Who Will Come: Predicting Freshman Registration Based on Decision Tree

Lei Yang[1], Li Feng[1, *], Liwei Tian[1] and Hongning Dai[1]

Abstract: The registration rate of freshmen has been a great concern at many colleges and universities, particularly private institutions. Traditionally, there are two inquiry methods: telephone and tuition-payment-status. Unfortunately, the former is not only time-consuming but also suffers from the fact that many students tend to keep their choices secret. On the other hand, the latter is not always feasible because only few students are willing to pay their university tuition fees in advance. It is often believed that it is impossible to predict incoming freshmen's choice of university due to the large amount of subjectivity. However, if we look at the two major considerations a potential freshman contemplates in making a choice, such as the geographical location of the university in relation to his/her home town, and testimonies about of that college life experience by previous graduates, we believe it is possible to predict future enrollment decisions. This paper is the first to find a way to solve the problem of predicting the choice of university a freshman will attend. Our contributions include the following: 1. we present a dataset on freshman registration; 2. we propose a decision-tree-based approach for freshman registration prediction. Study results show that freshman registration is predictable.

Keywords: Decision tree, prediction, registration.

1 Introduction

In universities, the possibility of accurately predicting the number of freshmen registration is low. Universities require a significant amount of confirmation before the arrival of freshmen, such as the number of dorms, recruitment of new teachers, and the expansion of canteens. These are all related to the number of freshmen. If the preparation is inadequate before the orientation begins, the university management will be disordered, which may affect its reputation.

For the prediction of students' registration, most universities tend to apply two traditional methods. The first is to confirm by telephone. Unfortunately, most students cannot be contacted because the telephone numbers they provided were the office numbers of their former high schools. Even if the students are contacted, they may have applied to

[1] Macau University of Science and Technology, Macau.

[*] Corresponding Author: Li Feng. Email: lfeng@must.edu.mo.

Received: 04 February 2020; Accepted: 30 May 2020.

multiple schools and are hesitant to tell the truth. Another method is through data acquisition after students' payments have been processed, which is not an optimal method as well. Typically, Chinese students prefer to pay on-site at the time of registration rather than pay in advance.

Therefore, universities typically perform an estimation based on the registration rate of previous years; however, this estimation is vague and places a high risk on the decision making of universities. After many years, universities have accumulated a large amount of data, i.e., student admission and registration data. However, universities typically do not fully utilize the data; hence, the data is known as sleeping data.

Thus far, no researcher has used machine learning to study this data. Therefore, we propose a new dataset, i.e., the new student's admission registration dataset. We performed a significant amount of preprocessing on the dataset, including data cleaning, reduction, and conversion such that it can be recognized by the machine. In this dataset, the data from the first three years was used for the training set, and those from the fourth year was used as the test set.

Wang et al. [Wang, Jiang, Luo et al. (2019)] discovered that the decision tree algorithm is often used to machine learn the dataset, and our results show that the registration prediction for freshmen is feasible. Song et al. [Song, Zeng, Li et al. (2017)] discovered that a fully grown decision tree is generated by the training set, which is known as the original tree. Next, the number of samples on the leaf nodes is optimized in the original tree, of which the resulting decision tree is known as a sample optimization tree. Subsequently, the number of layers of the original tree is pruned, and the resulting decision tree is known as a layer optimization tree [Alkhalid, Amin, Chikalov et al. (2013)].

These three types of trees have been used to separately predict test sets. The result shows that the accuracy rate can exceed 60%. To evaluate this study more accurately, the F-measure [Luo, Qin, Xiang et al. (2019); Shi, He and Wang (2019)] evaluation criterion was introduced. The result shows that the F-measure value is approximately 0.7, thereby proving the effectiveness of the study.

2 The dataset

2.1 Source

The data used in this study were obtained from a university in Guangzhou and comprised four years of admission data and registration data from 2009 to 2012. These data have been authorized by universities to be used only for research purposes. As the data relates to students' personal information, certain specific information will be desensitized herein to protect their privacy [Bertsimas and Dunn (2017); Linty, Farasin, Favenza et al. (2019)].

2.2 Features

The admission data were obtained from the candidate information database of the Guangdong Provincial Admissions Office. The basic information included the student's candidate number, name, gender, ID number, professional code, professional name, date of birth, household registration, telephone number, etc. The dataset contained 38 columns of data, as shown in Tab. 1.

Table 1: Admission dataset

School SN	Name	Official NO.	G-code	G-name	ID	Score	...	+score	Hometown
0106**	Liu**	094401**	2	Female	440**	307	...	20	Guangzhou
0511**	Lin**	094405**	1	Male	440**	429	...		Shenzhen
5122**	Zhan**	094451**	2	Female	445**	347	...		Dongguan
...	...	...	...	...	...	...	...	...	...

Meanwhile, the registration data were obtained from the educational administration system of the university. The basic information included the student's student number, name, gender, ID number, major, etc. The dataset contained seven columns of data, as shown in Tab. 2.

Table 2: Registration dataset

School ID	Name	Gender	ID	College	Profession	Class
2009**	Liang**	Male	441**	Vehicle	Auto Service	09 Auto 1
2009**	Lin**	Male	440**	Vehicle	Auto Service	09 Auto 1
2009**	Chen**	Male	441**	Vehicle	Auto Service	09 Auto 1
2009**	He**	Male	441**	Vehicle	Auto Service	09 Auto 1
...	...	...	...	...	...	...

The student's ID was used to perform a unique match between the admission data and the registration data. The matching information will be added to Tab. 1 in a new column to store values corresponding to the status, i.e., 0 means not registered, and 1 means registered, as shown in Tab. 3.

Table 3: New dataset

Status	School SN	Name	Official NO.	G-code	G-name	ID	Score	...	+score	Hometown
0	0106**	Liu**	094401**	2	Female	440**	307	...	20	Guangzhou
1	0511**	Lin**	094405**	1	Male	440**	429	...		Shenzhen
1	5122**	Zhan**	094451**	2	Female	445**	347	...		Dongguan
...	...	...	...	...	...	...	...	...	...	...

2.3 Dataset size

The entire dataset contained four years of data. The total amount of data was 10,382, of which 5,989 were registered and 4,393 not registered. The data from the first three years were used as the training set, whereas those from the fourth were used as the test set. The total number of samples in the training and test sets were 6,599 and 3,783, respectively.

3 Data preprocessing

The data must be preprocessed before machine learning is performed on them [Yang (2010)]. This is because most of the raw data collected were missing, noisy, repetitive, ambiguous, or incomplete [Lopez-Chau, Cervantes, Lopez-Garcia et al. (2013)]. These data cannot be directly used by the program; therefore, the raw data must be preprocessed. Furthermore, many data types were not directly involved in the operation, such as name, email address, secondary name, and other text-type data; therefore, these data types must be converted such that the program can perform operations [Wang, Qin, Xiang et al. (2019); Liu, Xiao, Liu et al. (2018)].

After the processing, the dataset was reduced from the original 38 columns to 17 columns, and the null value was padded to 0; the processed dataset is shown in Tab. 4.

Table 4: Processed dataset

	y	a_1	a_2	a_3	...	a_{14}	a_{15}	a_{16}
	Reg	Gender	Score	Course	...	Senior	Nation	Poli
x_1	1	1	465	0	...	1	1	3
x_2	1	1	463	0	...	1	1	13
x_3	1	2	462	0	...	1	1	13
...	...	...	...	...	...	...	...	...

The symbol y denotes the registration status, i.e., 1 for registered, and 0 for not registered. The other symbols are defined as follows:

Attribute set A=$\{a_1, a_2, a_3, ... a_{15}, a_{16}\}$
Sample $x_i=\{a_1^i, a_2^i, a_3^i, ... a_{15}^i, a_{16}^i\}$
Samples set $X=\{x_1, x_2, x_3, ... x_{10381}, x_{10382}\}$
Training set $D=\{(x_1, y_1), (x_2, y_2), …, (x_{6599}, y_{6599})\}$
e.g., (x_1, y_1)=(1 465 0 0 0 7 1421 1 1402011 1 1 514021 4 1 1 3, 1)

4 Proposed method

4.1 Original tree

A decision tree is a predictive analysis model of a tree structure that reflects the mapping between objects and their attribute values [Trabelsi, Elouedi and Lefevre (2019)]. It comprises a root node, branch node, and leaf node. The latter is the starting point of the entire decision tree and is located at the top. The branch node is a new attribute formed by dividing an upper node, representing a data subset of data. The leaf node represents the classification result [Liu, Xiang, Qin et al. (2019)]. The decision tree judges from the root node and selects the node according to the attribute value of the upper node in a top-down manner until the leaf node forms a new class. Each path of the decision tree from the root node to the leaf node is a predictive path that visually represents the relationship between attributes and results [Sempere (2019)]. The procedure above is detailed in Algorithm 1; we name it the original tree.

```
Algorithm 1 Original Tree Generate
INPUT: Training set $D$;
       Attribute set $A$
OUTPUT: An original tree rooted at the node

Procedure: function OriginalTreeGenerate ($D$, $A$)
Generate a node;
if The samples in $D$ all belong to the same type $C$ then
    Mark node as a $C$-type leaf node; return
end if
if $A$=Ø OR The sample in $D$ has the same value on $A$ then
    Mark node as a leaf node;
    The type is marked as the class with the most samples in $D$; return
end if
Select the optimal partition attribute $a_*$ from $A$;
for Each value $a_*^i$ in $a_*$ do
    Generate a branch for node;
    Let $D_i$ denote a subset of samples in $D$ that have a value of $a_*^i$ on $a_*$;
    if $D_i$ is empty then
        Mark branch nodes as leaf nodes;
        The type is marked as the class with the most samples in D; return
    else
        OriginalTreeGenerate ($D_i$, $A\backslash\{a_*\}$) as a branch node;
    end if
end for
```

4.2 Sample and layer optimization trees

To classify the training samples as accurately as possible when growing the original tree above, node division will be repeated, and a complete decision tree will be generated [Mu, Liu, Wang et al. (2019)]. The complete decision tree is not an optimal classification prediction tree because the complete decision tree is extremely "precise" to describe the training data, which will cause overfitting. Therefore, it is necessary to prune this luxuriant tree to improve its generalization ability.

To address overfitting [Pan, Qin, Chen et al. (2019)], two schemes are used for pruning. One is to prune by optimizing the number of samples of the leaf nodes [Mu, Remiszewski, Kon et al. (2018)]. The tree after being pruned is known as the sample optimization tree. The second is to prune by reducing the number of layers in the original tree. The tree processed using this method is known as the layer optimization tree.

On the sample optimization tree, we performed pruning by determining the optimal sample size of the leaf nodes. If the number of samples of a leaf node is less than this value, it will be trimmed. This value is determined by the classification loss function. A lower loss indicates a better predictive model. In this study, the classification error function was used to evaluate the loss and then determine the optimal sample value. The procedure above is detailed in Algorithm 2; we name it the sample optimization tree.

Algorithm 2 Sample optimization tree generation

```
INPUT: Original Tree;
        $N = \{n_1, n_1+10, n_1+20, ..., 500\}$
OUTPUT: Sample optimization tree
Procedure: function Sam_opt_Tree_Generate (Original Tree, $n$)
for Each $n_i$ in $N$ do
    Calculate cross validation errors $err_i$;
end for
Take the smallest $err$ corresponding to $n$;
Set the number of samples for the leaf node to $n$;
Delete a leaf node whose sample size is less than $n$;
```

On the layer optimization tree, we performed pruning by optimizing the number of layers in the decision tree. The specific approach is to gradually reduce the number of layers of the tree from the original tree, obtain the layer value of the least loss, and then use this value to trim the entire tree. The procedure above is similar to Algorithm 2; we name it the layer optimization tree.

5 Performance metric

For the binary classification problem [Lopez-Chau, Cervantes, Lopez-Garcia et al. (2013)], the sample can be categorized into four cases: true positive (TP), false positive (FP), true negative (TN), and false negative (FN), according to the combination of its real category and the classifier prediction category. The TP, FP, TN, and FN represent the corresponding samples [Langley, Dudzik and Cloutier (2018)]. The "confusion matrix" of the classification result is shown in Tab. 5.

Table 5: Confusion matrix of the classification

Actual	Predicted Positive	Predicted Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

According to the confusion matrix above, the accuracy, precision, and recall can be defined. Accuracy is the correct proportion of all predictions and is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision is correctly predicted as the proportion of Positive which is all Positive and is defined as

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is correctly predicted as the proportion of Positive, which is all practically positive, is defined as

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Another typical metric is the F-measure, which is the weighted average of the precision and recall, defined as

$$\frac{1}{F_\beta} = \frac{1}{1+\beta^2} \cdot \left(\frac{1}{P} + \frac{\beta^2}{R} \right) \tag{4}$$

The simplified formula is

$$F_\beta = \frac{\left(1+\beta^2\right) \times P \times R}{\left(\beta^2 \times P\right) + R} \tag{5}$$

In the formula, when β>0, it measures the relative importance of recall to precision. When β=1, it is the standard F1 score, where recall and precision are considered equally important. Furthermore, β>1 means more emphasis on recall, whereas β<1 means more emphasis on precision. In our study, our value for β was 1. Substituting Eqs. (2) and (3) into Eq. (5) yields

$$F_1 = \frac{2P * R}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{6}$$

The F1 score combines the results of precision and recall. When the F1 value is high, the classification model is ideal [Saettler, Laber and Pereira (2017)]. The F1 score ranges from 0 to 1.

6 Performance evaluation

MATLAB was used for machine learning the data. The data from the first three years were used as the training set, whereas those from the fourth year is used as the test set. First, the training set was used for training; subsequently, a fully grown decision tree was generated from the original tree. Next, this classifier was used to predict the data of the fourth year, and then the decision tree was processed for 4.2 bars. The sample and layer optimization trees were generated, and these two classifiers were used to perform the predictions again. Finally, the three classifiers were compared and analyzed using the performance indicators from the previous section. The dataset after the division is shown in Tab. 6.

Table 6: The dataset

Dataset partition	Total sample size	Not registered	Registered
Complete dataset	10382	4393	5989
Training set (first three years)	6599	2889	3710
Test set (fourth year)	3783	1504	2279

6.1 Original tree

The original tree obtained after training was large as it contained 60 layers. Owing to space limitations, we show only the top part of the tree in Fig. 1.

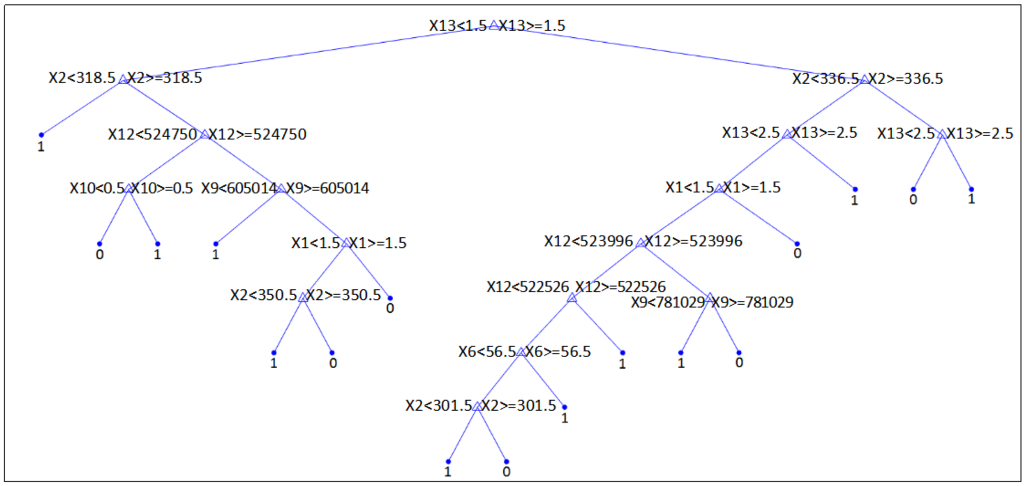

Figure 1: Part of the original tree

The meaning of the symbol x* on each node is provided in Section 6.4. The original tree was used to predict the data of the fourth year, and the confusion matrix of the classification results is shown in Tab. 7.

Table 7: Confusion matrix of classification results for original tree

Actual	Predicted Positive	Predicted Negative
Positive	1399(TP)	880(FN)
Negative	696(FP)	808(TN)

The following results can be calculated using Eqs. (1), (2), (3) and (6), as shown in Tab. 8.

Table 8: Performance metrics of original tree

Recall	Precision	Accuracy	F1
61.387%	66.778%	58.339%	0.63969

6.2 Samples optimization tree

Based on the original tree and using Algorithm 2, it was discovered that the minimum number of leaf node samples n was 110, and the corresponding error was 0.34399, as shown in Fig. 2.

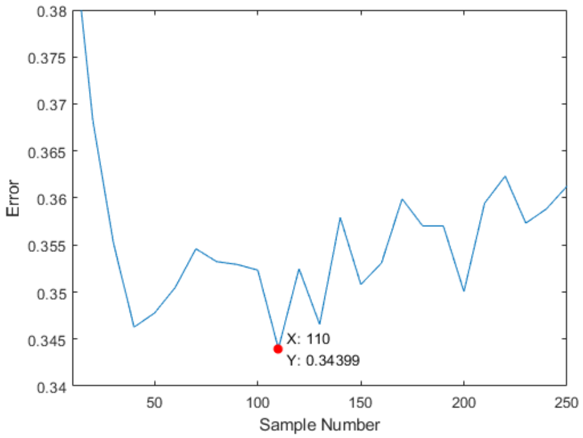

Figure 2: Relationship between sample number and error

According to this result, the sample optimization tree was obtained after pruning the original tree, as shown in Fig. 3.

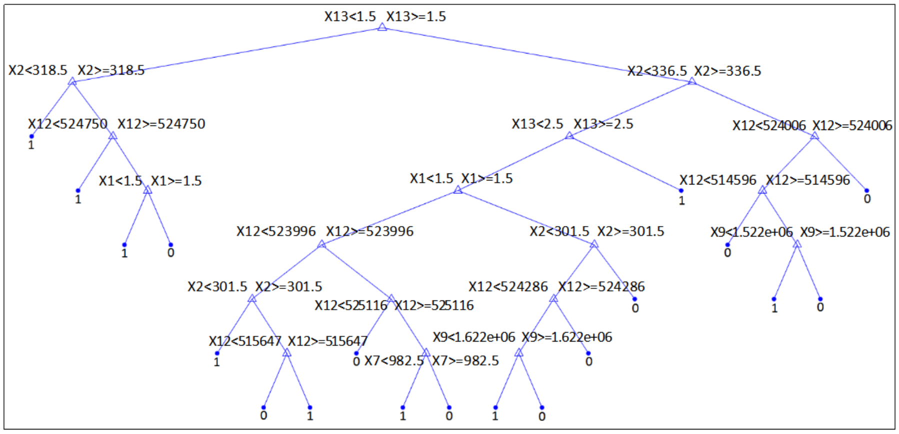

Figure 3: Sample optimization tree

Compared with the original tree, the sample optimization tress was more streamlined and was used to predict the data for the fourth year. Similar to the original tree, the following results can be calculated using Eqs. (1), (2), (3) and (6), as shown in Tab. 9.

Table 9: Performance metrics of sample optimization tree

Recall	Precision	Accuracy	F1
70.338%	65.940%	60.243%	0.68068

6.3 Layer optimization tree

Based on the original tree, the number of pruning layers was determined according to the method of Section 4.2, and the layer optimization tree obtained after pruning is shown in Fig. 4.

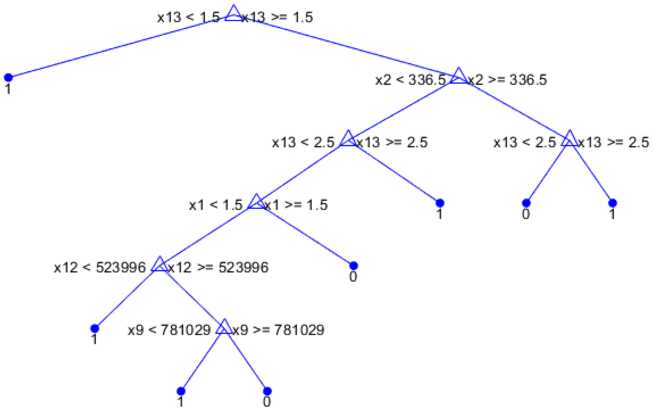

Figure 4: Layer optimization tree

This layer optimization tree was used to predict the data for the fourth year. Similarly, the following results can be calculated using Eqs. (1), (2), (3) and (6), as shown in Tab. 10.

Table 10: Performance metrics of layer optimization tree

Recall	Precision	Accuracy	F1
71.874%	68.108%	62.781%	0.6994

Compared with the original and sample optimization trees, the indicators have improved. A comparison of the indicators for the three trees is shown in Fig. 5.

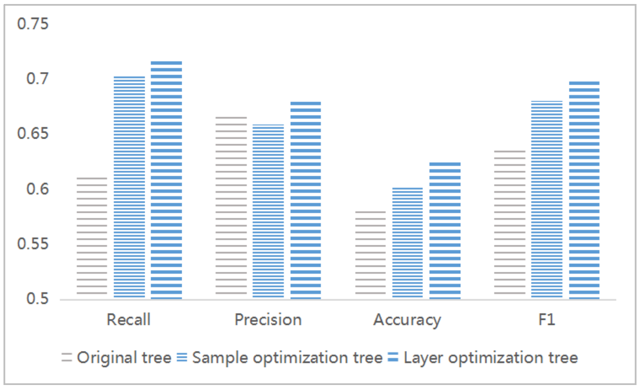

Figure 5: Comparison of three decision trees

6.4 Discussion

The meaning of the symbol x used in Figs. 1, 3 and 4 is shown in Tab. 11.

Table 11: Meaning of symbol x

Symbols	Meaning
x 13	1: Urban freshmen 2: Rural freshmen 3: Urban past students 4: Rural past students
x 2	Score
x 1	Gender: 1 male 2 female
x 12	Zip code
x 9	School code

Based on Fig. 4, to improve the registration rate, the university can consider the corresponding poverty alleviation activities. When issuing the admission notice, the publicity of the government loan system for poor students should be emphasized, and the funding system of the university should be introduced in detail to encourage students of poor economic status to register.

7 Conclusion

A new dataset was proposed herein based on the prerequisite that college freshmen would not be fully registered after admission. Unlike previous methods, a decision tree algorithm was used to predict the registration of freshmen, which proved to be feasible. After optimizing the original decision tree, the accuracy of the tree can reach approximately 63% for freshmen, and the F1 score was approximately 0.7. Nonetheless, the performance of this classification can be further improved in future studies.

Funding Statement: This work is funded in part by the National Nature Science Foundation of China (File Nos. 61872451 and 61872452) and in part by the Science and Technology Development Fund, Macau SAR (File Nos. 0098/2018/A3 and 0076/2019/A2). Li Feng is the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Alkhalid, A.; Amin, T.; Chikalov, I.; Hussain, S.; Moshkov, M. et al. (2013): Optimization and analysis of decision trees and rules: dynamic programming approach. *International Journal of General Systems*, vol. 42, no. 6, pp. 614-634.

Bertsimas, D.; Dunn, J. (2017): Optimal classification trees. *Machine Learning*, vol. 106, no. 7, pp. 1039-1082.

Langley, N. R.; Dudzik, B.; Cloutier, A. (2018): A decision tree for nonmetric sex assessment from the skull. *Journal of Forensic Sciences*, vol. 63, no. 1, pp. 31-37.

Linty, N.; Farasin, A.; Favenza, A.; Dovis, F. (2019): Detection of GNSS ionospheric scintillations based on machine learning decision tree. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 303-317.

Liu, Q.; Xiang, X.; Qin, J.; Tan, Y.; Tan, J. et al. (2019): Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. *Knowledge-Based Systems*, vol. 192, no. 1, pp. 105375-105389.

Liu, S.; Xiao, J.; Liu, J.; Wang, X.; Wu J. et al. (2018): Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 163-173.

Lopez-Chau, A.; Cervantes, J.; Lopez-Garcia, L.; Lamont, F. G. (2013): Fisher's decision tree. *Expert Systems with Applications*, vol. 40, no. 16, pp. 6283-6291.

Luo, Y.; Qin, J.; Xiang, X.; Tan, Y.; Liu, Q., Xiang, L. (2019): Coverless real-time image information hiding based on image block matching and dense convolutional network. *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125-135.

Mu, X. Y.; Remiszewski, S.; Kon, M.; Ergin, A.; Diem, M. (2018): Optimizing decision tree structures for spectral histopathology (SHP). *Analyst*, vol. 143, no. 24, pp. 5935-5939.

Mu, Y.; Liu, X.; Wang, L.; Asghar, A. B. (2019): A parallel tree node splitting criterion for fuzzy decision trees. *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, pp 1-17.

Pan, L.; Qin, J.; Chen, H.; Xiang, X.; Li, C. et al. (2019): Image augmentation-based food recognition with convolutional neural networks. *Computers, Materials & Continua*, vol. 59, no. 1, pp. 297-313.

Saettler, A.; Laber, E., Pereira, F. D. M. (2017): Decision tree classification with bounded number of errors. *Information Processing Letters*, vol. 127, no. 1, pp. 27-31.

Sempere, J. M. (2019): Modeling of decision trees through p systems. *New Generation Computing*, vol. 37, no. 3, pp. 325-337.

Shi, J.; He, Q.; Wang, Z. (2019): GMM clustering-based decision trees considering fault rate and cluster validity for analog circuit fault diagnosis. *IEEE Access*, vol. 7, no. 1, pp. 140637-140650.

Song, Y.; Zeng, Y.; Li, X. Y.; Cai, B. Y; Yang, G. B. (2017): Fast CU size decision and mode decision algorithm for intra prediction in HEVC. *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2001-2017.

Trabelsi, A.; Elouedi, Z.; Lefevre, E. (2019): Decision tree classifiers for evidential attribute values and class labels. *Fuzzy Sets and Systems*, vol. 366, no. 1, pp. 46-62.

Wang, J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Pan, N. (2019): CAPTCHA recognition based on deep convolutional neural network. *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851-5861.

Wang, W.; Jiang, Y. B.; Luo Y. H.; Li, J.; Wang X. et al. (2019): An advanced deep residual dense network (DRDN) approach for image super-resolution. *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592-1601.

Yang, J. (2010): Research and application of decision tree algorithms. *Computer Technology and Development*, vol. 20, no. 2, pp. 114-116+120.