ISSN 2278-3091

Volume 9, No.3, May - June 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse218932020.pdf

https://doi.org/10.30534/ijatcse/2020/218932020

The Impact of Fuzzy Discretization's Output on Classification Accuracy of Random Forest Classifier

Muhammad Nur Fikri[1], Mohd Fadzil Hassan[2], Duc Chung Tran[3]

[1]Department of Computer and Information Science, Universiti Teknologi Petronas, 32610, Seri Iskandar, Perak, Malaysia, nurfikri2014@gmail.com

[2]Department of Computer and Information Science, Universiti Teknologi Petronas, 32610, Seri Iskandar, Perak, Malaysia, mfadzil_hassan@utp.edu.my

[3]Computing Fundamental Department, FPT University, Hoa Lac Hi-Tech Park, Hanoi, Vietnam, 155300, ChungTD6@fe.edu.vn

ABSTRACT

Random Forest is known as among the widely used classification algorithms by researchers and machine learning enthusiast in solving classification problems. Recently, fuzzy discretization has been paired with Random Forest (RF) classifier to enhance the classification accuracy of Random Forest classifier when dealing with continuous variables. However, there are many different opinions on whether there is a need to perform discretization in data pre-processing for tree-based classifiers such as J48, Decision Tree and Random Forest. On top of that, it is known that different classification algorithms produce different classification accuracies depending on the type of data used. In other words, the output of data discretization process. Thus, to unravel this mentioned hypothesis, this study intends to shed some lights on the impact of different fuzzy discretization's output on the classification accuracy of Random Forest classifier. In this study, three version of simulations were done with different fuzzy discretization output. Those fuzzy discretization's outputs are 1) without fuzzy discretization 2) with fully fuzzy discretization and 3) with partial fuzzy discretization. Then, classification phase is done through Random Forest classifier and the classification accuracy for all the simulation versions were observed, recorded, and analyzed.

Key words: Classification Accuracy, Data Pre-Processing, Fuzzy Discretization, Random Forest Classifier.

1. INTRODUCTION

Data preparation or also known as data pre-processing is a vital process in majority of machine learning projects and activities [1]. Depending on the nature of data, there are many methods of data pre-processing that can be applied and data discretization is one of those methods [2]. Data discretization is applied when certain machine learning algorithm can produce better result like classification accuracy, with discrete data [3]. Thus, when using these types of classification algorithms, it is necessary to discretize continuous data in the dataset used. Among the classification algorithms that were used along with discretization in several studies were Bayesian Network (BN), RF, Decision Tree (DT). Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Fuzzy Rule-Based (FRB) [4].

However, for tree-based classification algorithm, it has been found that there was not much significance impact on classification accuracy when applying the existing discretization methods [5]–[10]. As there are many types of discretization, a taxonomy of discretization method has been developed to assist researchers to implement their preferred discretization method onto their projects [11], [12]. Supervised and unsupervised discretization methods are among the most used methods for studies that use tree-based classification algorithms. This has led to several arguments on the necessity of applying discretization method on tree-based classification algorithms since only little efforts of discretization are required and yet the output does not give significant impact on classification accuracy [13], [14].

For real-world problems, the presence of domain's knowledge is very helpful [15]. As real-world data and problems can be imprecise, large and requires linguistic interpretation, here is where fuzzy logic stepped into this process [16], [17]. The application of fuzzy in solving decision problems has been successfully implemented by integrating fuzzy with non-tree-based decision algorithms

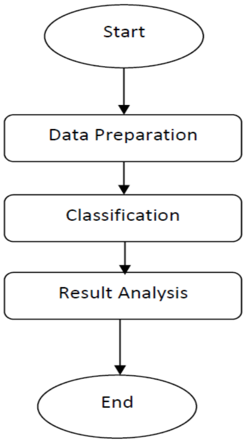


Figure 1: Overview of Proposed Work

This work is support by research grant YUTP [0153AA-E90]

[18], [19]. Recent study by [20], shown that a new framework of pairing fuzzy discretization and random forest, a tree-based decision algorithm, has been proposed to enhance the classification accuracy of random forest classification algorithm. However, since the framework is new, the authors has not yet thoroughly specified the impact of fuzzy in the discretization process, towards the classification accuracy of RF classification algorithm. Although the study able to enhance the classification accuracy of random forest classification algorithm, the reason and impact of fuzzy discretization remains unknown as the author only generally elaborate on the methodology and discussion of the study. Thus, the necessity of applying fuzzy discretization on random forest specifically remains unknown.

Thus, in this study, the objective is to solve the mentioned conflicting argument for tree-based algorithm, The methodology of the authors have been adopted with additional enhancement where the outputs of fuzzy discretization phase were varied to three data samples known as without fuzzy discretization, with fully fuzzy discretization and with partial fuzzy discretization. The preparation of these different fuzzy discretization outputs is necessary to discover which discretization output is more suitable for RF classification algorithm.

2. MATERIALS AND METHODS

2.1 Overview

Figure 1. visualized the overview of the proposed work. According to the flowchart, the process starts with data preparation. The process then continues with classification and ends with result analysis. The details of data preparation and classification phases will be discussed in the next sub-sections while for result analysis, it will be discussed in result and discussion sections

2.2 Data Preparation

Table 1: Characteristics of Dataset

No. of Attributes	No. of Categorical Attributes	No. of Numerical Attributes	No. of Continuous Attributes	No. of Discrete Attributes	Target Attribute
9	1	8	2	7	Class

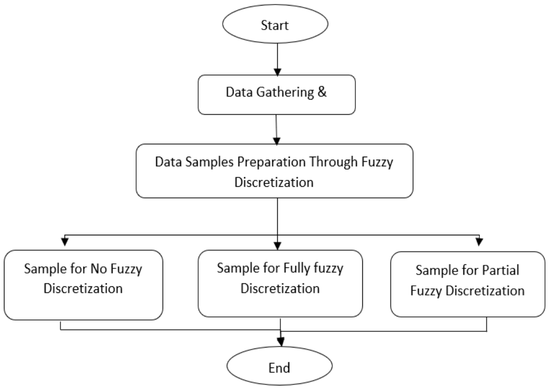

Start
Data Gathering &
Data Samples Preparation Through Fuzzy Discretization
Sample for No Fuzzy Discretization
Sample for Fully fuzzy Discretization
Sample for Partial Fuzzy Discretization
End

Figure 2: Overview of Data Preparation Phase.

Figure 2 illustrates the process flow of data preparation phase. The first step in this work is retrieving a dataset from an online dataset repository known as Kaggle. As the nature of this study is discretization, a dataset that have at least one continuous attribute should be chosen. For this study, Pima Indians Diabetes Dataset has been chosen (https://www.kaggle.com/uciml/pima-indians-diabetes-database). The dataset is then analyzed and the characteristics of data in the dataset are shown in Table 1.

To analyze the impact of fuzzy discretization, three versions of data samples for simulation will be created at the end of this phase. The characteristics of each data samples are:

1) without fuzzy in discretization
2) apply fuzzy in discretization to all attributes
3) partially apply fuzzy in discretization towards selected (continuous) attributes.

Thus, only data sample 2 and 3 needs preparation as data sample 1 require no further action.

2.2.1. Data Samples Preparation Through Fuzzy Discretization

The methods in this subsection consist of several component of fuzzy discretization namely Interval Discretization, Mapping of Fuzzy Linguistic Terms and Fuzzy Interval Values and Data Transformation. These methods are executed chronologically as mentioned above.

2.2.1.1. Interval Discretization

For data samples 2 and 3, the original datasets are first loaded into Weka Explorer (https://waikato.github.io/weka-wiki/downloading_weka/) and the discretization process is done through Equal-Width Binning technique. Through equal-width binning, attributes will be discretized into several equal-sized discrete intervals. To perform equal-width binning discretization, the steps, and equations below are adopted:

Table 2: Discretization for Data Sample 2 for All Attributes

Attributes Names	Values	Count
Pregnancies	0 – 3.4	424
	3.4 – 6.8	175
	6.8 – 10.2	135
	10.2 – 13.6	30
	13.6 - 17	4
Glucose	0 -39.8	5
	39.8 – 79.6	36
	79.6 – 119.4	367
	119.4 – 159.2	258
	159.2 - 199	102
Blood Pressure	0 – 24.4	36
	24.4 – 48.8	15
	48.8 – 73.2	368
	73.2 – 97.6	330
	97.6 - 122	19
Skin Thickness	0 - 19.8	338
	19.8 – 39.6	340
	39.6 – 59.4	87
	59.4 – 79.2	2
	79.2 - 99	1
Insulin	0 – 169.2	642
	169.2 – 338.4	100
	338.4 – 507.6	17
	507.6 – 676.8	6
	676.8 – 84.6	3
Age	21 – 33	474
	33 – 45	176
	45 – 57	76
	57 – 69	39
	69 - 81	3
Diabetes Pedigree Function (PEDI)	0.078 – 0.5464	524
	0.5464 – 1.0148	194
	1.0148 – 1.4832	40
	1.4832 – 1.9516	6
	1.9516 - 2.42	4
Body Mass Index (MASS)	0 – 13.42	11
	13.42 – 26.84	171
	26.84 – 40.26	492
	40.26 – 53.68	90
	54.68 – 67.1	4

a. Determine number of intervals, k

b. Finding max values of an attribute using the equation:

$$Vmax = \max \{V1,V2,V3\ldots Vn\} \qquad (1)$$

c. Finding min values of an attribute using the equation:

$$Vmin = \min \{V1,V2,V3\ldots Vn\} \qquad (2)$$

d. Finding width of interval using the equation:

$$W = (Vmax - Vmin) / k \qquad (3)$$

where Vmax and Vmin are maximum and minimum values of attributes, respectively. K is the number of intervals usually specified by domain's expert. For the purpose of simulation in this study, K is 5.

Table 2 shows the outputs of equal-width binning discretization for data sample 2. The results of interval discretization for data sample 3 is the same as data sample 2 but only PEDI and MASS attribute were discretized as these attributes are continuous attributes.

2.2.1.2. Mapping of Fuzzy Linguistic Term and Fuzzy Interval Values

Then the fuzzy discretization process continues with fuzzy linguistic terms and fuzzy discrete intervals values. The purpose of fuzzy linguistic terms is to label the equal-sized bins and the purpose of implementing fuzzy discrete interval values are to transform the original values from the dataset into the corresponding discrete fuzzy interval values. Table 3. shows the fuzzy discretization process through fuzzy linguistic terms and fuzzy discrete intervals values where the discretized attributes are labelled with the corresponding linguistic terms and interval values.

Table 3: Mapping of Fuzzy Linguistic Term and Fuzzy Interval Values

Attributes Names	Values	Count	Fuzzy Interval
Pregnancies	0 – 3.4	424	1
	3.4 – 6.8	175	2
	6.8 – 10.2	135	3
	10.2 – 13.6	30	4
	13.6 - 17	4	5
Glucose	0 -39.8	Very Low	1
	39.8 – 79.6	Low	2
	79.6 – 119.4	Moderate	3
	119.4 – 159.2	High	4
	159.2 - 199	Very High	5
Blood Pressure	0 – 24.4	Very Low	1
	24.4 – 48.8	Low	2
	48.8 – 73.2	Moderate	3
	73.2 – 97.6	High	4
	97.6 - 122	Very High	5
Skin Thickness	0 - 19.8	Very Low	1
	19.8 – 39.6	Low	2
	39.6 – 59.4	Moderate	3
	59.4 – 79.2	High	4
	79.2 - 99	Very High	5
Insulin	0 – 169.2	Very Low	1
	169.2 – 338.4	Low	2
	338.4 – 507.6	Moderate	3
	507.6 – 676.8	High	4

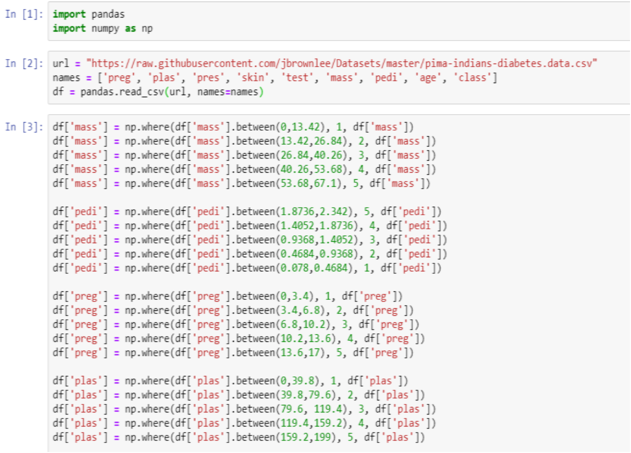
	676.8 – 84.6	Very High	5
Age	21 – 33	Very Low	1
	33 – 45	Low	2
	45 – 57	Moderate	3
	57 – 69	High	4
	69 - 81	Very High	5
Diabetes Pedigree Function (PEDI)	0.078 – 0.5464	Very Low	1
	0.5464 – 1.0148	Low	2
	1.0148 – 1.4832	Moderate	3
	1.4832 – 1.9516	High	4
	1.9516 - 2.42	Very High	5
Body Mass Index (MASS)	0 – 13.42	Very Low	1
	13.42 – 26.84	Low	2
	26.84 – 40.26	Moderate	3
	40.26 – 53.68	High	4
	54.68 – 67.1	Very High	5

2.2.1.3. Data Transformation

Then, the final steps in fuzzy discretization is data transformation where the original data will be replaced with the corresponding discretized data based on the table. In this step, the original datasets are loaded into Jupyter-Notebook. Phytons programming language and data analysis libraries such as Pandas and Numpy are used in this step

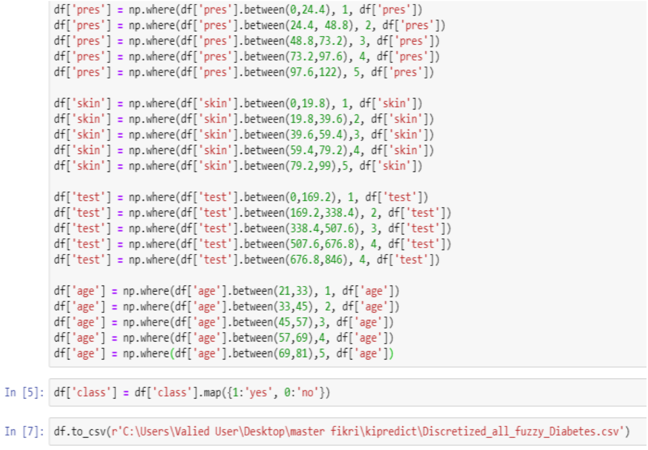
Figure 3. and Figure 4. show snapshots of the codes used to run the data transformation process for data samples 2 and 3 respectively. Pandas and Numpy libraries are first called to be used in the current Phyton environment in Jupyter-Notebook. Then, the dataset is pulled from the dataset repository and all the attributes are labelled with specific names. After that, the pulled dataset is converted into a python data frame. Based on the Table 3, for data sample 2, all original attributes' values are transformed with the discrete fuzzy interval values. Meanwhile, for data sample 3, all selected original attributes' values are also transformed with the corresponding discrete fuzzy interval values. As Weka classifier runs only with categorical output, thus, the numerical values for the "class" attributes are replaced with categorical values where 1 = yes and 0 = no. Lastly, the data frame is saved into a csv file format and ready to be used for classification phase.

The output of this data preparation phase are 3 data samples with different data preparation output values to test the classification accuracy of random forest classifier in the classification phase. Figure 5. shows the header of the original dataset. Meanwhile, Figure 6. and Figure 7. show the header of data samples 2 and 3 respectively.



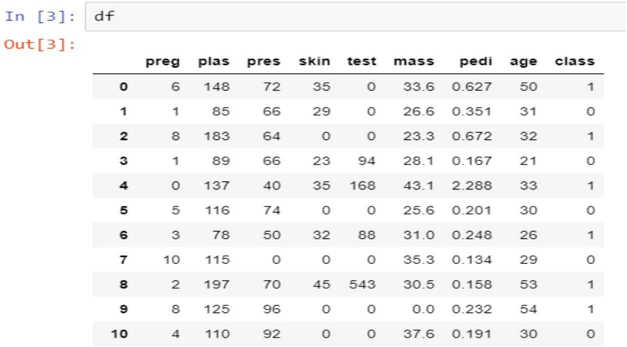
```
In [1]: import pandas
        import numpy as np

In [2]: url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
        names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
        df = pandas.read_csv(url, names=names)

In [3]: df['mass'] = np.where(df['mass'].between(0,13.42), 1, df['mass'])
        df['mass'] = np.where(df['mass'].between(13.42,26.84), 2, df['mass'])
        df['mass'] = np.where(df['mass'].between(26.84,40.26), 3, df['mass'])
        df['mass'] = np.where(df['mass'].between(40.26,53.68), 4, df['mass'])
        df['mass'] = np.where(df['mass'].between(53.68,67.1), 5, df['mass'])

        df['pedi'] = np.where(df['pedi'].between(1.8736,2.342), 5, df['pedi'])
        df['pedi'] = np.where(df['pedi'].between(1.4052,1.8736), 4, df['pedi'])
        df['pedi'] = np.where(df['pedi'].between(0.9368,1.4052), 3, df['pedi'])
        df['pedi'] = np.where(df['pedi'].between(0.4684,0.9368), 2, df['pedi'])
        df['pedi'] = np.where(df['pedi'].between(0.078,0.4684), 1, df['pedi'])

        df['preg'] = np.where(df['preg'].between(0,3.4), 1, df['preg'])
        df['preg'] = np.where(df['preg'].between(3.4,6.8), 2, df['preg'])
        df['preg'] = np.where(df['preg'].between(6.8,10.2), 3, df['preg'])
        df['preg'] = np.where(df['preg'].between(10.2,13.6), 4, df['preg'])
        df['preg'] = np.where(df['preg'].between(13.6,17), 5, df['preg'])

        df['plas'] = np.where(df['plas'].between(0,39.8), 1, df['plas'])
        df['plas'] = np.where(df['plas'].between(39.8,79.6), 2, df['plas'])
        df['plas'] = np.where(df['plas'].between(79.6, 119.4), 3, df['plas'])
        df['plas'] = np.where(df['plas'].between(119.4,159.2), 4, df['plas'])
        df['plas'] = np.where(df['plas'].between(159.2,199), 5, df['plas'])
```

Figure 3: Data Transformation



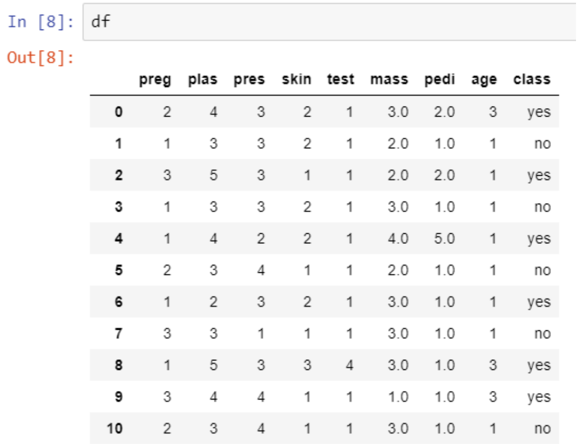
```
        df['pres'] = np.where(df['pres'].between(0,24.4), 1, df['pres'])
        df['pres'] = np.where(df['pres'].between(24.4, 48.8), 2, df['pres'])
        df['pres'] = np.where(df['pres'].between(48.8,73.2), 3, df['pres'])
        df['pres'] = np.where(df['pres'].between(73.2,97.6), 4, df['pres'])
        df['pres'] = np.where(df['pres'].between(97.6,122), 5, df['pres'])

        df['skin'] = np.where(df['skin'].between(0,19.8), 1, df['skin'])
        df['skin'] = np.where(df['skin'].between(19.8,39.6),2, df['skin'])
        df['skin'] = np.where(df['skin'].between(39.6,59.4),3, df['skin'])
        df['skin'] = np.where(df['skin'].between(59.4,79.2),4, df['skin'])
        df['skin'] = np.where(df['skin'].between(79.2,99),5, df['skin'])

        df['test'] = np.where(df['test'].between(0,169.2), 1, df['test'])
        df['test'] = np.where(df['test'].between(169.2,338.4), 2, df['test'])
        df['test'] = np.where(df['test'].between(338.4,507.6), 3, df['test'])
        df['test'] = np.where(df['test'].between(507.6,676.8), 4, df['test'])
        df['test'] = np.where(df['test'].between(676.8,846), 4, df['test'])

        df['age'] = np.where(df['age'].between(21,33), 1, df['age'])
        df['age'] = np.where(df['age'].between(33,45), 2, df['age'])
        df['age'] = np.where(df['age'].between(45,57),3, df['age'])
        df['age'] = np.where(df['age'].between(57,69),4, df['age'])
        df['age'] = np.where(df['age'].between(69,81),5, df['age'])

In [5]: df['class'] = df['class'].map({1:'yes', 0:'no'})

In [7]: df.to_csv(r'C:\Users\Valied User\Desktop\master fikri\kipredict\Discretized_all_fuzzy_Diabetes.csv')
```

Figure 4: Data Transformation (cont.)



```
In [3]: df
Out[3]:
        preg  plas  pres  skin  test  mass   pedi  age  class
     0     6   148    72    35     0  33.6  0.627   50      1
     1     1    85    66    29     0  26.6  0.351   31      0
     2     8   183    64     0     0  23.3  0.672   32      1
     3     1    89    66    23    94  28.1  0.167   21      0
     4     0   137    40    35   168  43.1  2.288   33      1
     5     5   116    74     0     0  25.6  0.201   30      0
     6     3    78    50    32    88  31.0  0.248   26      1
     7    10   115     0     0     0  35.3  0.134   29      0
     8     2   197    70    45   543  30.5  0.158   53      1
     9     8   125    96     0     0   0.0  0.232   54      1
    10     4   110    92     0     0  37.6  0.191   30      0
```

Figure 5: Original Dataset

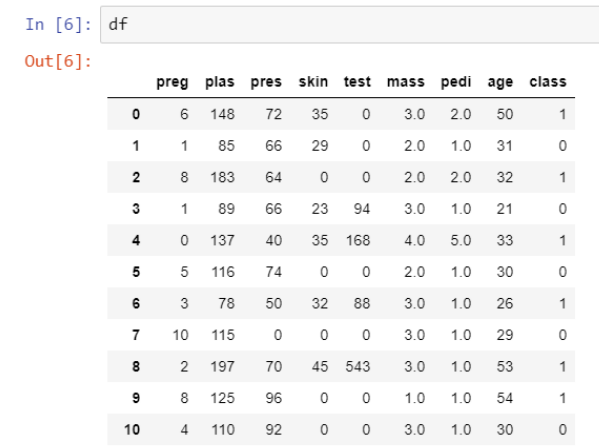


In [8]: df

Out[8]:

	preg	plas	pres	skin	test	mass	pedi	age	class
0	2	4	3	2	1	3.0	2.0	3	yes
1	1	3	3	2	1	2.0	1.0	1	no
2	3	5	3	1	1	2.0	2.0	1	yes
3	1	3	3	2	1	3.0	1.0	1	no
4	1	4	2	2	1	4.0	5.0	1	yes
5	2	3	4	1	1	2.0	1.0	1	no
6	1	2	3	2	1	3.0	1.0	1	yes
7	3	3	1	1	1	3.0	1.0	1	no
8	1	5	3	3	4	3.0	1.0	3	yes
9	3	4	4	1	1	1.0	1.0	3	yes
10	2	3	4	1	1	3.0	1.0	1	no

Figure 6: Transformed Dataset for Data Sample 2



In [6]: df

Out[6]:

	preg	plas	pres	skin	test	mass	pedi	age	class
0	6	148	72	35	0	3.0	2.0	50	1
1	1	85	66	29	0	2.0	1.0	31	0
2	8	183	64	0	0	2.0	2.0	32	1
3	1	89	66	23	94	3.0	1.0	21	0
4	0	137	40	35	168	4.0	5.0	33	1
5	5	116	74	0	0	2.0	1.0	30	0
6	3	78	50	32	88	3.0	1.0	26	1
7	10	115	0	0	0	3.0	1.0	29	0
8	2	197	70	45	543	3.0	1.0	53	1
9	8	125	96	0	0	1.0	1.0	54	1
10	4	110	92	0	0	3.0	1.0	30	0

Figure 7: Transformed Dataset for Data Sample 3

2.3 Classification

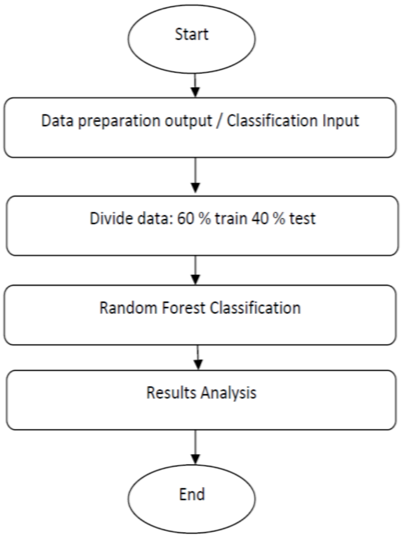

Start
Data preparation output / Classification Input
Divide data: 60 % train 40 % test
Random Forest Classification
Results Analysis
End

Figure 8: Overview of Classification Phase

Figure 8. shows the process flow of the classification phase. In this phase, all three samples created in the previous phase will be used as an input. The samples will be loaded into Weka Explorer for classification. The datasets are divided into 60% training and 40% testing and classification process is done using Random Forest classifier. The results' summary for all three simulations is observed and analyzed.

2.4 Result Analysis (Methods Used)

Two methods were used in measuring the performance of the proposed work namely "Confusion Matrix" and "Classification Accuracy".

2.4.1. Confusion Matrix

Table 4: Confusion Matrix's Components

	YES	NO
YES	True Positive	False Positive
NO	False Negative	True Negative

Table 4. shows the components of a confusion matrix. It contains four major components known as True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN), TP is the number of instances that are correctly classified into the positive decision class, FP is the number of instances that are correctly classified into the negative decision class, FN is number of instances that are incorrectly classified into the positive decision class and TN is number of instances that are correctly classified into the negative decision class

2.4.2. Classification Accuracy

The classification accuracy was computed using all the components in the confusion matrix (TP, TN, FP and FN). It consists of correctly and incorrectly classified instances are computed through the following formula (4) and (5):

$$CA = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

$$CI = 100\% - CA \quad (5)$$

where correctly classified instances are denoted as Classification Accuracy (CA) and incorrectly classified instances is denoted as Classification Inaccuracy (CI).

3. RESULTS AND DISCUSSIONS

Table 5: Summary of All Data Samples 's Confusion Matrix

	Without Fuzzy	Fully Fuzzy	Partial Fuzzy
TP	57	55	62
TN	150	156	154
FP	48	50	43
FN	52	46	48

Table 6: Summary of All Data Samples ‘s Classification Accuracy

	Without Fuzzy	**Fully Fuzzy**	**Partial Fuzzy**
Correctly Classified Instances	67.43%	68.73%	70.36%
Incorrectly Classified Instances	32.57%	31.27%	29.64%
Number of Decision Trees	221	359	269

This section shows the result of all three simulations. Table 5. shows the compilation of confusion matrixes of all data samples. Based on the table, it is observed that data sample without fuzzy discretization obtained the lowest correctly classified instances (combination of TP and TN) and the highest incorrectly classified instances (combination of FP and FN). Meanwhile for data sample with fully and partial fuzzy discretization, it observed that both data samples outperformed the first data sample in term of correctly and incorrectly classified instances. This indicates that the presence of fuzzy discretization able to helps the random forest classifier improve its classification ability.

Table 6. shows the summary of classification accuracy of the random forest classifier which consist of correctly and incorrectly classified instances and number of decision trees. The number of decision tress denoted the complexity of the model built. Although data sample without fuzzy discretization have the lowest complexity with 221 number of decision trees, it still obtained the least CA compared to data samples with the presence of fuzzy discretization due the presence of continuous attributes values in the data samples. The continuous attribute values make all the decision trees aggregated by the random forest to be having issue in classifying the attribute values during the splitting of the decision trees. This has led to inaccurate decision tree splitting for all the decision trees and thus producing lower percentage of CA and higher percentage of CI.

Next, it is observed that data sample with fully fuzzy discretization obtained the highest number of tress compared to other data samples. Although it has the highest complexity but having the large number of trees is good for random forest during aggregation process. More trees mean more variations of outputs that can be learn by the forest to have better classification results. However, it is also observed that this data sample obtained lower CI compared to data sample with partial fuzzy discretization. As all attributes are replaced into discrete fuzzy interval values, the data sample is “over-discretized” which means that some of the attribute values may be discretized into the wrong discrete interval which may affect the “class” output of the dataset. This will subsequently, affect the splitting process of decision tree computation and will lead to higher number of inaccurate class prediction. Thus, having large number of inaccurate trees will affect the classification accuracy of random forest classifier when the model is tested with the 40% of the original dataset.

Finally, it is found that data sample with partial fuzzy discretization yielded the highest CA and lowest CI with moderate number of trees compared to the other data samples. For this data sample, only continuous attributes values are discretized through fuzzy discretization thus making the final output of the data sample to be consisted of partially original discrete values and partially discrete fuzzy intervals. Thus, with the absence of continuous values and no “over-discretized” the other attributes allow this data samples to thrive further than the other data samples

4. CONCLUSION

In a nutshell, the simulation results showed three hypotheses can be concluded. First, the presence of fuzzy discretization does enhance the classification accuracy of random forest classification algorithms. Next, utilizing fuzzy discrete intervals on all attribute’s values can lead to degradation of classification accuracy for random forest classification algorithm as “over-discretization” can occur. Lastly, applying fuzzy discretization on only continuous attributes and leave the other attributes with its’ original discrete values can lead to a better classification accuracy of random forest compared to transforming all attributes into discrete fuzzy interval values. Thus, it is recommended to only apply fuzzy and discretize identified and selected continuous attributes values to reach better classification performance as showed in the result and discussion section previously.

ACKNOWLEDGEMENT

The authors thank Univerisiti Teknologi Petronas for providing the support in term of resources and materials used.

REFERENCES

[1] Z. Guan, T. Ji, X. Qian, and X. Hong, **“A Survey on Big Data Pre-Processing,”** pp. 2–8, 2017.

[2] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, **“The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning,”** *IEMECON 2019 - 9th Annu. Inf. Technol. Electromechanical Eng. Microelectron. Conf.*, pp. 279–283, 2019.

[3] R. Thaiphan and T. Phetkaew, **“Comparative Analysis of Discretization Algorithms on Decision Tree,”** *2018 IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci.*, pp. 63–67, 2018. https://doi.org/10.1109/ICIS.2018.8466449

[4] K. Lavangnananda, **“Study of Discretization**

Methods in Classification," no. February, 2017.

[5] M. Hacibeyoglu and A. Arslan, **"Improving Classification Accuracy with Discretization on Datasets Including Continuous Valued Features,"** no. June, 2011.

[6] E. A. Kareem and M. Duaimi, **"Improved Accuracy for Decision Tree Algorithm Based on Unsupervised Discretization,"** no. September, 2014.

[7] S. A. Ludwig, **"Analyzing gene expression data : Fuzzy decision tree algorithm applied to the classification of cancer data Analyzing Gene Expression Data : Fuzzy Decision Tree Algorithm applied to the Classification of Cancer Data,"** no. August, 2015.

[8] S. Sardari, M. Eftekhari, and F. Afsari, **"Hesitant fuzzy decision tree approach for highly imbalanced data classification,"** *Appl. Soft Comput. J.*, vol. 61, pp. 727–741, 2017.

[9] C. Tsai and Y. Chen, **"The optimal combination of feature selection and data discretization : An empirical study,"** *Inf. Sci. (Ny).*, vol. 505, pp. 282–293, 2019. https://doi.org/10.1016/j.ins.2019.07.091

[10] H. D. Gadade and D. D. K. Kirange, **"Machine Learning Approach towards Tomato Leaf Disease Classification,"** *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, 2020. https://doi.org/10.30534/ijatcse/2020/67912020

[11] S. Ram, D. Mart, and M. Ben, **"Data Discretization : Taxonomy and Big Data Challenge,"** pp. 1–26.

[12] A. A. Bakar, Z. A. Othman, N. Liyana, and M. Shuib, **"Building A New Taxonomy For Data Discretization Techniques,"** *2009 2nd Conf. Data Min. Optim.*, no. October, pp. 132–140, 2009.

[13] Z. Chen, **"The Application of Tree-based model to Unbalanced German Credit Data Analysis,"** *MATEC Web Conf.*, vol. 232, 2018.

[14] N. Satyanarayana, C. Ramalingaswamy, and Y. Ramadevi, **"Survey of Classification Techniques in Data Mining,"** *IJISET -International J. Innov. Sci. Eng. Technol.*, vol. 1, no. 9, pp. 268–278, 2014.

[15] P. Chahuara, F. Portet, and M. Vacher, **"Making Context Aware Decision from Uncertain Information in a Smart Home: A Markov Logic Network Approach."**

[16] J. Hariharakrishnan, S. Mohanavalli, Srividya, and K. B. Sundhara Kumar, **"Survey of pre-processing techniques for mining big data,"** *Int. Conf. Comput. Commun. Signal Process. Spec. Focus IoT, ICCCSP 2017*, 2017. https://doi.org/10.1109/ICCCSP.2017.7944072

[17] R. G. Mehta, D. P. Rana, and M. A. Zaveri, **"A Novel Fuzzy Based Classification for Data Mining using Fuzzy Discretization,"** *2009 WRI World Congr. Comput. Sci. Inf. Eng.*, vol. 3, pp. 713–717, 2009.

[18] N. I. Ahmad-Azami, N. Yusoff, and K. R. Ku-Mahamud, **"Fuzzy Discretization Technique for Bayesian Flood Disaster Model,"** vol. 2, no. 2, pp. 167–189, 2018.

[19] M. H. Abu Yazid, M. S. Talib, and M. H. Satria, **"Heart Disease Classification Framework Using Fuzzy and Flower Pollination Neural Network,"** *Int. J. Adv. Trends Comput. Sci. Eng. Available*, vol. 8, no. 3, pp. 195–200, 2019.

[20] M. N. F. Hishamuddin, M. F. Hassan, and Ai. A. Mokhtar, **"Improving Classification Accuracy of Random Forest Algorithm Using Unsupervised Discretization with Fuzzy Partition and Fuzzy Set Intervals,"** in *Proceedings of the 2020 9th International Conference on Software and Computer*, 2020, pp. 99–104. https://doi.org/10.1145/3384544.3384590