Effective diagnosis of heart disease imposed by incomplete data based on fuzzy random forest

1st Elzhan Zeinulla
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
elzhan.zeinulla@nu.edu.kz

2nd Karina Bekbayeva
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
karina.bekbayeva@nu.edu.kz

3rd Adnan Yazici
Department of Computer Science
Nazarbayev University
Nur-Sultan, Kazakhstan
adnan.yazici@nu.edu.kz

***Abstract*—This study presents data preprocessing and imputation techniques for creating a model from medical sensor data. We aim to solve the problem of creating a framework to diagnose heart diseases with an incomplete and dirty data, which is common with medical data. The medical dataset is often incomplete and dirty due to its small size, imbalance and many missing, false, inaccurate data. In this study, we utilize the synthetic minority oversampling technique with the combination of Tomek links to increase the size and eliminate the imbalance of the dataset. We performed a number of experiments and measurements on the Cleveland dataset and conducted a comparative study of various prediction models with recent algorithms in the literature. In order to process additional data from Budapest, Zurich and Basel, we apply the technique of semi-supervised pseudo-labelling, which means that the model has been trained on unlabeled data and combined with labelled data by predicting unlabeled values and making them pseudo-labelled. Then, the same algorithm that we used for Cleveland dataset was applied for the entire dataset. As the main classifier, Fuzzy Random Forest technique was implemented. The final accuracy of the approach proposed in this study is 93.4%, with the specificity and sensitivity values of 96.92% and 89.99%, respectively, which is superior to previous models included in the literature.**

***Index Terms*—Fuzzy Random Forest, Pseudo-labelling, Semi-Supervised Learning, SMOTE, Tomek, Multiple Imputation by Chained Equations (MICE), Data Preparation, Heart Disease**

I. Introduction

Heart disease is a disorder that affects the functioning of the body's blood-pump organ, including coronary artery heart disease (atherosclerosis), valvular heart disease, cardiomyopathy, heart rhythm disturbances (arrhythmias) and heart infections. These disorders lead to highly unpredictable heart attacks (myocardial infarction or MI), which occur when a blood clot appears and prevents the blood from flowing normally. However, cardiac emergency situations can be predictable through regular medical examinations. During these procedures, it is possible to identify the main non-modifiable risk factors, the major modifiable risk factors and the contributing risk factors. The main non-modifiable risk factors are those that cannot be changed, such as age, gender and heredity. The main modifiable risk factors can be treated and controlled by the patient. This group includes cigarette smoke, high blood cholesterol, high blood pressure, physical inactivity, obesity and diabetes. Other risk factors that play an important role in heart disease are stress, alcohol, birth control pills and sex hormones. All of those modifiable risk factors can be controlled by the patient or medical facilities. Therefore, it is essential to identify which factors have the greatest impact on the individual and to treat them appropriately, [1].

In recent decades, medicine has developed dramatically, but it is always difficult to anticipate a sudden disorder as a heart attack, even for highly experienced professionals. Recently, machine learning and deep learning algorithms have begun to significantly help medical doctors identify and predict dangerous diseases, including heart problems. However, one of the challenges that machine learning faces in medicine are the imbalance of data, numerous missing values and features, uncertain data, and noisy features.

The purpose of this study is to address these challenges and improve the performance of the series of the previous studies using feature engineering along with semi-supervised learning (SSL) approaches and creating new semi-synthetic data using a real-life dataset which is available for the research community. More specifically, the contributions of this article are to explore the method of solving unbalanced data by balancing minor and major classes, take the advantage of vast unlabelled data using SSL, implement Fuzzy Random Forest technique, contribute to the Basel, Budapest and Zurich heart disease dataset and make it publicly available [2], manage missing values, which is a common problem in medical data set, and thus predict anomalies, such as cardiac abnormalities, with greater accuracy compared to previous studies in the literature, [3], [4].

This study begins by exploring the related work done on the problem of predicting the presence of heart disease based on the same dataset. Of all the previous studies related to this research, our main focus is the best-performing study [4] in the literature. Then we give a detailed description of the dataset, defines the problem with missing columns and continues with the techniques that were used for preprocessing the data in Section 3. We also provide the basic background information needed on these techniques and explain how existing techniques are used with the necessary examples and tables. The SMOTE and Tomek methods are utilized in the combination to create the synthetic data and remove noise from the existing dataset. We then use the combination of the Fuzzy Random Forest model with the Semi-supervised

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

Pseudo-Labelling approach to propose an algorithm to impute the completely missing columns. We then explain the reasoning behind the choice of the Fuzzy Random Forest with the results of a comparative study with the k-fold cross-validation. We compare different machine learning algorithms, including deep learning and ensemble models, specifically for the application studied in this paper. In Section 4, we also present the results of the techniques proposed in this study and investigate the metrics of the model developed. We provide detailed tables and charts to analyze the results. Finally, in the conclusion section, we reaffirm our approach and suggest possible improvements to the study as future work.

II. Related work

The problem we present and study in this paper has already been addressed by several researchers in literature. One of the first works conducted on the basis of similar data was carried out by [5]. The authors describe a new discriminant function derived from logistic regression and compare it to the CADENZA Bayesian algorithm. The dataset used consists of 920 instances obtained from Cleveland, Hungary, VA Long Beach and Switzerland. The number of attributes is 14 and they are similar to those used in this study. As a result, the study in [5] obtains an accuracy of 77%.

Another more recent study on the same problem was carried out by [6]. The study detects coronary artery disease using a fuzzy PSO approach. This method is based on the Particle Swarm Optimization (PSO) algorithm combined with fuzzy boosting. The authors are able to achieve an accuracy of 85.76% using the En-PSO2 method on the same dataset as [5] used.

Later, this problem was addressed by [3]. The authors use an effective diagnosis of cardiac disease using sets of neural networks ensembles on the same sets of data that we also use in this study. This dataset is referred to as the "Heart Disease Data Set" of the UCI Machine Learning Repository, and all of these related studies focus only on the Cleveland dataset. The authors combine tree-independent neural networks as a whole and obtain a model with an accuracy of 89.01% using SAS software 9.1.3. They also reach 80.95% of the sensitivity and 95.91% of the specificity values, which represent the best performance among the studies conducted to date on the Cleveland dataset.

The most recent research on the same problem was conducted by [4] titled as "Diagnosis of Heart Disease using Fuzzy Resolution Mechanism." Kumar uses the fuzzy resolution mechanism implemented via MATLAB by combining an adaptive neuro-fuzzy interface system and a neural network. The method described in this study consists of five layers through which input values are passed. As a performance metric, the author reports an accuracy of 91.83%.

In order to solve the problem of the unbalanced dataset, an article was published in [7]. In this study, the SMOTE and Tomek links methods are applied together to effectively predict three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. The authors worked on medical dataset connected with Parkinson's disease and vertebral column taken from "The Data Mining Repository of University of California Irvine (UCI)". An application of the combination of SMOTE and Tomek together for data preprocessing and the treatment of the imbalance problem in the medical data has proven its effectiveness compared to the use of SMOTE alone.

III. Research methodology

This section begins with the description of the dataset. In addition, it introduces one of the semi-supervised learning methods - Pseudo-Labelling, data preprocessing techniques - Multiple Imputation by Chained Equations and the method of feature selection. After that, we present the SMOTE and Tomek links — oversampling and undersampling techniques - and the section ends with the reasons motivating the choice of the Fuzzy Random Forest classifier.

A. Dataset description

The dataset is from the UCI Machine Learning Repository, which is a publicly available database called the Heart Disease dataset, first reported in [8]. Part of the dataset is mostly labelled and its size is 303 instances, six of which contain missing values. This part was collected in V.A. Medical Center, Long Beach and Cleveland Clinic Foundation by Dr. Robert Detrano. Another part was collected at the Hungarina Institute of Cardiology by Andras Janosi and University Hospitals in Zurich by William Steinbrunn and in Basel by Matthias Pfisterer. It contains 610 instances, missing values and 3 unlabelled columns corresponding to the 11th, 12th and 13th columns, which have significant negative impact on the stage of training of the model. The number of attributes in the dataset is 75. This includes personal information such as name, age, gender, identification and social security numbers of patients, sensor values such as ECG (electrocardiogram), heart rate, blood pressure, etc. In addition to these attributes, the dataset also includes a type of chest pain, smoking habits, and several of observations during physical activity. The average age of all patients was 54, including 206 men. The dataset has categorical attributes and numerically represented values, as well as floating-point numbers. However, all published experiments only consider 13 of the attributes:

1) Age
2) Sex
3) Type of the chest pain(values from 1 to 4: typical angina, atypical angina, non-anginal pain, asymptomatic)
4) Blood pressure during rest
5) Serum cholesterol measured in mg/dl
6) Blood sugar during fast > 120 mg/dl (two values: 1 = true, 0 = false)
7) Resting electrocardiographic results
8) Maximum heart rate achieved
9) Exercise-induced angina
10) Old peak = ST depression induced by exercise relative to rest
11) The slope of the peak exercise ST segment

12) Number of major vessels (0-3) colored by fluoroscope
13) Thal(Thallium Stress Test): 3 = normal, 6 = fixed defect, 7=reversible defect

The last attribute, the 14th attribute, is the prediction attribute. It shows the presence of heart disease from 0 to 4, where 0 means absence of disease and 1 - 4 means presence of narrowing of cardiac vessels by more than 50% of the initial diameter. According to the coronary arteriograms, each patient was assigned to a number of calcified vessels. If a patent was classified as having coronary heart disease (1-4 values of the dependent variable), this meant that the stenosis of one of the four main coronary arteries was greater than 50% of the intraluminal diameter.

B. Data preprocessing

1) Missing values: Due to the inconsistency of patients health sensor records, some attributes lack the necessary data collected. Management of missing values is an important step in the processing of medical datasets. Medical data is often private and makes every part of datasets valuable. It does not allow to discard cases with missing values. There are many methods of dealing with missing values. The method we use in our study is Multivariate Imputation by Chained Equations (MICE). This technique is applicable only in the case of Missing At Random (MAR) values. This means that the probability of having a missing value depends on the observed values, [9]. Initially, MICE replaces each missing value with a "placeholder", for example, as a single imputation. Then, one by one the replaced values are set back to initial value. The next step is to use the cases where all attributes are completed by logistic regression and missing values are predicted using known variables. This regression equation gives all predictions for the given missing variable which is then imputed. This unique MICE step is called *predictive mean matching*, and is repeated for each missing variable in the dataset, [10].

The application of the MICE technique is illustrated by the following example. The random data has been generated synthetically, with artificial missing values, which can be found in Table I. This dataset includes the target variable *y* with the 20% of the missing values as well as our independent variables *x1, x2, x3* and *x4*. After applying the *predictive mean matching* on the target variable, the values for the first five iterations were imputed as shown in Table II. Taking the raw 4 with an initially missing value as an observation example, it is clear that the values of the imputation are significantly modified at each iteration = [-9, -5, 4, -6, 1]. These results cannot be interpreted as poor imputation results, since the mean predictive matching predicts a deviation of the results. The difference between the imputed values shows the *uncertainty of the imputation*. To converge these values to a fixed number, the number of iterations must be increased. As [9] proposes, the recommended number of iterations in MICE must be between 10 and 20 iterations. Here is an example of using *predictive mean matching* on a target column. In MICE, this technique is applied to each attribute, which contains some missing values.

Table I
GENERATED SYNTHETIC DATA

Y1	X1	X2	X3	X4
8	38	-3	6	1
1	50	-9	5	0
5	43	20	5	1
NA	9	13	3	0
-4	40	-9	6	0
NA	29	-6	5	1

Table II
PREDICTIVE MEAN MATCHING AFTER FIVE ITERATIONS

Y0	Y1	Y2	Y3	Y4	Y5	X1	X2	X3	X4
8	8	8	8	8	8	38	-3	6	1
1	1	1	1	1	1	50	-9	5	0
5	5	5	5	5	5	43	20	5	1
NA	-9	-5	4	-6	1	9	13	3	0
-4	-4	-4	-4	-4	-4	40	-9	6	0
NA	-3	9	0	1	-3	29	-6	5	1

2) Feature selection: To eliminate excessive and noisy features, we use the correlation matrix heatmap method. Heatmap analysis is a two-dimensional visualization method generally applied for high-dimensional data, for example the field of genetics. It makes it possible to distinguish the numerical values by color's intensity. The correlation matrix represents the strength of the relationships between features. The variables are placed symmetrically so that the correlation values mirror each other on the main diagonal. The main diagonal itself presents the correlation of each variable with itself. To construct the correlation matrix, we use the Pearson correlation coefficient that presents the relationship between features in a linear fashion. The values of coefficients vary from -1 to +1, where values closer to -1 or +1 indicate a strong linear correlation and values closer to 0 mean no relationship, [11].

C. Semi-Supervised Learning

Semi-Supervised learning trains the model by using both labelled and unlabelled features by extending an existing real dataset. In these learning techniques, the labelled data is collected by human and represents the supervised part, while the unsupervised one does not have output labels, [12]. Pseudo-Labelling is part of semi-supervised learning techniques. It provides "pseudo-labelled" values by training a model using dataset containing labelled data. Later, Pseudo-Labelled data can be combined with labelled data to obtain a larger dataset for a more accurate learning model. For the Pseudo-Labelling training model, it is possible to combine different training methods and neural network models.

D. Oversampling and undersampling techniques

1) SMOTE: The existing real-life dataset is unbalanced due to the unequal distribution of its classes. To solve the

problem of an unbalanced dataset, SMOTE (Synthetic Minority Over-sampling Technique) and Tomek links technique are applied together. The SMOTE approach is applied on over-sampled minority classes by creating synthetic examples instead of replicating or replacing already existing examples. The majority class samples remain unchanged, thus avoiding the overfitting problem, [13]. The idea is to randomly identify k-nearest neighbors belonging to the current minority class x_i. The number of nearest neighbors chosen at random $\widehat{x}_i$ varies depending on the oversampling required. A similar algorithm is repeated for each minority class to be over-sampled:

$$x_{new} = x_i + (\widehat{x}_i - x_i) \cdot \delta$$

The new sample x_{new} is obtained by interpolation using the formula above, [13], where δ is a random number between 0 and 1.

Sample application of the SMOTE technique on the sampling point is as follows. Suppose there is a point x_i with the value $(1, 3)$ and that of the k-nearest neighbor is the point with the value $(4, 6)$. In this case, $f_{1.1} = 1$; $f_{1.2} = 3$; $f_{2.1} = 4$; $f_{2.2} = 6$. Then the value of $\widehat{x}_i - x_i$ is computed as follows: $\widehat{x}_i - x_i$ = ($f_{2.1}$ - $f_{1.1}$, $f_{2.2}$ - $f_{2.1}$) = $(3, 3)$. Finally, the result is calculated as follows: $x_{new} = (1, 3) + rand(0—1) \cdot (3, 3)$.

2) Tomek links technique: Tomek links are pairs of points *x*, *y*, where *x* belongs to the minority class and *y* belongs to the majority class, which are the closest neighbors and represent opposite classes. Suppose the distance between these two points is *d(x,y)*. They are considered as a Tomek link if there is no such a point *z* that: $d(x, z) < d(x, y)$ *or* $d(y, z) < d(x, y)$. After identifying the Tomek link, it removes majority instance of that pair. This makes it possible to distinguish the boundaries between the majority classes and the minority classes, [13].

This can be illustrated in the following example. Suppose there is a majority class A with the following points = [(0,1), (3,4), (4,3), (5,5), (7,0)] and the minority class B with the points = [(9,10), (7,3), (8,7)]. Suppose the point *x* is (0,1) and the point *y* is (7,3), then to classify the pair (*x*,*y*) as that Tomek link, for each *z*, other than *x* and *y*, there should be no values satisfying one of the following conditions:

$$d((0, 1), z) < \ d((0, 1), (8, 7))$$

or

$$d((8, 7), z) < \ d((0, 1), (8, 7))$$

By comparing each z, no single point breaks this condition. Therefore, the values ((0, 1), (8,7)) are the Tomek link. If any of the two examples is a Tomek link, then there are two possible cases: one of these examples are noisy or both are located on the boundary of the classes.

E. Fuzzy Random Forest classifier

Fuzzy Random Forest is a multiple classifier system similar to the classic Random Forest, but it uses fuzzy decision trees as a classifier. Comparing with Random Forest, Fuzzy Random Forest reduces biased that can be caused by the presence of correlated features, [14]. Although decision tree techniques have proved to be interpretable, efficient and capable of dealing with large datasets, they are highly unstable when small disturbances are introduced in training datasets. For this reason, fuzzy logic has been incorporated in the decision tree construction techniques. Leveraging its intrinsic elasticity, fuzzy logic offers a solution to overcome this instability. This integration has preserved the advantages of both components: uncertainty management with the comprehensibility of linguistic variables, and popularity and easy application of decision trees. The resulting trees show an increased robustness to noise, an extended applicability to uncertain or vague contexts, and support for the comprehensibility of the tree structure, which remains the principal representation of the resulting knowledge, [15].

To measure the performance and avoid overfitting problem, we applied the k-fold cross validation method, which divides the dataset into a k number of equally sized folds. Then, the model is trained on the k-1 folds and the rest is a validation fold used to predict, [16]. The performance of the given algorithm can be measured by desired metrics, such as accuracy. We use 10 folds and each result obtained after completion is averaged.

F. Normalization

Normalization is a technique often applied as part of data preparation for machine learning. In our paper, we use Z-score Normalization (standardization). Feature standardization makes the values of each feature in the data have zero-mean (when subtracting the mean in the numerator) and unit-variance. This method is widely used for normalization in many machine learning algorithms (e.g., support vector machines, logistic regression, and artificial neural networks), [17].

IV. Experimental Results

During our research, the following machine learning algorithms were tested and compared using our dataset: C4.5, Artificial Neural Network, Naive Bayes, Catboost, Adaboost, Xgboost, Logistic Regression, Random Forest and Fuzzy Random Forest. The table of accuracy values for each model and ROC graph for the first 5 best-performed classification models are presented in Table III and Figure 1. As we can see, Fuzzy Fuzzy Random Forest gives the best accuracy on our dataset among these models. In addition, the performance gain of the random forest before and after using fuzzy logic is 1.27%. We performed a z-score test with the null hypothesis of equal accuracy and significance level with a p value of 0.05. The results of the z-test show that p is equal to 0.0102, which means that we can reject the null hypothesis. We repeated this test with the models: FRF vs DANN + LightGBM and FRF vs XGBoost. The calculated p values are equal to 0.0031 and 0.0002, respectively. All these results confirm that the improvement of our approach in terms of accuracy is significant. This is why we have chosen Fuzzy Random Forest as the main classifier for our application.

Table III
COMPARATIVE STUDY OF ML ALGORITHMS

Algorithm	Accuracy
Fuzzy Random Forest [14]	85.28%
Random Forest [18]	84.01%
DANN + LightGBM [19]	83.57%
XGBoost [20]	81.21%
DANN [21]	80.89%
CatBoost [22]	78.15%
Naive Bayes [23]	77.81%
Adaboost [24]	75.90%
C4.5 [25]	75.79%
Logistic Regression [26]	74.12%

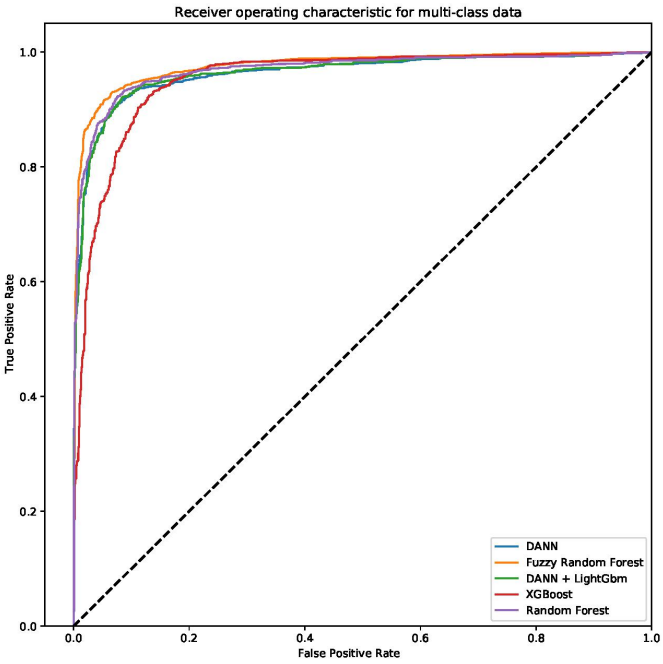

Figure 1. Comparative study of ML algorithms - ROC

As a preliminary step for our main research, we did the data engineering including Multiple Imputation with Chained Equations, feature engineering with correlation matrix and applied the hybrid SMOTE and Tomek on the Cleveland dataset only. The model obtained after these steps, gave the score of accuracy of 92.02%.

One of the challenges in applying this algorithm to this dataset is the size of the Cleveland dataset. In particular, it is hard to apply the powerful state-of-the-art classifiers because of its small size and imbalance. To solve this problem, new semi-synthetic data has been created via an hybrid approach. By combining the SMOTE technique with the Tomek under-sampling technique, the size of the Cleveland dataset has been increased from 227 to 599 instances of the training set.

Feature engineering is done to remove noisy features. The correlation matrix heatmap is constructed and was published [27]. Squares colors depend on the values which range from -1 to 1, with -1 being red and 1 being green. The colors of the correlation matrix are only intended for a better visualisation and understanding of the feature correlations. As the correlation coefficient is closer to 0, it becomes brighter, therefore less influential on each other. Values are sorted by the influence of each attribute on the target variable and removed from a column until the accuracy of the model decreases or does not remain the same. The least influential attribute for the target variable is the blood sugar (fbs) attribute with a correlation coefficient of 0.059. After removing this attribute, the accuracy of the model is improved by 1.1%, which means that this feature is noisy. A further reduction of the attributes does not improve the performance results, so no more reduction is needed.

In our paper we use the Breiman's methodology to construct the random forest of "fuzzy" decision trees. As the result of the experiment, we can confirm that this methodology combines the robustness of multiple classifier systems along with the imperfect data handling of fuzzy systems. We use the fuzzy transformation approach similar to [28]. The proposed solution is shown on Algorithm 1 and 2, corresponding prediction of the Fuzzy Random Forest shown on the Algorithm 3.

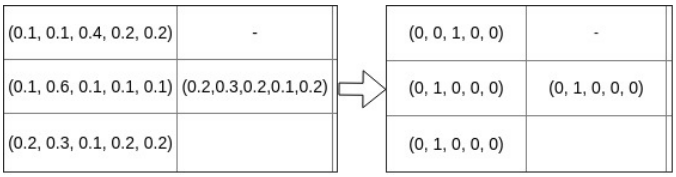

Figure 2. Transformation

Applying the Transformation, described in the Figure 2 onto the $supportVecMat$ from Algorithm 4, we can see that each leaf assigns a simple vote to the majority class. The methods $MajorityVoting1$ and $MajorityVoting2$ are follows:

$$MajorityVoting1(t, i, supportVecMat) =$$

$$\begin{cases} 1 & \text{if } i = \underset{j,j=1,\ldots,I}{\arg\max} \left\{ \sum_{n=1}^{N_t} L_L supportVecMat[t][n][j] \right\} \\ 0 & \text{otherwise} \end{cases}$$

Each tree t assigns a simple vote to the most voted class among N, reached by the leaves.

$$MajorityVoting2(i, confTreeMat) = \sum_{t=1}^{T} confTreeMat_{t,i}$$

In order to fuzzify the crisp values we use a technique proposed in [28]. The domain of each attribute is represented by trapezoid of fuzzy sets. This fuzzy partitioning algorithm guarantees completeness and it is **strong** fuzzy partition.

The main problem is 3 completely missing columns in the second part of the dataset. To append the second part and create a more powerful model, we use a combination of Pseudo-labelling and Fuzzy Random Forest. As indicated on the pseudo-code (Algorithm 4), we form models to predict

Algorithm 1: Fuzzy Random Forest

function FuzzyRandomForest $(dataset, c, N)$;
Input : $dataset$ - expected to be fuzzified data, N - number of trees to be generated, c - predicted column
foreach i *in the range(*N*)* **do**
    // divide the examples set of entry in subsets;
    randomly sample the training data D with replacement to produce D_i;
    create a root node N_i *containing* D_i;
    call BuildFuzzyTree(N_i*)*;
end
return *generated model*;

Algorithm 2: Fuzzy decision tree

function BuildFuzzyTree $(data)$;
Input : $data$ - fuzzified data
foreach e *in the* $data$ **do**
    $\chi_{\text{fuzzy-tree-root}}$(e) = 1
end
// get the set of attributes, where numeric variables fuzzy-partitioned;
S ← SetOfAttributes(data)
// choosing attribute to make a selection at node N;
while *NotSingleClass(N) and NotEmpty(S)* **do**
    // random selection of attributes from S;
    R ← SelectRandom(S);
    maxGain ← 0 ;
    selectedAttr ← null;
    foreach $attr$ *in the* R **do**
        gain = ComputeInformationGain($attr$, $\chi_{\text{fuzzy-tree-N}}$(e))
        if *gain* > *maxGain* **then**
            maxGain ← gain;
            selectedAttr ← $attr$;
        end
    end
    Partition(M, selectedAttr)
end

the missing columns using the Cleveland dataset. Then, using them, we label the missing columns in the second part of the dataset and retrain the model using a combination of labelled and pseudo-labelled data. Then, using those models, we impute the missing columns. The imputation order is important to form the next model and we need to consider the recently imputed column. We experimentally determine that imputing the best model would provide a more powerful model. The accuracies of the models to predict the slope, the number of major vessels colored by fluoroscope and Thallium Stress Test results are 81.12%, 86.11% and 90.01% respectively. Finally, we impute the new dataset into the existing labeled one and feed the entire dataset into the Fuzzy Random Forest classifier.

One of the main parameters of this problem is the multi-class confusion matrix, illustrated in Table IV. The corresponding indexes of rows and columns mapped to the number of vessels narrowing by more than 50%. In this confusion matrix, the *true positives* of the particular class are placed on the diagonal. That is, the true positives of a particular class x are $C(x, x)$, the false positives are $\sum_{i=0} \sum_{j=0} C(i, j) - C(x, x)$, the true negatives are $\sum_{i=0} C(x, i)$ and the false negatives are $\sum_{j=0} C(j, x)$. To obtain the results in the same format as it is in the study in [3], the macro average of the positives and negatives of all the confusion matrices is computed. According to the results obtained, the recall, which is also known as specificity or rate of true positives, is equal to 96.93% and sensitivity is 89.92%, which is higher than the values obtained in the previous study by 1.22% and 1.26%, respectively.

One of the parameters proposed by our model is the Receiver Operating Characteristic (ROC) chart. It illustrates the diagnostic capability of the binary classifier system in which its discrimination varies because the proposed problem is the prediction of the multiclass classification. To binarize input, the One-Vs-Rest classifier was built. True and false positive rates were graphically represented in the ROC graph for each class, as shown in Figure 3. The area under the curve of each

Algorithm 3: Fuzzy Random Forest prediction

Function *FuzzyRandomForest*($data$, $fuzzyRandomForest$)
  $confTreeMat \leftarrow$ DecisionsOfTrees($data$, $fuzzyRandomForest$);
  return DecisionOfForest($confTreeMat$)
end
Function *DecisionsOfTrees(data, $fuzzyRandomForest$)*
  foreach t *in the* $fuzzyRandomForest$ **do**
    // obtain the matrix where each element of the matrix is a vector containing the support for every class provided by every activated leaf of each tree t;
    supportVecMat $\leftarrow$ getsupportVecMat($fuzzyRandomForest$)
    foreach i *in the classes* **do**
      // obtain confidence matrix for each class i assigned by tree t;
      confTreeMat[t][i] $\leftarrow$ MajorityVoting1(t,i, $supportVecMat$)
    end
    return confidenceMatrix
  end
end
Function *DecisionOfForest($confTreeMat$)*
  foreach i *in the classes* **do**
    confidenceForestMatrix$_i$ $\leftarrow$ MajorityVoting2(i, $confTreeMat$);
  end
  $prediction \leftarrow \underset{i,i=1,\ldots,I}{argmax}\{confidenceForestMatrix_i\}$;
  return $prediction$
end

class is close to the value of 1.0, which shows a high accuracy of the model compared to previously proposed techniques.

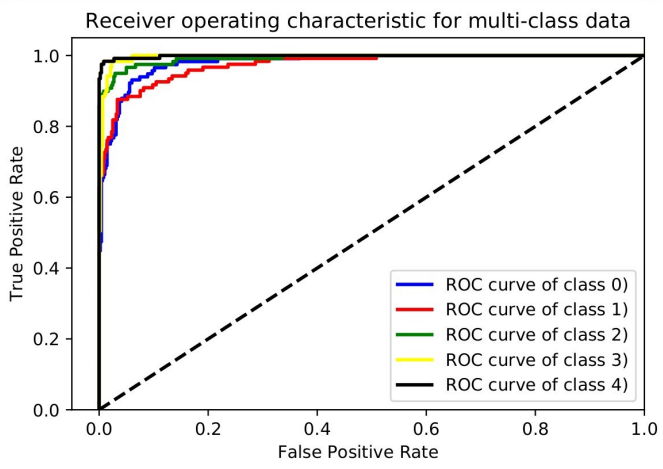

Figure 3. ROC curve of the model

One of the most important metrics is the accuracy of the model. As mentioned earlier, this paper uses k-fold cross-validation, where k = 10. We determine the accuracy of each fold and use the mean of all to obtain the accuracy of our model. Using this technique, the accuracy of our model is 93.2%, which is 1.37% higher than the best existing result in the literature.

V. Conclusion

Imbalance, inaccurate and missing data are important factors for the success of data science techniques. The purpose of this study is to address these challenges and improve the performance of previous series of studies, Table V, using some preprocessing and semi-supervised learning techniques and creating new semi-synthetic data for more effective data science solutions. For prediction, the Fuzzy Random Forest classifier was used, among others, including C4.5, ANN, Naive Bayes, Catboost, Adaboost, Xgboost, Random Forest and Logistic Regression. The Fuzzy Random Forest classifier is a new yet powerful machine learning algorithm, widely used in the field of data science and other fields, from image classification to recommendation systems. The k-fold cross validation was applied to check whether the model was overfitted or not, where k=10. The following metrics were therefore used to evaluate the performance of the model: sensitivity and specificity, ROC curve graph on Figure 3 and accuracy. The accuracy of the model is 93.45% with a specificity and sensitivity of 96.92% and 89.99%, respectively. In conclusion, our approach has improved the performance of each metric, that is, significantly better than the results of the previous published results.

In this paper, we show that unlabeled data, when used in conjunction with some amount of labeled data, can produce a considerable improvement in learning accuracy. This introduced algorithm can greatly help to merge and manipulate multiple datasets, with the imputation of missing columns,

Algorithm 4: Final model

function Train $(dataset1, dataset2, P)$;
Input : Cleveland $dataset1$ and $dataset2$ with missing columns, P - predicted column
missingColumns←[11, 12, 13]
dataset1 ← ApplyMice(dataset1);
dataset1 ← SMOTETomek(dataset1);
foreach *c in the missingColumns* **do**
    `// Apply MICE to the dataset2, but excluding the missing columns;`
    dataset2 ← ApplyMice(dataset2, -e);
end
rank features and select optimal number;
initialize the array with models;
foreach *c in the missingColumns* **do**
    `// train models to predict 11, 12, 13 columns, missing in the dataset2;`
    models[c] ← FuzzyRandomForest(dataset1, c, 300);
end
foreach *c in the missingColumns* **do**
    `// impute the predicted values using models trained on dataset1;`
    dataset2[c]←Predict(models[c], dataset2, c);
end
foreach *c in the missingColumns* **do**
    `// retrain the models including labelled and pseudo-labelled data;`
    models[c] ← FuzzyRandomForest(dataset1+dataset2, c);
end
foreach *c in the missingColumns* **do**
    `// impute missing columns using the Semi-supervised model;`
    dataset2[c]←Predict(models[c], dataset2, c);
end
finalModel←FuzzyRandomForest(dataset1+dataset2, 300);
return finalModel

Table IV
MULTICLASS CONFUSION MATRIX

99	15	9	1	1
13	98	3	6	1
2	5	110	2	0
1	1	0	115	4
1	1	1	0	119

to obtain a more powerful and accurate model. The dataset we created, in addition to this UCI heart disease dataset, is publicly available for the researchers [2]. Additionally, we have implemented the "Forest" of Decision Trees, according to the classic Brienman's Random Forest implementation. By applying few changes to the algorithm, we have implemented the fully functioning Fuzzy Random Forest algorithm as shown on the Algorithm 1, 2 and 3.

The model we have developed here can be applied to various fields of applications. Our model can be used in smartphones, smartwatches or fitness trackers with built-in wearable sensors to track heart conditions and be aware of potential risks. In addition, it can be adapted to smart home technologies or used as an emergency call to medical institutions.

REFERENCES

[1] D. Mozaffarian, E. J. Benjamin, A. S. Go, D. K. Arnett, M. J. Blaha, M. Cushman, S. R. Das, S. De Ferranti, J.-P. Després, H. J. Fullerton *et al.*, "Heart disease and stroke statistics—2016 update," *Circulation*, 2016.

[2] E. Zeinulla, K. Bekbayeva, and A. Yazici, "Heart Disease Dataset extended version," https://github.com/zelzhan/Machine-learning/blob/master/research_extension/output.csv, 2019, [Online; accessed 14-Oct-2019].

[3] Das, Turkoglu, and Sengur, "Effective diagnosis of heart disease through neural networks ensembles." *Expert Systems with Applications*, vol. 36, no. 4, pp. 7675–7680, 2009.

[4] Kumar, "Diagnosis of Heart Disease using Fuzzy Resolution Mechanism," *Journal of Artificial Intelligence*, vol. 5, no. 1, pp. 47–55, 2012.

[5] Detrano, Janosi *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *The American Journal of Cardiology*, vol. 64, no. 5, 1989. [Online]. Available: 10.1016/0002-9149(89)90524-9

[6] Hedeshi and Abadeh, "Coronary Artery Disease

Table V
ACCURACIES OF MODEL WITH THE OTHER STUDIES THAT HAS BEEN CONDUCTED ON THIS DATA SET.

Author	Method	Accuracy (%)
ToolDiag, RA	MLP + BP	65.60
Newton Cheung	Naive Bayes	81.48
WEKA, RA	Naive Bayes	83.60
Polat et al.	Fuzzy-AIRS-Knn based system	87.00
SAS base	Neural networks ensemble	89.01
Senthil Kumar	Combined ANFIS and ANN	91.83
Our proposal - Data engineering without SSL	Fuzzy Random Forest	92.19
Our proposal - Data engineering with our proposed algorithm	Pseudo-Labeling + Fuzzy Random Forest	93.45

Detection Using a Fuzzy-Boosting PSO Approach," *Computational Intelligence and Neuroscience*, vol. 2014, pp. 1–12, 2014. [Online]. Available: 10.1155/2014/783734

[7] Zeng, Zou, Wei, Liu, Wang *et al.*, "Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data," in *2016 IEEE International Conference Of Online Analysis And Computing Science (ICOACS)*, 2016.

[8] D. Dua and C. Graff, "University of California, Irvine (UCI) Machine Learning Repository," 2017. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[9] Azur, Stuart, Frangakis, Leaf *et al.*, "Multiple imputation by chained equations: what is it and how does it work?" *International Journal Of Methods In Psychiatric Research*, vol. 20, no. 1, pp. 40–49, 2011.

[10] Lakshminarayan, Harp, Samad *et al.*, *Applied Intelligence*, vol. 11, no. 3, pp. 259–275, 1999.

[11] Sedgwick, "Pearson's correlation coefficient," *BMJ*, vol. 345, no. 04 1, pp. 4483–4483, 2012.

[12] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[13] Elhassan, Aljourf, Al-Mohanna, Shoukri *et al.*, "Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method," *Global Journal Of Technology And Optimization*, vol. 1, no. 1, 2016.

[14] D. Conn, T. Ngun, G. Li, and C. Ramirez, "Fuzzy forests: extending random forests for correlated, high-dimensional data," 2015.

[15] P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, "A fuzzy random forest," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 729–747, 2010.

[16] Ross, Jensen, Snodgrass, Dyreson, Jensen, Snodgrass *et al.*, "Cross-Validation. Encyclopedia Of Database Systems, 532-538," 2009.

[17] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[18] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[20] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[21] B. Anuradha and V. V. Reddy, "Ann for classification of cardiac arrhythmias," *ARPN Journal of Engineering and Applied Sciences*, vol. 3, no. 3, pp. 1–6, 2008.

[22] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[23] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, 2001, pp. 41–46.

[24] X. Li, L. Wang, and E. Sung, "Adaboost with svm-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, 2008.

[25] S. Ruggieri, "Efficient c4. 5 [classification algorithm]," *IEEE transactions on knowledge and data engineering*, vol. 14, no. 2, pp. 438–444, 2002.

[26] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[27] E. Zeinulla, K. Bekbayeva, and A. Yazici, "Heart Disease Dataset extended version," https://github.com/zelzhan/Machine-learning/blob/master/research_extension/heatmap.pdf, 2020, [Online; accessed 14-Jan-2020].

[28] J. M. Cadenas, M. C. Garrido, R. Martínez, and P. P. Bonissone, "Extending information processing in a fuzzy random forest ensemble," *Soft Computing*, vol. 16, no. 5, pp. 845–861, 2012.