JOURNAL OF NORTHEASTERN UNIVERSITY

Volume 25 Issue 04, 2022 ISSN: 1005-3026 https://dbdxxb.cn/ Original Research Paper

A FRAMEWORK FOR UNCERTAINTY IDENTIFICATION AND CLASSIFICATION USING DECISION TREE AND NEURAL NETWORK

Shabana Pathan[1] , Sanjeev Kumar Sharma[2]

[1]Department of Computer Science & Engineering, Oriental University, Indore, (M.P), India, sbn.pathan@gmail.com

[2]Department of Computer Science & Engineering, Oriental Institute of Science and Technology, Bhopal, (M.P), India, spd50020@gmail.com

Abstract: Inadequate or restricted data, missing data, ambiguous and noisy data are all examples of characteristics that might lead to data set uncertainty. As the datasets used in this research showed, when a data set is noisy or ambiguous, the result is uncertainty. Real-world datasets are far from perfect: they are typically affected by various types of uncertainty (including missing values) that are primarily related to either the data collection technique or the complexity (e.g., volatility) of the phenomena under investigation, or both. These kinds of uncertainties are usually classified into two groups: Data that isn't there and supervision that isn't up to standard. In this research, goal is to handle uncertainty using various techniques while also disambiguating the uncertain situations. In this paper, a framework for uncertainty estimation and handling using several strategies is proposed. The optimal decision tree (ODT) approach is given for increasing the predictive performance of any model when the ML model is selected as a decision tree. Precision, recall, f-measure, and accuracy are some of the evaluation parameters used to evaluate the proposed ODT method. For handling uncertainty, an optimum strategy is identified based on various comparisons.

Keywords: Uncertainty, Decision Tree, Classification, Gini Impurity and Optimal Decision Tree (ODT).

I. INTRODUCTION

Data mining is useful in a variety of industries, including medical, marketing, finance, power, banking, manufacturing, and telecommunications. However, the data in these real-world applications is unreliable, confusing, inconsistent, and noisy, adding to the uncertainty. In addition, uncertainty emerges as a result of missing or insufficient data. Because of the ambiguities in the data, we have poor information, which makes data mining activities more difficult. Handling these uncertainties in data is a vital responsibility or task for a decision-making system to perform intelligent data analysis.

Machine learning, soft computing, pattern recognition, statistics, artificial intelligence, data visualization, and reasoning with uncertainty are just some of the ideas, algorithms, and methodologies used in data mining and decision making processes. To deal with data uncertainties, soft computing techniques were utilized to create data mining models. "The

Submitted: 21/10/2022 **Accepted**: 24/11/2022

guiding idea of soft computing is to utilize the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, resilience, cheap solution cost, and improved rapport with reality," said Professor Zadeh, the pioneer of soft computing(Zadeh, 1996, Yager et al., 1994) . It differs from traditional (hard) computing in that it allows for imprecision, uncertainty, partial truth, and approximation, which traditional (hard) computing does not (Ibrahim, 2016).

1.1 Motivation : A meta-cognitive framework based on transformation techniques, missing data treatment, and classification models is used to simulate uncertainty issues in categorical and numerical data in decision making systems. A decision tree is a basic but effective classifier for extracting meaningful data from enormous datasets. To increase the performance of a decision tree, many algorithms have been created. When dealing with large amounts of data, many academics choose to use a decision tree for classification. Apart from the difficulty of implementation, it has a high level of accuracy when compared to other categorization systems. It builds a supervised model to demonstrate the relationship between instances and characteristics (Han et al., 2006).

1.2 Problem Statement : Prediction accuracy for unknown instances is used to evaluate decision tree performance. This research intends to preprocess data in such a way that it produces optimized decision trees with high classification accuracy while also reducing decision tree size. Some parts of the decision tree are addressed and improved in this study. To improve prediction performance, the algorithms should generate accurate and optimal decision trees. It also aspires to develop a globally optimized decision tree algorithm with high prediction accuracy and a small footprint.

1.3 Challenges : The classification algorithm produces good results and accuracy in terms of target value prediction, but it can also produce inefficient or erroneous predictions. As a result, we propose optimal decision trees to improve the classifier's prediction performance. The following issues arose during the implementation of the optimal decision tree:

a) Find best splitter node
b) Handling Uncertainty in data

1.4 Our Contribution : Both of the difficulties are addressed in our suggested approach. Missing data treatment and data transformation strategies are two approaches to dealing with data uncertainty. The Gini impurity is also used to determine the optimum splitter node. Following the resolution of these issues, the dataset can be categorized using an optimal decision tree (ODT). The parameters and Gini impurity are used to create an optimal decision tree. Finally, we used precision, recall, f-measure, and accuracy to evaluate the best decision tree algorithm. Finally, we discovered the most effective optimization algorithm.

II. LITERATURE SURVEY

Various academics have worked on decision tree-based algorithms to improve performance, but there is still something missing, such as how to handle uncertainty in data using decision trees. The following is a summary of some relevant work-

Traditional decision tree classifiers operate on data with known and accurate values. Tsang et al. improved these classifiers to handle data containing ambiguous information, such as measurement/quantization mistakes, data staleness, many repeated measurements, and so on. Multiple values form a probability distribution function to describe value uncertainty (pdf). They discovered that taking into account the entire pdf rather than just a single statistic improves the accuracy of a decision tree classifier significantly. They improved on traditional decision tree-building algorithms to deal with data tuples with ambiguous values. They presented a number of pruning approaches that can considerably increase the efficiency of the development of decision trees because processing pdfs is computationally more expensive(Tsang et al., 2009).Because of the uncertainties in the data, many traditional systems are ineffective at handling data mining jobs. When the input pattern is ambiguous and the classes are overlapping or ill-defined, uncertainty occurs, especially in decision-making systems with many categories or decision classes. Classifiers are decision-making systems that attempt to accurately classify or map an input object into two or more decision classes. There is a constant demand for multi-class classifiers that can be used to solve real-world problems, but most classification learning algorithms are essentially binary, and their extension for multi-class classification is elaborate and costly, resulting in performance degradation(Fernández et al., 2013, Kang et al., 2015).

Liang and colleagues introduced a decision tree for dynamic and unpredictable data streams. Current research on data stream categorization focuses mostly on specific data, where a precise and definite value is typically expected. However, data with uncertainty is ubiquitous in real-world applications due to a variety of causes such as faulty measurement, recurrent sampling, and network difficulties. The classification of unclear data streams was the subject of this study. Based on CVFDT and DTU, we created the UCVFDT (Uncertainty-handling and Concept-adapting Very Fast Decision Tree) algorithm, which not only maintains CVFDT's ability to cope with concept drift at high speeds, but also adds the ability to handle data with uncertain properties. The suggested UCVFDT algorithm is effective in identifying dynamic data streams with uncertain numerical attributes and is computationally efficient, according to an experimental research(Liang et al., 2010).For forecasting students' academic achievement, Kolo and Adepoju presented a decision tree approach. Education is the foundation upon which a society's members' quality of life improves. To increase the quality of education, it is necessary to be able to forecast pupils' academic success. The Chi-Square Automatic Interaction Detection (CHAID) is applied in the decision tree structure using the IBM Statistical Package for Social Studies (SPSS). Students' financial situation, motivation to learn, and gender have all been found to have an impact on their performance. Sixty-six percent of students were expected to pass, while 33.2 percent were expected to fail. It was discovered that a far higher percentage of students were likely to pass, and that male students were more likely to pass than female students(Kolo and Adepoju, 2015).

Xia et al. suggested a sequential ensemble credit scoring model using a variant of the gradient boosting machine- XGBoost. Three phases make up the majority of the model. Data preprocessing is utilised first, followed by the removal of redundant variables using a model-based feature selection technique based on relative feature significance ratings in the second stage. Third, the hyper-parameters of XGBoost are adaptively modified and used to train the model using a specified feature subset using Bayesian hyper-parameter optimization. The experiment uses a variety of hyper-parameter optimization approaches using baseline classifiers as reference points. According to the findings, Bayesian hyper-parameter optimization beats random, grid, and manual search procedures. Furthermore, on four assessment criteria, the proposed model outperforms baseline models: accuracy, error rate, AUC-H measure, and Brier score (Xia et al., 2017). To deal with uncertainty in Machine Learning, Campagner developed the three-way decision (TWD) framework and the trisecting-acting-outcome model (ML). They separated between dealing with uncertainty in ML models' inputs, where TWD is used to identify and account for uncertain situations, and dealing with uncertainty in ML models' outputs, where TWD is used to allow the ML model to abstain. They then give a narrative review of the current state of TWD applications in relation to the framework's multiple areas of concern, stressing both the three-way technique's strengths as well as future research opportunities (Campagner et al., 2020).

Medical disease analysis, text categorization, user smartphone classification, pictures, and a variety of other disciplines have all employed decision tree classifiers. The decision trees were thoroughly examined by Charbuty and Abdulazeez. Furthermore, the study's contents, including the methodology utilised, datasets used, and results obtained, are thoroughly analysed and disputed. In addition, all of the methodologies were examined in order to determine the most accurate classifiers and to demonstrate the writers' themes. As a result, the applications and effects of various types of datasets are studied (Charbuty and Abdulazeez, 2021). Based on decision trees and tabu search approaches, Hafeez et al. suggested a new and robust classifier. In order to increase performance, their proposed approach constructs numerous decision trees while employing a tabu search algorithm to continuously monitor the leaf and decision nodes in the related decision trees. The tabu search method is also in charge of balancing the entropy of the decision trees in question. They trained the system to detect whether a patient is suffering using clinical data from COVID-19 patients. Their proposed classifier, which is based on Python's built-in sci-kit learn package, was used to acquire the experimental findings. Big O and statistical analysis for standard supervised machine learning algorithms were used to conduct a thorough examination for the performance comparison. Furthermore, the classifier's performance is compared to those of optimised state-of-the-art classifiers. As indicated by the achieved accuracy of 98 percent, the required execution time of 55.6 ms, and the area under receiver operating characteristic (AUROC) for the proposed technique of 0.95, the proposed classifier algorithm is acceptable for large datasets (Hafeez et al., 2021).

Zhao et al. devised and implemented a personal credit evaluation method based on a decision tree with a boosting algorithm. When compared to the traditional decision tree technique, it can be seen that the boosting algorithm reduces processing time. The boosting algorithm increased the accuracy of the Classification and Regression Tree (CART) to 90.95 percent, slightly higher than the 90.31 percent accuracy without boosting. They investigated cross-validation and demonstrated the findings with simulation to avoid overfitting the model on the training set due to illogical data set division; hypermeters of the model were applied, and the model fitting effect was validated. A confusion matrix is used to find the best fit for the specified decision tree model. Relevant assessment measures are also presented here in order to evaluate the performance of the proposed model. The obtained result demonstrates that when the boosting strategy is used, the decision tree model's performance improves (Zhao et al., 2021).

III. PROPOSED FRAMEWORK

In this paper, a framework for uncertainty estimation and handling using several strategies is given, as shown in fig.1. Uncertainty estimation, uncertainty handling, optimal decision tree, and performance evaluation are the four phases of the framework. The raw dataset is first applied to the uncertainty estimation phase, where deep learning models are utilized to identify dataset uncertainty. If there is uncertainty in the dataset, it is resolved during the uncertainty handling phase, which employs a variety of strategies such as missing data treatment and transformation techniques. After the data has been cleaned, the next step is to classify it using a decision tree classifier, which uses parameter optimization and impurity optimization methods to get accurate results. The performance of the algorithms is assessed in the following phase using several metrics such as accuracy, precision, recall, and f-score. For handling uncertainty, an optimum strategy is identified based on various comparisons. The flow of the entire process is described in the algorithm, which is discussed in the next section. Below is a full description of each phase.

- **Uncertainty Estimation:** Epistemic uncertainty is one of the types of uncertainty which occurs due to limited data and knowledge gain at the time of data collection and observation. When training samples are fewer there can be chances of more uncertainty. So this type of uncertainty can be resolved by giving enough training samples. Input data dependent uncertainty called as heteroscedastic uncertainty which can be resolved by improving input data. These types of uncertainties can be estimated using different deep neural models which uses Bayesian statistics. Bayesian statistics allows obtaining conclusions based on both data and our prior comprehension about the fundamental trends. Uncertainty can be assessed using dropout function in deep learning in which parameters are distributed as a substitute of predetermined weights. Instead of learning the model's parameters, a distribution of weights could be learned, allowing for the estimation of uncertainty. Deep Ensembling is a strong technique in which a huge number of models or re-multiple copies of a model are trained on different datasets and their predictions are combined to create a predictive distribution. Because assembling can necessitate a lot of computing power, a different solution was suggested: A model ensemble's dropout as a Bayesian approximation. Dropout is a common method in deep learning as a regularizer to avoid overfitting. It comprises of selecting network nodes at random and discarding them during training. Dropout eliminates neurons at random using a Bernoulli distribution. In Bayesian models, there appears to be a major connection between regularization and prior distributions. Dropout isn't the only case in point. The L2 regularization, which is widely used, is effectively a Gaussian prior. A Bayesian approximation of the Gaussian process is mathematically equal to a deep neural network with dropout applied before each weight layer. Each subset of nodes that is not dropped out defines a new network with droupout. The training process can be thought of as simultaneously training two or more models, where m is the number of nodes in the network. A randomly picked set among these models is trained for each batch. The fundamental concept is to drop out throughout both training and testing. The research advises that at test time, you repeat the prediction a few hundred times using random dropout. The estimate is the average of all predictions. We simply calculate the deviation of predictions for the uncertainty interval. This determines the ensemble's level of uncertainty.

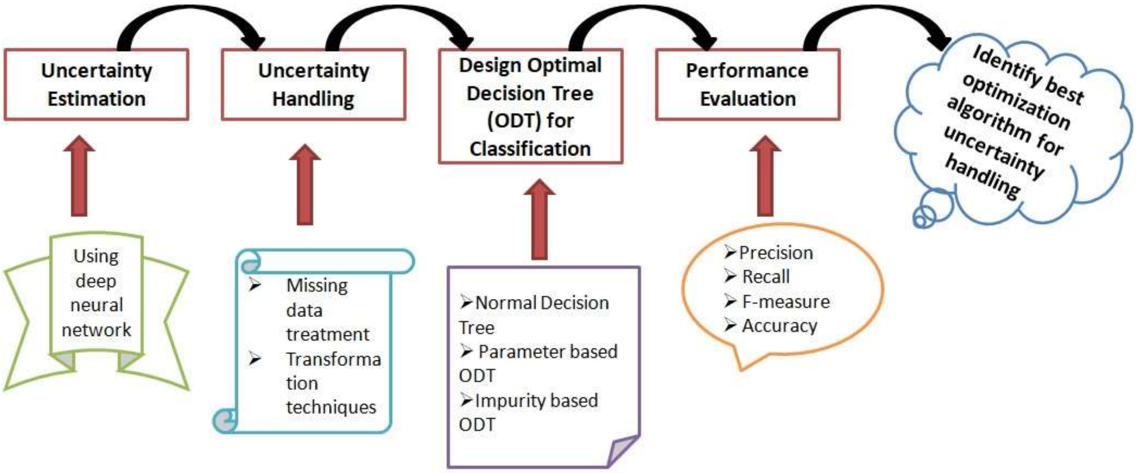

Figure.1 Framework for Uncertainty Handling

- **Uncertainty Handling:** When there is uncertainty in a dataset, it can be resolved using a variety of strategies, including missing data treatment and transformation techniques. Cleaning and replacing any NAN values requires missing data handling. Missing values in categorical data can be replaced by the column's mode value, and missing values in numerical data by the mean and median values. For more exact findings, transformation techniques such as standard scaler, maximum absolute scaler, MinMax scaler, and L2 normalization are used separately after missing data treatment.

- **Optimal Decision Tree:** The decision tree classifier is applied to the dataset after the transformation results have been obtained. In this phase, three types of decision trees are used: a simple decision tree, a decision tree with parameter optimization, and a decision tree with gini index impurity optimization. The purpose of this phase is to develop a more efficient way for dealing with uncertainty that affects classification results.

.

- **Performance Evaluation:** The performance of the system is evaluated using different measures such as accuracy, precision, recall and f-score. Accuracy shows the model's overall accuracy, which is the percentage of total samples correctly identified by the classifier. Precision indicates what percentage of predictions that were labelled as positive classes were in fact positive. Recall shows the percentage of all positive samples the classifier accurately predicted as positive. F-score is a measure that unites recall and precision. It's just the harmonic mean of precision and recall in mathematics.

3.1 Algorithm

The algorithm for uncertainty handling is mentioned below. The input is raw dataset in which uncertainty should be identified and resolved if exists using different techniques. The output of algorithm is optimized method for decision tree classification with different performance measures.

Algorithm: Optimized Algorithm for Uncertainty Handling

```
Inputs: Raw dataset
Output: Optimal decision tree,  Accuracy, Precision, Recall, F-Score
1.Dataset collection
  Df →load dataset()
2. perform uncertainty estimation using neural network (NN)
  Create NN architecture with dropout and without dropout layer
  Plot uncertainty on data
•    model_without_dropout = architecture(layers_shape=[5,10,20,10,5], input_dim= 1, output_dim=1,
                    dropout_proba=0, reg=0, act='relu', verbose=1)

•    model_with_dropout = architecture(layers_shape=[5,10,20,10,5], input_dim= 1, out put_dim=1,
                    dropout_proba=0.05, reg=0.00475, act='relu', verbose=1)
3. Pre-process dataset
4. if (missing data exists):
      treat_missingData()
     apply replacer()
   Else:
     Proceed cleaned_data()
5. Apply transformation techniques
   ss =   StandardScaler()
          mm = MinMaxScaler()
          ma =  MaxAbsScaler()
          ns =   Normalizer(norm = 'l2')
6. split dataset into train and test data with random state
```

```
x_train, x_test, y_train, y_test = train_test_split (x,y,test_size = 0.3, random_state= 21)
7. Apply decision tree classifier
model = DecisionTreeClassifier()
8. Apply decision tree classifier with parameter optimization
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)
9. identify accuracy, precision, recall, f-score, execution time
•       accuracy = accuracy_score (y_test, model.predict(x_test))
•       precision = precision_score (y_test, model.predict(x_test))
•       recall  = recall_score (y_test, model.predict(x_test))
10. Perform comparison between with uncertainty results and without uncertainty
11. Identify optimized algorithm for uncertainty handling
```

IV. RESULTS AND DISCUSSIONS

The proposed framework evaluates the epistemic uncertainty of a regression problem by utilising data that was generated by adding noise with a normally distributed to the function y=x in the following manner:

Between the coordinates x=-2 and x=-3, the left cloud receives the generation of 100 data points. In the right cloud, 100 data points are produced between the values of x=2 and x=3 respectively. The noise that is introduced to the left cloud has a variance that is ten times higher than the noise that is added to the right cloud. The figure 2 shows the data with uncertainty on different data points.

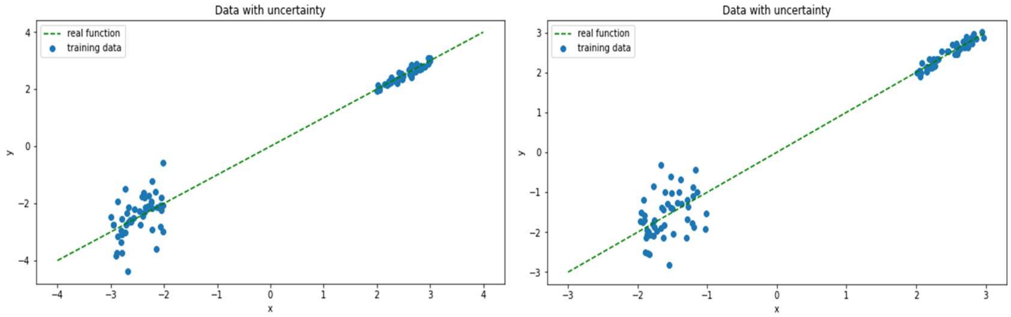

Figure 2. Predicting Epistemic Uncertainty on different data points

The experiments show the construction of two basic neural networks: the first will have no dropout layers between the hidden layers, while the second will have one, the performance shown in figure 3. During each training and inference batch, the dropout layer is responsible for randomly turning off 5% of the neurons. During optimization, we additionally make use of L2 regularizers in order to impose penalties on layer parameters. When training batches of 10 points, the rmsprop optimizer is utilised to achieve the goal of decreasing the mean squared errors. The results of the training are presented in the following table. Both models converge quite quickly on the same solution. The model that includes dropout displays a little larger loss along with a more random behaviour. This occurs because, during training, random portions of the network are turned off, which causes the optimizer to skip over areas of the loss function that have local minima.

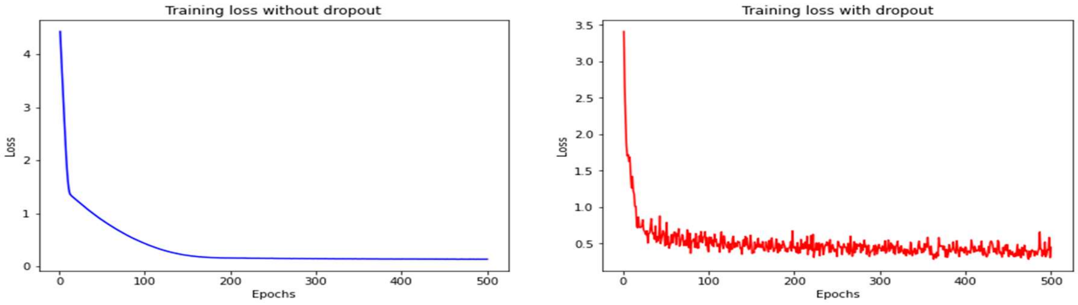

Figure 3. Neural network model performance with dropout and without dropout for uncertainty estimation

In the following, we will demonstrate how the models fare when applied to test data. The model that does not account for dropout produces accurate predictions in the form of a straight line with a flawless R2 score. The inclusion of dropout resulted in a nonlinear prediction line that had a score of 0.79 for the R2 statistic. Even though dropout overfits less, has larger bias, and decreases accuracy, it shows uncertainty in predictions in the regions that do not have training samples. The prediction line has a higher variation in such places, and this variance can be used to determine the epistemic uncertainty.

V. CONCLUSION

This study proposes a framework for estimating and managing uncertainty using a variety of tactics. The framework's four steps are uncertainty estimation, uncertainty handling, optimal decision tree, and performance evaluation. Deep learning models are used to identify dataset uncertainty once the raw dataset is submitted to the uncertainty estimation step. If there is any uncertainty in the dataset, it is resolved during the uncertainty handling phase, which uses several procedures such missing data treatment and transformation approaches to resolve it. Decision tree classification is used to categorize the data with parameter optimization and impurity optimization approaches to provide accurate results. The algorithms' performance is evaluated using a variety of measures such as accuracy, precision, recall, and f-score. Based on numerous comparisons, an optimal technique for dealing with uncertainty is established. Based on the performance it is concluded that on most of the dataset impurity optimization method performed best. Uncertainty estimation using deep neural model with dropout mechanism gave better identification which is concluded based on the accurate results.

References

1. Campagner, A., Cabitza, F. & Ciucci, D. Three-way decision for handling uncertainty in machine learning: A narrative review. International Joint Conference on Rough Sets, 2020. Springer, 137-152.

2. Charbuty, B. & Abdulazeez, A. 2021. Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2, 20-28.

3. Fernández, A., López, V., Galar, M., Del Jesus, M. J. & Herrera, F. 2013. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowledge-based systems, 42, 97-110.

4. Hafeez, m. A., rashid, m., tariq, h., abideen, z. U., alotaibi, s. S. & sinky, m. H. 2021. Performance improvement of decision tree: A robust classifier using tabu search algorithm. Applied Sciences, 11, 6728.

5. Han, j., kamber, m. & mining, d. 2006. Concepts and techniques. Morgan Kaufmann, 340, 94104-3205.

6. Ibrahim, d. 2016. An overview of soft computing. Procedia Computer Science, 102, 34-38.

7. Kang, s., cho, s. & kang, p. 2015. Constructing a multi-class classifier using one-against-one approach with different binary classifiers. Neurocomputing, 149, 677-682.

8. Kolo, d. K. & adepoju, s. A. 2015. A decision tree approach for predicting students academic performance.

9. Liang, C., Zhang, Y. & Song, Q. Decision tree for dynamic and uncertain data streams. Proceedings of 2nd Asian Conference on Machine Learning, 2010. JMLR Workshop and Conference Proceedings, 209-224.

10. Tsang, S., Kao, B., Yip, K. Y., Ho, W.-S. & Lee, S. D. 2009. Decision trees for uncertain data. IEEE transactions on knowledge and data engineering, 23, 64-78.

11. Xia, Y., Liu, C., Li, Y. & Liu, N. 2017. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. Expert Systems with Applications, 78, 225-241.

12. Yager, R. R., Zadeh, L. A., Kosko, B. & Grossberg, S. 1994. Fuzzy sets, neural networks, and soft computing.

13. Zadeh, L. A. 1996. Soft computing and fuzzy logic. Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi a Zadeh. World Scientific.

14. Zhao, L., Lee, S. & Jeong, S.-P. 2021. Decision Tree Application to Classification Problems with Boosting Algorithm. Electronics, 10, 1903.

15. Kumar, Ram and Patil, Manoj, Improved the Image Enhancement Using Filtering and Wavelet Transformation Methodologies (July 22, 2022). Available at SSRN: https://ssrn.com/abstract=4182372

16. Kumar, R., Singh, J.P., Srivastava, G. (2014). Altered Fingerprint Identification and Classification Using SP Detection and Fuzzy Classification. In: , et al. Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012. Advances in Intelligent Systems and Computing, vol 236. Springer, New Delhi. https://doi.org/10.1007/978-81-322-1602-5_139