APPENDIX

To make this paper self-contained, here we briefly describe methods used in the paper for data visualization, data balancing, feature selection, and classification, as well as evaluation metrics.

Data Visualization Methods

t-distributed tochastic neighbor embedding (t-SNE) algorithm: The t-SNE method is a non-linear dimensionality reduction method, particularly well-suited for projecting high dimensional data onto low dimensional space for analysis and visualization purpose. Distinguished from other dimensionality reduction methods, the t-SNE method was designed to project high-dimensional data onto low-dimensional space with minimum structural information loss. So that the points close to each other on the low-dimensional surface represent states that are similar in the high-dimensional space [61].

As Van der Maaten and Hinton explained [16] "The similarity of datapoint x_j to datapoint x_i is the conditional probability $p_{j|i}$, that x_i would pick as its neighbor x_j if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i."

This method [61] starts with converting the high-dimensional Euclidean distance between data points (the Cartesian coordinates of each frame in the simulation) into the conditional|probability p_j i. Given x_i and x_j as two data points representing two structures in Cartesian coordinate, the probability density distribution of its neighboring data points for x_i is assumed as a Gaussian function centered at x_i with variance σ_i. The probabilityof x_j to be selected as the neighbor of x_i is a conditional probability calculated as

Equation 10:

$$p_{j|i} = \frac{\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right)}$$

Adaptive Synthetic (ADASYN) algorithm: ADASYN is a method of generating synthetic examples for minority classes using a weighted distribution as shown in (**Figure 1a**). The algorithm flowchart is described in detailin [15].

Data Balancing Methods

K-means clustering: k-means clustering performed on the majority class of the dataset yielded 743 cluster centers. The value of k is based on idea of equalizing minority class with majority class and generating 743 clusters for majority class. These points were mixed were the datapoints of the minority class to remove bias, and generate atotal of 1486 datapoints. The 24 prediction algorithms were run on this new dataset and the results were validated using 5-fold cross validation. Table shows the validation results.

Suppose we are given a dataset $X = x_1, \ldots, x_N$, $x_n \in R^d$ as stated in [62]. The M-clustering problem aims at partitioning this data set into M disjoint subsets (clusters) $C_1, \ldots, C_M$, such that a clustering criterion is optimized.

The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point x_i and the centroid m_k (cluster center) of the subset C_k which contains x_i. This criterion is called clustering error and depends on the cluster centers $m_1, \ldots, m_M$:

Equation 11:

$$E(m_1, \ldots, m_M) = \sum_{i=1}^{N}\sum_{k=1}^{M} I(x_i \in C_k)|x_i - m_k|^2$$

where I(X)=1 if X is true and 0 otherwise

SMOTE method of oversampling: The SMOTE (Synthetic Minority Over-Sampling Technique) func- tion takes the feature vectors with dimension (r,n) and the target class with dimension (r,1) as the input and returns final features vectors with dimension (r',n) and the target class with dimension (r',1) as the output [17]. The minority class was oversampled and the new dataset was run through the algorithms. The highest accuracy wasgiven by Fine Decision Tree of 95.4%.

Downsampling method: [63] The majority class dataset can be downsampled by an integer sampling factor,n. It samples the dataset by keeping the first sample and then every nth sample after that. In case of several columns in the dataset, each column will be treated as a separate sequence. After feeding the downsampled dataset into 24 algorithms, the highest accuracy was reported by Quadratic SVM of 56.4% only.y = downsample (x,n) [63] decreases the sample rate of x by keeping the first sample and then every nth sample after the first. If x is a matrix, the function treats each column as a separate sequence.

Feature Selection Methods

Chi-square tests(fscchi2): fscchi2(Tbl,ResponseVarName) ranks features (predictors) using chi-square tests [63]. The table Tbl contains predictor variables and a response variable , and ResponseVarName is the name of the response variable in Tbl. The function returns idx, which contains the indices of predictors ordered by predictor importance, meaning idx(1) is the index of the most important predictor. You can use idx to select important predictors for classification problems

As stated in [64], the χ2 test is applied to test the independence of two events, where two events A and B aredefined to be independent if P(AB) = P(A)P(B) or, equivalently, P(A B) = P(A) and P(B A) = P(B). In feature selection, the two events are occurrence of the term and occurrence of the class. We then rank terms with respectto the following quantity:

Equation 12:

$$\chi^2(D,t,c) = \sum_{e_t \in (0,1)} \sum_{e_c \in (0,1)} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

where,

N =observed frequency in D,

E=expected frequency,

e_t = 1(the document contains term t) and et = 0(the document does not contain term t),

C is a random variable that takes values ec = 1(the document is in class C) and ec = 0(the document is not in class c)

Minimum redundancy maximum relevance (MRMR) (fscmrmr): [65] This algorithm tends to select a subset of features having the most correlation with the class (output) and the least correlation between themselves. It ranks features according to the minimal-redundancy-maximal-relevance criterion which is based on mutual information. Minimal redundancy as stated in [66] will make the feature set a better representation of the entire dataset. Let S denote the subset of features we are seeking. The minimum redundancy condition is:

Equation 13:

$$minW_I, W_I = \frac{1}{|S|^2} \sum_{i,j \in S} I(i,j)$$

where, |S| is the number of features in S

Fitcensemble: Fitcensemble(Tbl,ResponseVarName) returns the trained classification ensemble model object (Mdl) that contains the results of boosting 100 classification trees and the predictor and response data in the table Tbl. ResponseVarName is the name of the response variable in Tbl [63]. By default, fitcensemble uses LogitBoost for binary classification and AdaBoostM2 for multiclass classification.

As stated in [67], let each (y, x) case in L be independently drawn from the probability distribution P. SupposeY is numerical and $\phi(x, L)$ the predictor. Then the aggregated predictor is the average over L of $\phi(x, L)$, i.e.

Equation 14:

$$\varphi A(x) = EL\varphi(x.L)$$

Fitctree: Estimate predictor importance for classification using a binary decision tree. fitctree(Tbl,VarName) returns a fitted binary classification decision tree based on the input variables (also known as predictors, features, or attributes) contained in the table Tbl and output (response or labels) contained in Tbl, VarName [63][68]. The returned binary tree splits branching nodes based on the values of a column of Tbl. We consider the problem of finding the best test for a nominal attribute with n values in a k-class L-ary decision tree as stated in [69]. We are particularly concerned with cases in which n or k, or both, are quite large; this situation arises in many pattern recognition problems and in some large real data mining applications. The problem is to find the optimal partition of n elements into L bins. A partition of the n distinct values of the attribute induces a partition of the examples: each example is put into the branch corresponding to the value that the given attribute takes for that example. The class impurity of the examples in each branch is computed, weighted, summed and assigned to the given partition. An n by k contingency matrix, computed at the start of the procedure, can be used to compute the impurity measure for each partition that is considered. The number of distinct partitions of n elements is exponential in n: for example, if L = 2, a binary tree, there are 2^{n-1}-1 two-way partitions [69].

Rank key features by class separability criteria (rankfeatures) IDX = rankfeatures(X,GROUP) ranks the features in X using an independent evaluation criterion for binary classification. X is a matrix where every column is an observed vector and the number of rows corresponds to the original number of features. GROUP contains the class labels. IDX is a list of indices to the rows of X with the most significant features

The Bhattacharyya distance [70] is a common method for measuring the separation between two multivariate gaussians. Therefore, we will have to use this method based on the assumption that data is drawn from a Gaussian distribution. Because we have five classes in our data, we first estimate a Gaussian distribution from which thedata is drawn, then we calculate all the possible combinations among the classes. Finally we add all this distances to produce a ranking for each feature. The Bhattacharyya distance for two multivariate distributions P1 and P2can be calculated as follows:

Equation 15:

$$BhatDistance(P1, P2) == (1/8)(m1m2)TP1(m1m2) + 0.5ln\,\frac{det\,P}{\sqrt{detP1detP2}}$$

Equation 16:

$$P = \frac{P1}{P1+P2}$$

The Wilcoxon signed rank test as stated in [71]is used to test that a distribution is symmetric about some hypothesized value, which is equivalent to the test for location. We illustrate with a test of a hypothesized median, which is performed as follows:

- Rank the magnitudes (absolute values) of the deviations of the observed values from the hypothesized median, adjusting for ties if they exist.
- Assign to each rank the sign (+ or -) of the deviation (thus, the name signed rank).
- Compute the sum of positive ranks, T(+), or negative ranks, T(-) , the choice depending on which is easier to calculate. The sum of T(+) and T(-) is n(n+1)/2, so either can be calculated from the other.
- Choose the smaller of T(+) and T(-), and call this T.
- Since the test statistic is the minimum of T(+) and T(-), the critical region consists of the left tail of the dis- tribution, containing a probability of at most $\alpha/2$. If n is large, the sampling distribution of T is approximately normal with

Equation 17:

$$\mu = n(n+1)/4, and \sigma 2 = n(n+1)(2n+1)/24$$

which can be used to compute a z-statistic for the hypothesis test.
In case of *entropy* criterion, atypical [72] ranking process consists of four steps:

- Initialize set F to the whole set of p features. S is an empty set.
- For all features $f \in F$ compute J(f) coefficient.
- Find feature f that maximizes J(f) and move it to $S \leftarrow S \cup \{f\}$, $F \in F\{f\}$
- Repeat until the cardinal of S is p

where J(f) is a criterion function (specific for a given algorithm) which gives a measure of dependency between fea- tures (f) and classes (C). Feature (variable) selection problems can be formulated into a [73] cardinality optimization (In general, it is NP hard)

Equation 18:

$$\beta = \underset{\beta \in \mathbb{R}^p}{argmin} Q(\beta) = \sum_{i=1}^{n} q(X_i^T \beta, y_i) s.t. \|\beta\|_0 \leq s$$

p is the number of features,
Xi, β ∈ Rp,
n is the number of training samples,
$X \in R^n \times p$, $y \in R^n$ denotes training data;
s is the given sparsity level,
Q () is convex and smooth [74].

PredictorImportance: [63] Imp = oobPermutedPredictorImportance(Mdl) returns a vector of out-of-bag, pre- dictor importance estimates by permutation using the random forest of regression trees Mdl.

A classification tree [75] is a rule for predicting the class of an object from the values of its predictor variables. The tree is constructed by recursively partitioning a learning sample of data in which the class label and the values of the predictor variables for each case are known. Each partition is represented by a node in the tree.

If X is an ordered variable, this approach searches over all possible values c for splits of the form $X \leq c$. A case is sent to the left subnode if the inequality is satisfied and to the right subnode otherwise. The values of c are usually restricted to mid-points between consecutively ordered data values. If X is a categorical predictor (i.e., a predictor variable that takes values in an unordered set), the search is over all splits of the form $X \in A$ where A isa non-empty subset of the set of values taken by X.

Predictor Importance [63] estimates predictor importance of the predictors for each tree learner in the ensemble ens and returns the weighted average imp computed using ens.TrainedWeight. The output imp has one element for each predictor. predictorImportance computes importance measures of the predictors in a tree by summing changes in the node risk due to splits on every predictor, and then dividing the sum by the total number of branch nodes. The change in the node risk is the difference between the risk for the parent node and the total risk for the two children. For example, if a tree splits a parent node (for example, node 1) into two child nodes (for example, nodes 2 and 3), then predictor Importance increases the importance of the split predictor by [63]:

Equation 19:

$$(R_1 R_2 R_3)/N_b$$

where R_i is node risk of node i, and N_b is the total number of branch nodes. A node risk is defined as a node error weighted by the node probability:

Equation 20:

$$Ri = PiEi$$

where R_i is the node probability of node i, and E_i is the mean squared error of node i.

Corrcoef: Corrcoef(A) returns the matrix of correlation coefficients for A, [63] where the columns of A represent random variables and the rows represent observations Correlation as stated in [76] is a measure of a monotonic association between 2 variables. A monotonic relationship between 2 variables is a one in which either (1) as the value of 1 variable increases, so does the value of the other variable; or (2) as the value of 1 variable increases, the other variable value decreases. In correlated data, therefore, the change in the magnitude of 1 variable is associated with a change in the magnitude of another variable, either in the same or in the opposite direction. In otherwords, higher values of 1 variable tend to be associated with either higher (positive correlation) or lower (negative correlation) values of the other variable, and vice versa. Assumptions of a Pearson correlation are as follows:

- As is actually true for any statistical inference, the data are derived from a random, or at least representative, sample. If the data are not representative of the population of interest, one cannot draw meaningful conclusionsabout that population.

- Both variables are continuous, jointly normally distributed, random variables. They follow a bivariate normal distribution in the population from which they were sampled. The bivariate normal distribution is beyond the scope of this tutorial but need not be fully understood to use a Pearson coefficient. The equation for correlationcoefficent is represented as follows [77]:

Equation 21

$$Pxy = \frac{cov(X,Y)}{\sigma x \sigma y}$$

where,
cov is the covariance,
σ_x is the standard deviation of X,
σ_y is the standard deviation of Y

Infinite Latent Feature Selection (ILFS):

Consider a training set $X = \vec{x}_1, \ldots, \vec{x}_n$, such that the distribution of the values assumed by the i^{th} features is given by m×1 vector $\vec{x}_I$, taking into account m samples. An undirected graph G is formed so that the features are represented by the nodes and the inter-node relationships are represented by the edges. If a_{ij} is an element of the adjacency matrix, A associated with G, that represents the pairwise relationship between the features x_i and x_j (1≤I,j≤n) G can be represented by the binary function [78]:

Equation 22:

$$aij = \phi(x_i, x_j)$$

where ϕ is a real valued potential function. The probability of each co-occurrence in x_i and x_j is framed as a mixture of conditionally independent multinomial distribution, where parameters are learned using Expectation Maximization(EM) algorithm.

Feature selection via Eigenvector Centrality (ECFS): The adjacency matrix of the above graph G canbe written as [21]:

Equation 23:

$$A = \alpha k + (1 - \alpha)\,\Sigma(i,j)$$

where α [0, 1] is a loading coefficient. In Eigenvector Centrality measure (EC), v_o is calculated as the eigen vectorof A associated with the largest eigen value. If e is any vector,

Equation 24:

$$\lim_{l \to L}[A^l e] = vo$$

Relieff: Relieff is an algorithm developed by Kira and Rendell [79] [80] in 1992 that uses filter-method approach for feature selection. If a dataset consists of *n* instances of *p* features, belonging to two classes. At each iteration, X is a feature vector belonging to one random instance and the feature vectors of the instances closest to X from each class using Manhattan L1 norm are chosen. 'Near hit' is the closest instance of the same class and 'Near miss' is the closest instance of different class. The weight vector is updated as follows [81]:

Equation 25:

$$W_i = W_i - (x_i - nearHit i_i)^2 - (x_i - nearMiss i_i)^2$$

Feature Selection Concave (FSV): If matrices $A \in R^{m\times n}$ and $B \in R^{K\times n}$ are two point sets, then they can be discriminated by a separating plane, P as in [22]:

Equation 26:

$$P = \{x | x \in R^n, x^T w = \gamma\}$$

where normal $w \in R^n$ and 1-norm distance to the origin is defined as $\frac{(|\gamma|)}{||w||_{inf}}$

Laplacian. A parameter used in this algorithm is the Laplacian Score(LS) which means that two pointsare related to the same topic if they are close to each other. Laplacian score of the r^{th} feature is calculated as follows [23]:

Equation 27:

$$L_r = \frac{(\widetilde{f_r})^T L \widetilde{f_r}^T}{(\widetilde{f_r})^T D \widetilde{f_r}}$$

Unsupervised Discriminative Feature Selection (UDFS): UDFS aims to select the most discriminative features for data representation, where manifold structure is considered. $X = \{x_1, x_2, ..., x_n\}$ is the training set, $x_i \in R^d (1 < i < n)$ is the i^{th} datum and n is the number of data points in the training set. The objective function of this algorithm is: For an arbitrary matrix, $A \in R^{r \times p}$, its $l_{2,1}$-norm [24] is:

Equation 28:

$$\| A \|_{2,1} = \sum_{i=1}^{R} \sqrt{\sum_{j=1}^{P} A_{ij}^2}$$

Local Learning Clustering based Feature Selection (LLCFS): This algorithm constructs the k -nearest neighbor graph in the weighted feature space. It performs joint clustering and feature weight learning [25].

Correlation based Feature Selection(CFS): This algorithm [26] performs feature selection on the basis of the hypothesis,"good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other". A merit function is a function that measures the agreement between data and the fitting model, for a particular choice of parameters. By definition, the merit function is small when the agreement is good. The merit function of a feature subset S consisting of k features is given as [82]:

Equation 29:

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

where $\overline{r_{cf}}$ is the mean of all feature-classification correlations, and
$\overline{r_{ff}}$ is the mean of all feature-feature correlations.

Classification Methods

Bayesian network: A Bayesian network is a directed acyclic graph with some quantitative probability infor- mation assigned to each node that corresponds to a random variable. It has many other synonyms, viz, belief networks, probabilistic network, causal network and knowledge map [35]. A conditional probability distribution P(xi parents(Xi)) defines the relationship between each node and its parents. It is defined by the following equa- tion [35]:

Equation 30:

$$P(x_1,\ldots\ldots,x_n) = \prod_{i=1}^{n} P\,(x_i|parents(X_i))$$

where $P(x_1,\ldots\ldots,x_n)$ = probability of joint conjunction of events x_1, x_2,...x_n.

Decision tree: The goal attribute is true if and only if the input attributes follow the paths towards a leafwith value true. This assertion gives a decision tree and its propositional logic can be written as follows [35]

Equation 31:

$$\text{Goal} \Leftrightarrow \text{Path}_1\text{V Path}_2\text{V}\ldots$$

In MATLAB definition, *fine trees* have the highest model flexibility [63] as they have many leaves to make manyfine distinctions between classes. They allow a maximum of 100 splits. In case of *medium* trees, the model flexibility is medium. They allow a maximum of 20 splits. In case of *coarse* trees, the model flexibility is low and they allowa maximum of 4 splits.

Logistic regression: The logistic function is given by the following equation [35]:

Equation 32:

$$\text{Logistic(z)} = \frac{1}{1+e^{-z}}$$

It gives the probability of belonging to the class labeled 1. The process of fitting the weights of this model tominimize loss on a data set is called logistic regression.

RUS boosted trees: Random under-sampling (RUS) is used to balance an imbalanced class that is a commonproblem for any datasets having rare occurrences of a particular event, the algorithm of which can be found in [83].

Bagged trees: "Bagging predictors [67] is a method for generating multiple versions of a predictor and using these to get an aggregated predicton"- Breiman. Consider data $\{(y_n, x_n), n = 1...,N\}$ in a learning set, where the y's are either class labels or a numerical response. If the input is x we predict y by φ(x,L), taking repeated bootstrap samples $\{L^B\}$ from L, and forming {φ(x,LB)} and if y is numerical

Equation 33:

$$\phi_B{}^{(x)} = av_b\phi(x, L^{(B)})$$

If y is a class label, let the $\{\varphi(x,L^B)\}$ vote to form $\varphi B^{(x)}$. This is called "bootstrap aggregating" or bagging.

k-means clustering: k-means clustering performed on the majority class of the dataset yielded 743 cluster centers. The value of k is based on idea of equalizing minority class with majority class and generating 743 clusters for majority class. These points were mixed were the datapoints of the minority class to remove bias, and generate a total of 1486 datapoints. The 24 prediction algorithms were run on this new dataset and the results were validated using 5-fold cross validation. Suppose we are given a dataset X = x1, ..., xN, xn ∈ R^d as stated in [62]. The M-clustering problem aims at partitioning this data set into M disjoint subsets (clusters) C1, ...,CM, such that a clustering criterion is optimized. The most widely used clustering criterion is the sum of the squared Euclidean distance between each data point xi and the centroid mk (cluster center) of the subset Ck which contains xi. This criterion is called clustering error and depends on the cluster centers m1, ...,mM:

Equation 34:

$$E(m_1,\ldots,m_M) = \Sigma_{i=1}^{N}\sum_{k=1}^{M} I\,(x_i \in C_k)|\mathrm{x}_i - \mathrm{m}_k|)^2$$

where I(X)=1 if X is true and 0 otherwise

where μ_i is the centroid of cluster S_i imbalanced dataset by increasing the number of samples of the

minority class. The algorithm flowchart is described in detail in [17].
Support Vector Machine: SVM is a type of supervised learning, where data that is not linearly separable can be easily separated by mapping them into higher dimensional space. The optimal SVM separator is found by solving the following [35] :

Equation 35:

$$\underset{a}{arg\ max} \sum_{j} \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_j y_j y_k (x_j . x_k))$$

K-nearest neighbor KNN algorithm classification is a type of clustering where the nearest k datapoints NN(k,xq) are considered. The distance metric is measured using Minkowski distance as follows [35]:

Equation 36:

$$L^p(x_j, x_q) = (\sum_i |x_{j,i} - x_{q,i}|^p)^{\frac{1}{p}}$$

Evaluation Matrices

The statistical parameters calculated are as follows [84]:

$$Accuracy = \frac{t_p + t_n}{total}$$

$$Precision = \frac{t_p}{t_p + f_p}$$

$$Recall = \frac{t_p}{t_p + f_n}$$

$$F1 - Score = \frac{2 \text{ x } Precision \text{ x } Recall}{Precision + Recall}$$