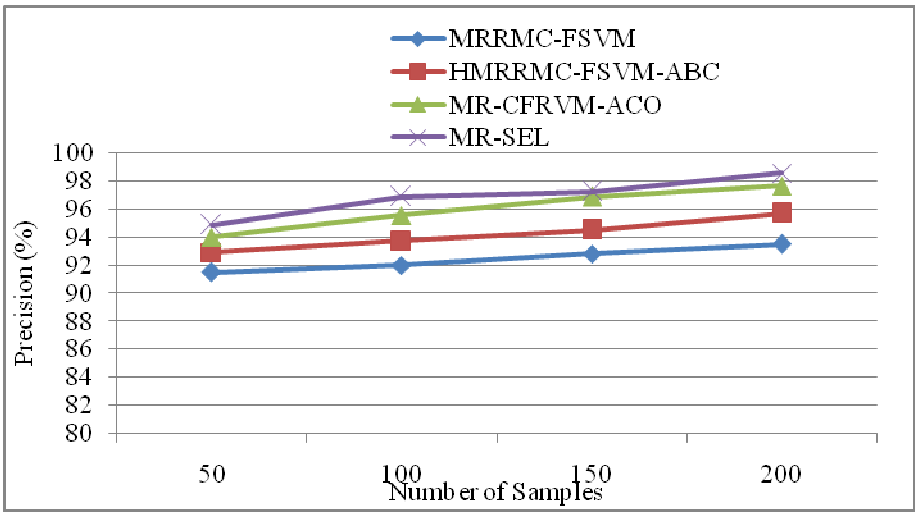


Figure 6: Precision Comparisons on Various Classifiers

Figure 6 shows precision comparison on various techniques and have proven that the proposed method has high precision value as 94.85% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC, and MR-CFRVM-ACO have precision as 91.50%, 92.90%, and 94% respectively.

***For Dataset 2:***

Table 2: Comparison of Various Techniques for Datamining Consists of 200 Samples for Dataset 2

| **Techniques/Matrices** | **MRRMC-FSVM** | **HMRRMC-FSVM-ABC** | **MR-CFRVM-ACO** | **MR-SEL** |
|---|---|---|---|---|
| Accuracy | 75.75 | 86.90 | 95.20 | 96.35 |
| Recall | 78.25 | 87.28 | 95.45 | 96.87 |
| Precision | 74.25 | 85.48 | 93.43 | 94.58 |

Table 2 shows the comparison of various techniques for 200 samples of dataset 2 (PD600). The classifiers used for the comparison are MRRMC-FSVM, and MR-CFRVM-ACO. It shows that the proposed MR-SEL has higher accuracy, recall and precision as 96.35%, 96.87% and 94.58% respectively at the range of 200 users.

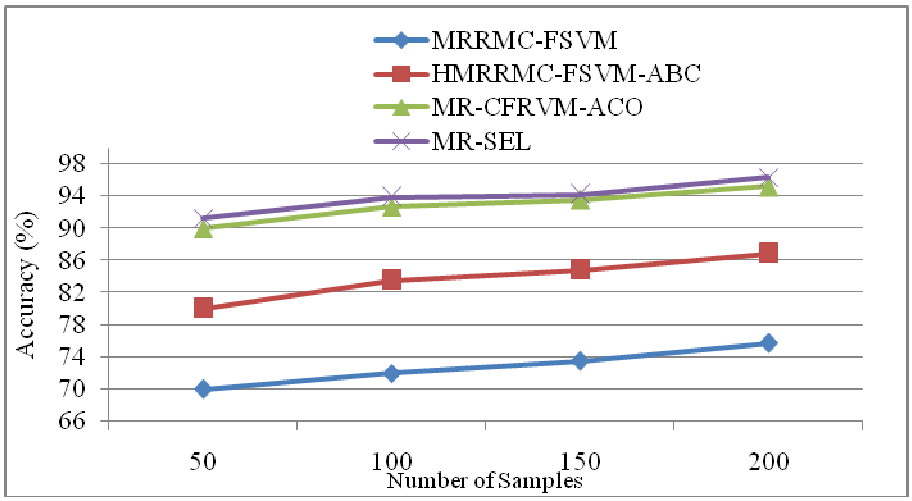


Figure 7: Accuracy Comparisons on Various Classifiers

Figure 7 shows accuracy comparison on various techniques and have proven that the proposed method has high accuracy value as 91.24% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC , and MR-CFRVM-ACO have accuracy as 70%, 80%, and 90% respectively.

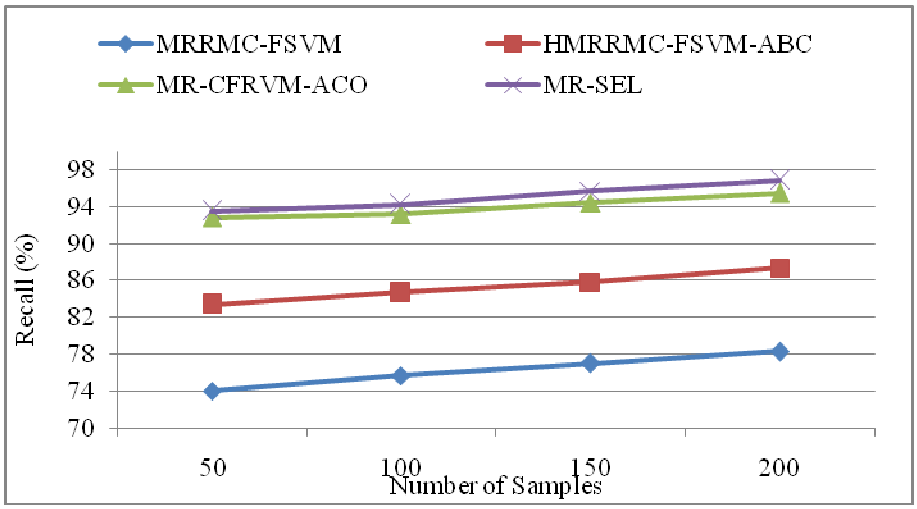


Figure 8: Recall Comparisons on Various Classifiers

Figure 8 shows recall comparison on various techniques and have proven that the proposed method has high recall value as 93.57% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC, and MR-CFRVM-ACO have recall as 74%, 83.33%, and 92.86% respectively.

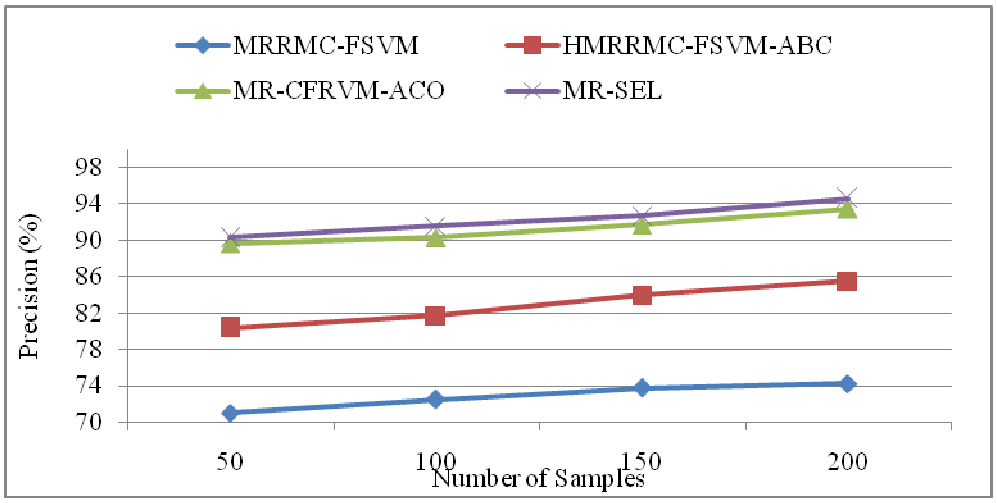


Figure 9: Precision Comparisons on Various Classifiers

Figure 9 shows precision comparison on various techniques and have proven that the proposed method has high precision value as 90.24% under the consideration of 50 users. Similarly for other existing classifiers MRRMC-FSVM, HMRRMC-FSVM-ABC, and MR-CFRVM-ACO have precision as 71%, 80.36%, and 89.66% respectively.

## 4. Conclusion

Monotonic prior domain knowledge is considered in proposed stacking ensemble learning. In decision justification and explanation, an important role is played by this prior domain knowledge. Moreover, with respect to monotonic prior knowledge, proposed classifier has given a better results in classification problem having monotonic prior knowledge as proven theoretically. A new classifier can be formed by combining different classifiers like Modified Convolutional Neural Network (MCNN) and Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS), which exhibites a better performance than constituent classifier. First the datasets are parallelized using a map reduced method to reduce the time consumption effectively. Feature extraction is for samples are takes place to reduce the attributes for reducing error rate and for improving accuracy. Thirdly outliers are removed by Fuzzy C means (FCM) clustering algorithm. Further in classification model, virtual pair is selected automatically by using Ant Colony Optimization (ACO) method and then Stacking Ensemble Learning (SEL) has been developed for classification reducing error rate and for improving accuracy.Gaussian divergence function training generalization capability, convergence rapidity, approximation precision enhancement are allowed by adding training error in EANFIS structure's third layer.. Performance comparison results of various classifiers under two benchmark datasets such as Wisconsin Diagnostic Breast Cancer (WDBC) and PD600.

## REFERENCES

1. Hong, T., & Han, I. (2002). Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert systems with applications*, *23*(1), 1-8.
2. Kotłowski, W., & Słowiński, R. (2008). Statistical approach to ordinal classification with monotonicity constraints. In *Preference Learning ECML/PKDD 2008 Workshop*.
3. Cano, J. R., Gutiérrez, P. A., Krawczyk, B., Woźniak, M., & García, S. (2019). Monotonic classification: an overview on algorithms, performance measures and data sets. *Neurocomputing*, *341*, 168-182.

4. Ben-David, A., Sterling, L., & Tran, T. (2009). Adding monotonicity to learning algorithms may impair their accuracy. *Expert Systems with Applications*, *36*(3), 6627-6634.
5. Daniels, H., & Feelders, A. (2000). *Combining domain knowledge and data in datamining systems*. Tilburg University.
6. Feelders, A., & Pardoel, M. (2003, August). Pruning for monotone classification trees. In *International Symposium on Intelligent Data Analysis* (pp. 1-12). Springer, Berlin, Heidelberg.
7. McKenzie, D., & Low, L. H. (1992). The construction of computerized classification systems using machine learning algorithms: An overview. *Computers in human behavior*, *8*(2-3), 155-167.
8. Zhu, H., Tsang, E. C., Wang, X. Z., & Ashfaq, R. A. R. (2017). Monotonic classification extreme learning machine. *Neurocomputing*, *225*, 205-213.
9. Alcalá-Fdez, J., Alcalá, R., González, S., Nojima, Y., & García, S. (2017). Evolutionary fuzzy rule-based methods for monotonic classification. *IEEE Transactions on Fuzzy Systems*, *25*(6), 1376-1390.
10. Gutiérrez, P. A., & García, S. (2016). Current prospects on ordinal and monotonic classification. *Progress in Artificial Intelligence*, *5*(3), 171-179.
11. Archer, N. P., & Wang, S. (1993). Learning bias in neural networks and an approach to controlling its effect in monotonic classification. *IEEE transactions on pattern analysis and machine intelligence*, *15*(9), 962-966.
12. García, J., AlBar, A. M., Aljohani, N. R., Cano, J. R., & García, S. (2016). Hyperrectangles selection for monotonic classification by using evolutionary algorithms. *International Journal of Computational Intelligence Systems*, *9*(1), 184-201.
13. Doumpos, M., & Zopounidis, C. (2009). Monotonic support vector machines for credit risk rating. *New Mathematics and Natural Computation*, *5*(03), 557-570.
14. González, S., Herrera, F., & García, S. (2015). Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity. *New Generation Computing*, *33*(4), 367-388.
15. Stiglic, G., Pernek, I., Kokol, P., & Obradovic, Z. (2012, August). Disease prediction based on prior knowledge. In *Proceedings of the ACM SIGKDD Workshop on Health Informatics, in Conjunction with 18th SIGKDD Conference on Knowledge Discovery and Data Mining*.
16. Li, S. T., & Chen, C. C. (2014). A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge. *IEEE Transactions on Fuzzy Systems*, *23*(5), 1713-1727.
17. Hu, Q., Pan, W., Song, Y., & Yu, D. (2012). Large-margin feature selection for monotonic classification. *Knowledge-Based Systems*, *31*, 8-18.
18. Rus, V., Lintean, M., & Azevedo, R. (2009). Automatic Detection of Student Mental Models during Prior Knowledge Activation in MetaTutor. *International working group on educational data mining*.
19. Chen, C. C., & Li, S. T. (2014). Credit rating with a monotonicity-constrained support vector machine model. *Expert Systems with Applications*, *41*(16), 7235-7247.
20. Pan, W., & Hu, Q. (2016). An improved feature selection algorithm for ordinal classification. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *99*(12), 2266-2274.
21. Cano, J. R., & García, S. (2017). Training set selection for monotonic ordinal classification. *Data & Knowledge Engineering*, *112*, 94-105.

22. Bartley, C., Liu, W., & Reynolds, M. (2016). A novel technique for integrating monotone domain knowledge into the random forest classifier. In *Fourteenth Australasian Data Mining Conference (AusDM 2016)* (Vol. 170).
23. Pavya, k., and srinivasan, B. 2005. Feature selection techniques in data mining: A Study.IJSDR International Journal of Scientific Development and Research. 2(6), pp. 594-598
24. Arauzo-Azofra, A., Aznarte, J.L. and Benítez, J.M., 2011. Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, *38*(7), pp.8170-8177.
25. Pal, N. R., Pal, K., Keller, J. M., & Bezdek, J. C. (2005). A possibilistic fuzzy c-means clustering algorithm. *IEEE transactions on fuzzy systems*, *13*(4), 517-530.
26. Dorigo, M. and Socha, K., 2006. An introduction to ant colony optimization. *Universit de Libre de Bruxelles, CP*, *194*(6).
27. Sameh, A. and Magdy, K., 2010. Data Mining Ant Colony for Classifiers. *International Journal of Basic & Applied Sciences*, *10*(3), pp.28-35.
28. Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, *38*(1), 223-230.
29. Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, *23*(3), 665-685.
30. Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems* (pp. 2042-2050).
31. C. L. Blake and C. J. Merz. (1998). UCI Repository of Machine Learning Databases. [Online]. Available: http://www.ics.uci. edu/mlearn/MLRepository.html