UMEÅ UNIVERSITY

An ensemble learning approach based on decision trees and probabilistic argumentation

Istiak Ahmed

Istiak Ahmed
Spring 2020
Degree Project in Computing Science, 30 credits
Supervisor: Juan Carlos Nieves Sanchez
Examiner: Jerry Eriksson
Master's Program in Computing Science, 120 credits

Abstract

This research discusses a decision support system that includes different machine learning approaches (e.g. ensemble learning, decision trees) and a symbolic reasoning approach (e.g. argumentation). The purpose of this study is to define an ensemble learning algorithm based on formal argumentation and decision trees. Using a decision tree algorithm as a base learning algorithm and an argumentation framework as a decision fusion technique of an ensemble architecture, the proposed system produces outcomes. The introduced algorithm is a hybrid ensemble learning approach based on a formal argumentation-based method. It is evaluated with sample data sets (e.g. an open-access data set and an extracted data set from ultrasound images) and it provides satisfactory outcomes. This study approaches the problem that is related to an ensemble learning algorithm and a formal argumentation approach. A probabilistic argumentation framework is implemented as a decision fusion in an ensemble learning approach. An open-access library is also developed for the user. The generic version of the library can be used in different purposes.

Acknowledgements

I would like to thank my supervisors Juan Carlos Nieves and Christer Grönlund, who has contributed expertise in computer science topics and medically related topics, granting access to data used, and introducing the idea for this research, as well as providing assistance by correcting this report.

Contents

1 Introduction

Machine learning approaches have a considerable contribution to develop different types of decision support systems. Learning algorithms are important for classifying data and predicting outcomes based on training data [27]. A learning algorithm is an algorithm used in machine learning to assist applications to imitate the human learning process [14]. It can learn something new from a given situation. On the other hand, the symbolic reasoning is an operation of cognition that allows rules, axioms, facts, etc to generate new knowledge from existing knowledge [1]. The combined effort of the learning algorithms and symbolic reasoning can produce better outcomes [7] [18]. Whereas, it is not possible for a single approach. Many studies have been done for developing decision support systems in medical science. Several artificial intelligence techniques and machine learning approaches, for instance, neural network, decision tree algorithm, multi-agents system, etc are implemented to develop a system that can easily detect dangerous diseases [15] [22] [19]. Nevertheless, there are some limitations, e.g., produce conscious outcomes, produce outcomes in case of data overlapping. The general problem statement of this research is to develop a system for the sake of resolving the limitations of previous researches that can produce conscious and satisfactory outcomes for practical usage, e.g., medical decision support systems, by implementing an ensemble architecture that includes different approaches. The research presented by Kahneman et al. (2010) introduces system 1 and system 2 [17] where system 1 produces unconscious outcomes based on training data and system 2 produces conscious outcomes through a decision making process. In this research, a learning algorithm and a probabilistic argumentation framework (PAF) are implemented as the learning approach and symbolic reasoning approach in an ensemble architecture. The learning approach and symbolic reasoning approach are implemented as system 1 and system 2 respectively. An ensemble learning algorithm (ELA) usually executes a base learning algorithm multiple times and creates a voting classifier for the resulting hypothesis [5]. ELA provides more accurate outcomes than any of its component models. It is beneficial to implement an ensemble learning approach in this specific problem. An ensemble learning approach produces better outcomes than any single learning algorithm [24].

In this thesis, a data set of ultrasound images is used to evaluate the introduced ELA approach. Moreover, decision tree algorithms (DTA) are used as the base learning algorithm of ELA that generates outcomes by classifying data set and PAF generates more accurate outcomes from the existing outcomes. DTA classifies data set from ultrasound images and generates satisfactory output with the help of the probabilistic argumentation framework. Sample data set such as an open access data set is used for evaluation of the introduced ensemble learning approach. DTA classifies data set into three different groups. Classification is a process to divide data into several groups or categories based on different decision rules. Data classification is important to categorize data set for prediction. The proposed system of combining ELA and PAF is designed to diagnosis cardiovascular diseases. Cardiovascular disease atherosclerosis is a life-threatening disease. It amasses plaque in the carotid and coronary corridor divider, which can prompt dangerous conditions, for example, brain stroke and heart stroke [16]. It is important to develop an efficient system that can recommend the specific diagnosis to a patient by analyzing data set from ultrasound images of carotid and coronary arteries. Machine learning is useful for solving problems in the medical sector that provide promising outcomes. It is interesting to implement an argumentation approach and an ensemble learning algorithm in medical science for solving puzzles. An argumentation approach has been used for modeling aggregation and decision making [20]. The combining effort of both approaches produces accurate outcomes that are essential for medical decision support systems. Furthermore, it is an innovative idea to implement ELA and PAF in medical decision support systems. On the other hand, It is difficult to get accurate outcomes by implementing only machine learning approaches for detecting stroke diseases by analyzing images of the carotid and coronary arteries. The accuracy of the system depends on the machine learning model and data set.

The proposed system implements DTA for classifying the sample data set. The different approaches of DTA work like different models and they produce different outcomes. PAF generates an outcome combining the outcomes of different models. The main concern of this study is to combine the outcomes from several models of a learning algorithm (DTA) and produce feasible outputs through an argument process (PAF). The PAF works combined with DTA to generate more accurate outcomes. Both of the learning algorithm and the argumentation framework are crucial for this research. PAF produces more accurate output by using the outcomes from different models of DTA. The contribution of this research is to integrate PAF with ELA for obtaining feasible outcomes. The different models of DTA work like a multi-agents system and the agents argue through PAF for producing an accurate and satisfactory result. A pipeline architecture is implemented to combine the learning algorithm and the argumentation framework. An argumentation framework works as a decision fusion in the ensemble learning approach that combines outcomes from multiple models.

1.1 The aim of this research

This study aims to introduce a probability argumentation approach as a decision fusion in an ensemble architecture. The probability argumentation approach combines outputs generated from different models of an ensemble learning approach and produces final outputs. The research question is given below.

Research question: How to define an ensemble learning approach based on formal argumentation?

1.2 Organization of this research

The rest of the paper is designed as follows: chapter 2 presents the working procedure of ELA, DTA, and PAF; chapter 3 describes some related works in the literature; chapter 4 presents a pipeline architecture and the algorithm being used; chapter 5 analyzes the experimental results; chapter 6 discusses an open-access library, chapter 7 discusses the discussion and finally, chapter 8 presents conclusions and future work.

2 Background

This chapter discusses different approaches of ELA, DTA, and PAF. It also explains the working procedure of each approach.

2.1 Ensemble Learning Algorithm (ELA)

An ensemble learning algorithm (ELA) conducts a base learning algorithm multiple times and produces an outcome. A decision fusion technique such as a voting classifier helps to elect the optimal outcome. ELA follows a different approach such as it finds a set of hypotheses rather than finding a best hypothesis to explain the data and the hypotheses participate in a voting process [5]. The voting process selects an optimal hypothesis to illustrate the data. More definitely, an ensemble approach constructs a set of hypothesis $\{k_1, k_2, ..., k_n\}$, picks a set of weight $\{w_1, w_2, ..., w_n\}$ and constructs a voting process $H(x) = \{w_1 k_1(x) + ... + w_n k_n(x)\}$. If $H(x) \geq 0$ then the voting process will return +1 otherwise, it will return -1. The base learner components generate the hypotheses by executing a base learning algorithm which can be neural network, decision tree, or other kinds of machine learning algorithms [28]. A number of base learners can be constructed in a parallel or sequential style. The efficiency of an ensemble method depends on the outcomes of base learners. The outcomes should be more exact and more diverse as possible. Several processes are available such as cross-validation, hold-out test, etc for calculating the accuracy of the outcomes. There are two distinct ways to design ELA. The first way is to construct the hypotheses independently that the resulting set of hypotheses is accurate and diverse. That is why each hypothesis has a low error rate for making predictions. Such an ensemble of hypotheses ensures a more accurate prediction than any of its component classifiers. According to the second way, to construct the hypotheses in a coupled fashion that the weighted vote of the hypotheses provides a better structure to the data. Breiman(1996) introduces a method called Bagging (Bootstrap Aggregation) [5] [24] [2]. In this method, there is a given data set m. The different samples of data set $\{d_1.....d_n\}$ will be provided to the different models (base learners) $\{m_1.....m_n\}$ of the base learning algorithm. The size of each sample data set $\{d_1.....d_n\}$ is less than the source data set m. The models generates a set of result $\{r_1.....r_n\}$. The generated result from each model or base learner will be considered a vote. Then the voting classifier combines all of the different votes from several models and produces a final result. The result from the voting classifier is finalized based on the majority of the voting [2]. The following example explains, how does the voting classifier produce an output. For instance, five different models $(m_1, m_2, m_3, m_4, m_5)$ exist in an ensemble approach. The models predict outcomes based on classified data and training data. The class label contains two different values, one is YES and the other is NO. A total of three models (m_1, m_3, m_5) predict YES as outcomes and the rest of the models (m_2, m_4) predict NO as outcomes. The voting classifier calculates the majority of voting and it produces YES as an output. The number of the vote as YES is greater than NO and that is why it generates YES as an output.

2.2 Decision Tree Algorithm (DTA)

Decision tree algorithm (DTA) is a tree-shaped diagram used to determine a course of action and each branch of the tree represents a possible decision. DTA is suitable for recommending and predicting based on training data set. In generally, DTA constructs a training model which can use to predict class label or value of target variables. A training model consists of a set of training data and a set of decision rule and the model predicts value by learning decision rules. A training data set $TS = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where, $x_i \in X$ and $y_i \in Y$ $(i = 1, ..., n)$. DTA starts processing from the root node for predicting a class label. It compares the values of the root attribute with the record's attribute. Then the algorithm follows the branch corresponding to that value and jumps to the next node. It also compares the record's attribute values with other internal nodes of the decision tree. The process will be continued until reach the leaf node. The leaf node contains predicted class value. A decision tree follows a disjunctive normal form that presents the sum of product. Every branch from the root node to the leaf node has the conjunction (product of values) class and different branches ending in the disjunction (sum) class. The greatest challenge in DTA is to find out the attribute as a root node. Different attribute selection techniques such as information gain, Gini index, and so on can be implemented to identify the root node at each level. In the proposed research entropy is implemented as an attribute selection technique.

DTA has two approaches for implementation one is decision tree classifier and another is decision tree regression [23]. In the classification approach, the target variable can be a discrete set of values. Furthermore, the tree produces a categorical solution like true or false, yes or no, 0 or 1 and in regression, the tree predicts values analyzing the values from previous states. In the regression approach, the target variable can take continuous values. Algorithms like Naive Bayes, Logistic Regression, K-Means can work in classification problems. These algorithms are perfect for working with simpler data [12]. However, DTA is suitable to work in a classification problem with large and complex data sets. It is simple to understand, interpret, visualize and little effort required for data preparation. Decision Trees can deal with both numerical and the group of data [23]. The decision tree algorithm follows the greedy approach [27]. It uses a top-down recursive method for tree construction. The ID3 algorithm is implemented in the proposed research. The algorithm constructs decision trees implementing a top-down greedy search approach. The tree has several decision nodes and each decision node has two or more branches. It starts with the sample data set as the root node. Then it calculates the entropy of each attribute. It selects the attribute based on the highest or lowest entropy. The selected attribute splits the sample data set into smaller subsets. It is a recursive process. The algorithm always selects the attributes that never selected before. The leaf node represents a decision.

2.2.1 Entropy

Entropy is an important term in DTA. Entropy is the measurement of randomness and unpredictability in the data set. It is high in the decision or root nodes of the Decision Tree and it decreases gradually in the child nodes [23]. Entropy ensures the splitting points of a data set. DTA picks a node with the highest information gain to construct or split a decision Tree. Entropy helps to select the right attribute or node for constructing a decision tree.

The lowest entropy is calculated by the following equation (2.1).

$$Entropy = \sum_{i=1}^{e} P_i(-log_2 P_i) \tag{2.1}$$

Let P_i be a discrete random variable that takes values in the domain $\{P_1,..., P_e\}$.

The overall entropy is calculated by the following equation (2.2).

$$OverallEntropy = \sum_{j=1}^{2} P_j(lowestEntropy) \tag{2.2}$$

The overall entropy determines the best splitting nodes.

2.3 Argumentation Framework

Argumentation is the process of how decisions can be decided through logical reasoning. It includes deliberation and negotiation that are important for a decision-making process. Two or more agents can participate in an argumentation process and finally, a decision can be made through an argumentation process. The strength of an argument depends on the social values and one argument attacks another. The success of an argument depends on the comparative strength of the values [1]. The argumentation framework is a pair of arguments and attacks. $AF = (AR, attacks)$ where AR is a set of arguments and attacks is a binary relation AR [6]. For example, there are two arguments A and B. The meaning of $attacks(A,B)$ is that A attacks on B. A set of arguments S attacks on B that means B is attacked by an argument that exists in the arguments set S. The important thing is whether a given argument should be accepted or unaccepted [6]. The example below (Fig.1) illustrates more clearly the working method of an argumentation approach. For instance, A is a set of abstract elements, and R is a set of attacks. The argumentation system $S = (A,R)$, where, $A = \{a,b,c,d\}$, and $R = \{(a,b),(b,c),(d,c)\}$ (Fig.1). A contains four arguments and R contains three attacks (a attacks b, b attacks c, d attacks c). An argument $a \in A$ is acceptable with respect to $E \in A$ if E defends a, that is $\forall b \in A$ such that $(b,a) \in R$, $\exists c \in E$ such that $(c,b) \in R$. A set of argument E is conflict-free if there is no attack between its arguments $\forall a,b \in E$, $(a,b) \notin R$. An argument set E is admissible if and only if it is conflict-free and all its arguments are acceptable with respect to E. Argument labeling is a way to express the status of an argument such as where it is accepted (in), rejected (out), or undefined (undec). The following statements explain the status of an argument. $\forall a \in A$, $L(a) = in$ if and only if $\forall b \in A$ such that $(b,a) \in R$, $L(b) = out$. $\forall a \in A$, $L(a) = out$ if and only if $\exists b \in A$ such that $(b,a) \in R$, $L(b) = in$. $\forall a \in A$, $L(a) = undec$ if and only if $L(a) \neq in$ and $L(a) \neq out$. L is a set of arguments labeling that consists of a set of arguments and labeling, $L = (arguments, labeling)$. There are different labels and the outcomes of the last example $L = \{(a,in),(b,out),(c,out),(d,in)\}$.

An argumentation approach is suitable to take an accurate decision in a complex situation. Multiple agents can argue to establish their ideas through an argumentation process. This approach provides optimal results in case of data conflicting or overlapping. That is why

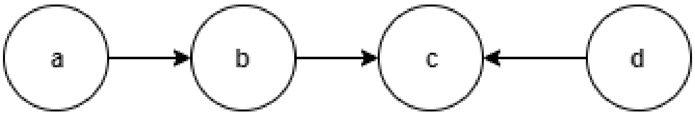


Figure 1: There are four nodes a, b, c, and d define four arguments and the edges presents the attacking relation between arguments.

it is fruitful to implement an argumentation approach for generating more reliable results. In the proposed system, an argumentation approach is implemented as a decision fusion approach instead of a voting classifier.

2.3.1 Probabilistic Argumentation Framework (PAF)

Probabilistic argumentation defines different formal frameworks regarding probabilistic logic. There are different approaches of the probabilistic argumentation frameworks [8]. In the probabilistic argumentation process, qualitative aspects can be handled by an underlying logic and quantitative aspects can be captured by probabilistic measures. The underlying logic can be different rules and the probabilistic measure can be the probability, weight, or other measurements. The process can be divided into the constellations approach and the epistemic approach [9] [26]. In the constellations approach, the probabilities of arguments and attacks are considered as probabilistic assessments. The probabilistic assessments evaluate the acceptability of arguments. In the epistemic approach, one believes more in an argument rather than an argument attacking it. This approach is modeling beliefs and agents. The agents are unable to change or directly add in the argumentation graph. The probabilities of arguments are denoting as the belief that an argument is acceptable. P is a probability function and A is an argument, $P(A) \geq 0.5$ denotes that the argument is believed, $P(A) \leq 0.5$ denotes that the argument is disbelieved, and $P(A) = 0.5$ denotes that the argument is nither believed or disbelieved [9]. The arguments in the graph that are believed to be acceptable to some degree (i.e. $P(A) \geq 0.5$). Figure 2 illustrates an argumentation process where A, B, C are three arguments. The attacking relation of arguments is defined according to the rules of the admissible arguments framework. $\{A,C\}$ is an admissible argument set. An argument A attacks an argument C because $\{A,C\}$ is conflict-free and they satisfy the rules of the admissible argumentation framework. Whereas, $\{A,B\}$ and $\{B,C\}$ are not conflict free, A directly attacks B and C directly attacks B.

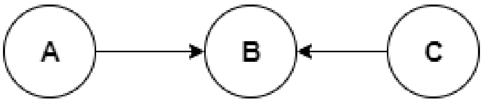


Figure 2: There are three nodes A, B, and C defines three arguments and the edges presents the attacking relation between arguments.

In the proposed study, two approaches of DTA namely decision tree classifiers (DTC) and decision tree regressors (DTR) work as two models component of ELA. The two different result sets from DTC and DTR act like two nodes and each node acts as an argument. A and B are the two different sets of arguments. In admissible arguments, A does not attack itself, but A attacks B and B attacks A. Figure 3 illustrates the admissible arguments, where $\{A, B\}$ is a set of arguments and they attack each other based on the probabilities of the arguments. In this scenario, *Attacks* is defined in a different way in the settings of ELA. The accuracy of an argument is calculated by different probability functions that are discussed in the Evaluation chapter. The accuracy works as the probability of an argument. The probability values of arguments define attacking relations. It decides which argument will attack whom. The argument with high probability attacks the argument with less probability.

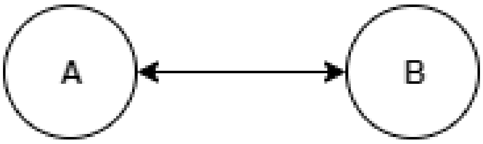


Figure 3: There are two nodes A and B defines two arguments and the edges presents the attacking relation between two arguments.

$$AF = (AR, Attacks) \tag{2.3}$$

According to the Figure 3, $AR = \{A,B\}$ and $Attacks = \{(A,B)\}or\{(B,A)\}$. The attacking relations depend on the probabilities of the arguments. An argument with high probability attacks the argument that holds less probability. For example $P(A) = 0.8$ and $P(B) = 0.46$ that means $P(A)$ attacks $P(B)$ and $P(B)$ can defend. In this example, $P(A) > 0.5$ which means $L(A) = in$. $L(A)$ is the labeling status of argument A. $L(A) = in$ means A is accepted. Let's discuss another example, where $P(A) = 0.48$ and $P(B) = 0.76$ that means $P(B)$ attacks $P(A)$ and $P(A)$ can defend. In this example, $P(B) > 0.5$ which means $L(B) = in$. $L(B)$ is the labeling status of argument B. $L(B) = in$ means B is accepted. On the other hand, $P(A) < 0.5$ and $L(A) = out$ that mean A is unaccepted.

If the probabilities of both arguments are above 0.5 then the argument with high probability will be accepted and the other one will be defeated. If the values of both arguments are similar then they will not attack each other (an extended rule that is implemented in the proposed system). The number of arguments nodes depends on the number of models of ELA. The outcomes from each model act as arguments. If the number of models is 3 then the number of arguments node will be 3 such as A, B, C. In case of more than two arguments, an argument with maximum probability attacks an argument that holds minimum probability and the accepted argument attacks an argument that holds the next maximum probability. The attacking process will be continued until the last argument in the process. For example $P(A) = 0.8$, $P(B) = 0.46$ and $P(C) = 0.66$. In this scenario, $P(A)$ attacks $P(B)$ and $P(B)$ can defend. $P(A) > 0.5$ which means $L(A) = in$. $L(A) = in$ means A is accepted. Then, $P(A)$ attacks $P(C)$ and $P(A) > P(C)$, $L(A) = in$ that means A is accepted. There are different labels such as in, out, and undec. The labels depend on the status (accepted, unaccepted, and undefined) of arguments. The proposed system implements modified rules of probabilistic argumentation framework in the settings of ELA for producing better outcomes.

A pipeline architecture is discussed in chapter 4 that implements the concepts of an ensemble learning architecture. In this architecture, decision tree classifiers and decision tree regressors are implemented as models and a probabilistic argumentation framework is implemented as a decision fusion technique.

3 Related Works

This chapter discusses previous researches that are related to ensemble architectures, learning algorithms, and argumentation approaches.

3.1 Implementation of Learning Algorithm

The learning algorithm is implemented for data classification. The algorithm classifies data based on different class labels for prediction. It predicts the outcomes based on classified data and training data. The research presented by [13] explains a robust technique for diagnosing brain diseases such as ischemic stroke, hemorrhage and hematoma and tumor from brain magnetic resonance (MR) and computerized tomography (CT) images. The system aims to help the physicians to detect different types of diseases for further treatments. It is implemented in a decision tree classifier to classify the data set. The data set is sent to the decision tree classifier and the classifier optimizes the data set for generating classification results. CT scan images of the brain are used for constructing training and testing data set. The classifier classifies data based on the training data set. The proposed system provides a better solution to the physicians to identify the actual diseases. The article provides a tangible idea to implement a decision tree algorithm for the proposed study.

The paper discusses three decision support systems and implemented machine learning approaches. [3]. The first decision support system helps an individual to select a hospital. The system makes the decision to suggest the appropriate hospital to the individual based on mortality, complication, and travel distance. Machine learning and optimization techniques help the system to make an appropriate decision. The second system assists an individual by suggesting the diagnostic test based on the types of diseases. The system can accelerate the diagnostic process, decrease the overuse of medical tests, save costs, and improve the accuracy of diagnosis. The third decision support system recommends the best lifestyle changes for an individual to minimize the risk of cardiovascular diseases (CVD). This system also uses machine learning and optimization techniques for recommendation. This research is an example of implementing different approaches of a decision tree algorithm for performing different tasks. In the proposed study, two approaches of a decision tree algorithm namely decision tree classifier (DTC) and decision tree regressor (DTR) are implemented.

3.2 Multi-Agents System

The research presented by [19] explains a multi-agents system where each agent remains in the network and represents a formal argumentation approach to take a consensus decision. A model-based architecture ensures reflective capabilities among the surrounding environments and the agents. The model is more successful for the agents to achieve their goals.

The model will be updated when the observations of an agent's surroundings indicate that the model is inaccurate. The agents can communicate with each other and share information among them. Moreover, the agents can alter their model to reflect the model aspect. The argumentation framework provides a consistent truth in case of conflicting and overlapping situations. In the multi-agents system (MAS), each agent has a Bayesian Network as a model and all of the agents unitedly construct a consistent network. The formal argumentation framework plays an important role to construct a consistent network. Several models of the agents argue through the formal argumentation framework and consequently, a consensus network is developed. Different models represent their joint model through the argumentation framework that works like a joint domain knowledge. The idea of working with multi-agents and argumentation framework is inspirational for the proposed research. In the proposed research, DTC and DTR work like multi-agents and the probabilistic argumentation framework (PAF) combines the outcomes from different agents and conducts an argumentation process.

3.3 Data Mining Techniques

The research introduced by [10] illustrates a decision tree-based approach for cardiovascular diagnosis. Data mining helps to find out the unknown patterns of the data exploring a large data set. The data set contains heart bit rate, blood pressure, mental stress, breathing rate and so on. There are three phases namely data pre-processing, data modeling, and data post-processing are applied for data processing and pattern recognition. The decision tree algorithm creates a model that learns simple decision rules from the data features and predicts outcomes. The ID3 algorithm is implemented to develop the system and the C4.5 algorithm ensures some improvements that enhance the performance of the ID3 algorithm. The algorithm smartly conducts some crucial tasks such as choosing appropriate attributes, handling training data with missing attribute values, managing attributes with different costs, pruning the decision tree, and operating continuous attributes. The algorithm constructs a decision tree from the training data set using the concept of entropy. It follows the divide and conquers approach for tree construction. This research is an excellent example of a decision tree implementation for cardiovascular diagnosis. This article provides a mature knowledge to use data mining techniques for data preparation. Furthermore, it explains the decision tree construction process for detecting a particular disease.

3.4 Argument Mining

Argument mining research presented by [4] explains the process of extracting arguments component and the predicting relations (attack and support) between the participant arguments. The research is implemented neural network models for performing an argument mining approach. It mentions several neural network architectures for predicting the attacking or supporting relations between the arguments based on different data sets. The neural network models use a binary classification approach to classify the data sets. A neural network determines parent and child relation between the arguments. Four types of features namely embeddings, textual entailment, sentiment features, and syntactic features are calculated to define the parent and child relation. The parent and child arguments pass through different layers and generate outputs. This research is an inspiration to work simultaneously with learning algorithms and an argumentation approach.

3.5 Summary

Several studies follow different approaches for solving classification problems. The previous studies implement different approaches such as decision tree algorithms, neural networks, arguments mining, and so on to classify data set for predicting outcomes. The data sets are pre-processed by different data mining techniques before transmitting to different approaches.

After all, the performance (e.g. accuracy and F-score) of a classification system is important and one should not neglect the fact that explainable outcomes might be of great importance for practical usage in e.g. medical decision support systems.

To the best of our knowledge, no previous study exists which uses an ensemble learning algorithm with an argumentation approach for analyzing classification problem. Although previous studies have analyzed similar assessment tasks by using classification algorithms or artificial intelligence (AI) approaches.

4 Theoretical Framework

This chapter introduces a pipeline architecture that integrates DTC, DTR, and PAF. It also introduces a formal algorithm of the proposed system.

4.1 The formal integration between decision tree and probabilistic argumentation

This study introduces a pipeline architecture of an ensemble learning approach. The architecture consists of different models of a learning algorithm (DTA) and a probabilistic argumentation framework. Several data mining techniques such as data fusion, data preprocessing, learning method, and decision fusion are important for pattern recognition of a data set extracted from ultrasound images. A decision tree algorithm (DTA) is implemented as a learning algorithm in the proposed system. Two different approaches of DTA namely decision tree classifier (DTC) and decision tree regressor (DTR) work as two different learning models. DTC and DTR classify the sample data set for prediction. Each model splits the sample data set into training and testing data set and predicts the output based on training data set. The models always update their knowledge base through a learning process for future prediction. They learn from various types of data sets and situations. The predicted outputs from the models will be performed as inputs of a probabilistic argumentation framework (PAF). A classical ensemble learning algorithm implements a voting classifier as a decision fusion approach. The proposed architecture is implemented PAF instead of

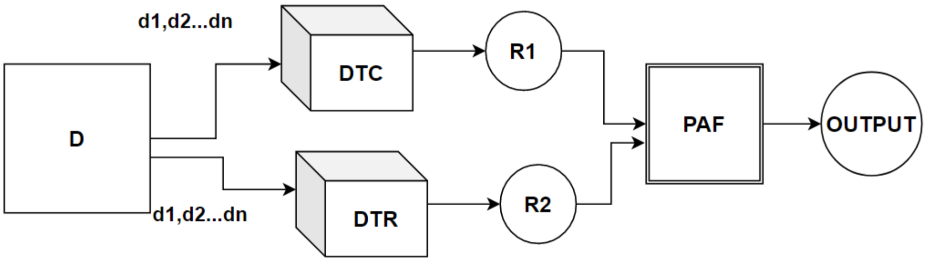


Figure 4: Represent the pipeline architecture of the proposed system where D is a data source that transmits different subsets of data, e.g., $d_1, d_2, ..., d_n$, to the models DTC and DTR. The size of the subsets should be less than or equal to the size of the data source. $R1$ and $R2$ contain the outputs generated by DTC and DTR. $R1$ and $R2$ send the outputs to the probabilistic argumentation framework (PAF) and PAF generates the final output.

a voting classifier. PAF generates outcomes through a decision-making process where the predicted outcomes from different models can argue. Furthermore, PAF helps to generate a consistent outcome in complex situations such as conflicting and overlapping information [19]. Figure 4 illustrates the pipeline architecture of the proposed system. According to Figure 4, D is a data source that contains a sample data set. The data source D divides the data set into subsets $d_1, d_2, ..., d_n$. D transmits different subsets of data, e.g., $d_1, d_2, ..., d_n$, to the models namely DTC and DTR. The size of a subset is less than or equal to the size of the source data set. DTC and DTR classify the given subsets of data and predict different outputs $R1$ and $R2$. The PAF conducts an argumentation process to produce an output. $R1$ and $R2$ are the participant arguments in the argumentation process and they attack each other. A probability function decides the attacking relation (which argument attacks to whom). If the values of both arguments are similar then they will not attack each other. The accepted argument will be produced as an output. The combined effort of the models and PAF generates a reliable output.

The prototype of the proposed system is designed in Python language. Anaconda and Jupyter notebook is the integrated development environment for implementation.

4.2 Algorithm and analysis

Algorithm 1 illustrates the implemented algorithm of the proposed system. According to Algorithm 1 it analyses a sample data set S, target variables T, and attributes att. S, T, att, and $estimator$ (number of models) are the parameter of the main function. S is a data set (e.g. test.csv) that contains numerical or categorical values. T and att are numpy array type of variables. $estimator$ is an integer variable. $left$ and $right$ are boolean variables. S should not be empty and att should be greater than 0. "createNode" function creates nodes for all models based on the sample data set and target variables. It creates a new node by checking the availability of the left and right branches of a node. The newly created nodes will be converted into root nodes if the condition will be satisfied. "classify" function classifies the data set for the created nodes according to the different class labels. "findBestSplit" function calculates the best splitting points by calculating the entropy of different attributes. It selects an attribute as the best splitting point that holds the lowest entropy. The entropy calculation method is described in the Background chapter. Each root node has two or more child nodes. A child node can be converted into a root node by satisfying the condition of a root node. It is a recursive process. The process will be continued until finding the leaf nodes. A leaf node holds a decision. MR is an array that holds the outcomes from different models. "prediction" function predicts the outcomes of different models based on training TR and testing TS data sets. The "argumentation" function generates outcomes using MR as a set of arguments and the probabilities (P) of arguments. An argument that contains maximum probability attacks an argument that holds minimum probability. If the values of both arguments are equal then they will not attack each other. The labeling status (L) of an argument depends on the probability function. If the probability of an argument is greater than 0.5 then the labeling status will be accepted. If the probability of an argument is less than 0.5 then the labeling status will be unaccepted. If the probability of an argument is equal to 0.5 then the labeling status will be undecided. "prediction" function sends the predicted outcome sets (MR) of different models to "argumentation" function as a parameter. The values of MR argue in "argumentation" function based on probabilities and the accepted value is considered as an output. Algorithm 2 represents the “argumentation” function. It is a simple probabilistic argumentation process. “argumentation” checks the probability of arguments. If there are more than two arguments in the process then an argument with maximum probability attacks the argument that holds the next maximum probability. The process will be continued. The experimental results of the algorithm are explained in the Evaluation chapter.

Algorithm 1 EnsembleArgumentation algorithm

```
procedure ENSEMBLEARGUMENTATION($(S, T, att, estimator)$)
    file $S$ = sampledataset
    array $T$ = targetvariable
    array $att$ = attributes
    integer $estimator$ = numberofmodel
    boolean $left$ = leftbranchofnode
    boolean $right$ = rightbranchofnode
    $s \neq 0$ & $att > 0$
    function createNode () do
    $node$=null
    if $left$ or $right$ then
        $node = root$
        return $node$
    end if
    return $node$
    end function
    function findBestSplit$(S, att)$ do
    $maxGain \leftarrow 0$
    $splitS \leftarrow$ null
    $e \leftarrow entropy.att$
    for all $a$ in $S$ do
        $gain \leftarrow$ infogain $(a, e)$
        if $(gain > maxGain)$ then
            $maxGain \leftarrow gain$
            $splitS \leftarrow a$
        end if
    end for
    end function
    for all $es$ in $estimator$ do
        if stopcondition$(S, T)! = 0$ then
            $leaf$ = createNode ()
            $leaf$.classLabel = classify ()
            return $leaf$
        end if
        $rootNode$ = createNode ()
        $rootNode$.condition = findBestSplit ( S, att )
        $N$ = $\{n|n$ outcome of $rootNode$.condition$\}$
        for all $n$ in $N$ do
            $S_v$ = $\{s|rootNode$.condition $s \in S\}$
            $child = S_v$
            $root \leftarrow child$
        end for
    end for
    array $MR$ = estimator.prediction$(TR, TS)$
    array $selectedarg$ = argumentation $(MR, P)$
    return $selectedarg$
```

Algorithm 2 Function: argumentation()

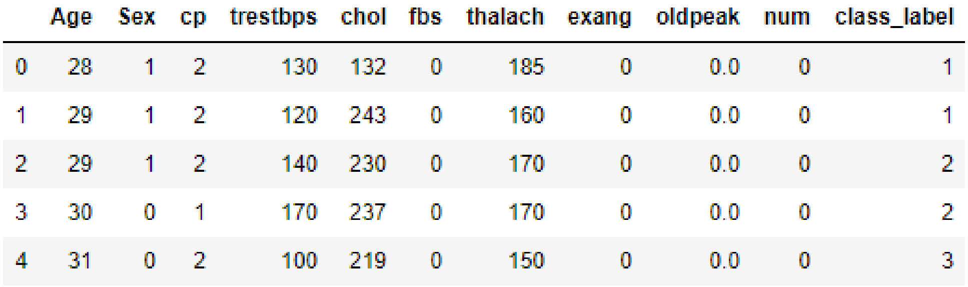
```
function $argumentation(MR, P)$ do
$a_{ij} \in MR$
$a_i = max$
$a_j = nextmax$
if $value(a_i) = value(a_j)$ then
    No attack
end if
if $value(a_i)! = value(a_j)$ and $P(a_i) = max$ then
    $max(P(ai))$ attacks $min(P(aj))$
end if
if $value(a_i)! = value(a_j)$ and $P(a_j) = max$ then
    $max(P(a_j))$ attacks $min(P(a_i))$
end if
if $P(a_{i,or,j}) > 0.5$ then
    $L(a_{i,or,j}) = in$
end if
if $P(a_{i,or,j}) < 0.5$ then
    $L(a_{i,or,j}) = out$
end if
if $P(a_{i,or,j}) = 0.5$ then
    $L(a_{i,or,j}) = undec$
end if
return $L(a_{i,or,j})$
end function
=0
```

5 Evaluation

The implementation of the argumentation-based ensemble learning algorithm and the experimentation with different data sets such as an extracted data set from ultrasound images and an open-access data set are discussed in this chapter.

5.1 An argumentation-based ensemble learning algorithm

ELA is defined for implementing the proposed system that can predict the result sets based on training data sets. Two kinds of data sets were used to evaluate the proposed system. One is an open-access data set and the other is an extracted data set from ultrasound images. Figure 5 illustrates the open-access data set. The data set contains 5500 rows of data and a total of 11 attributes (Age, Sex, cp, testbps, chol, fbs, thalach, exang, oldpeak, num, and classlabel). The "classlabel" is a dependant variable and the rest of the attributes are the independent variables. The dependant variable contains three class labels such as 1, 2, and 3 which are equivalent to C1, C2, and C3. The values of the dependant variable depend on the values of the independent variables.

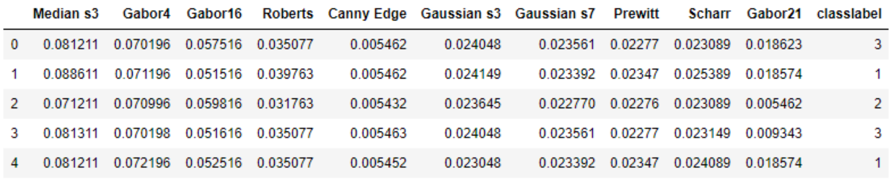


	Age	Sex	cp	trestbps	chol	fbs	thalach	exang	oldpeak	num	class_label
0	28	1	2	130	132	0	185	0	0.0	0	1
1	29	1	2	120	243	0	160	0	0.0	0	1
2	29	1	2	140	230	0	170	0	0.0	0	2
3	30	0	1	170	237	0	170	0	0.0	0	2
4	31	0	2	100	219	0	150	0	0.0	0	3

Figure 5: An open-access data set with a total of 11 attributes where classlabel is a dependant variable and the rest of the attributes are the independent variable. The dependant variable holds three class labels such as 1, 2, and 3.

Figure 6 illustrates the extracted data set from ultrasound images. The ultrasound images contain information and it is important to extract the information from ultrasound images to construct a data set. The data set contains 4900 rows of data and a total of eleven attributes. The classlabel attribute is the dependant variable that contains three values 1, 2, 3 that are equivalent to C1, C2, C3 and the rest of the attributes are the independent variables.

The image segmentation approach helps to extract important information as a data set from ultrasound images. Several image filtering techniques namely Gabor, Robert, Sobel, Prewitt, Gaussian s3, Gaussian s7, Scharr, and Median s3 applied to filter the ultrasound images for constructing the data set (Figure 6). The implemented filter techniques are used in image processing for feature extraction, stereo disparity estimation, texture analysis, and analysis



	Median s3	Gabor4	Gabor16	Roberts	Canny Edge	Gaussian s3	Gaussian s7	Prewitt	Scharr	Gabor21	classlabel
0	0.081211	0.070196	0.057516	0.035077	0.005462	0.024048	0.023561	0.02277	0.023089	0.018623	3
1	0.088611	0.071196	0.051516	0.039763	0.005462	0.024149	0.023392	0.02347	0.025389	0.018574	1
2	0.071211	0.070996	0.059816	0.031763	0.005432	0.023645	0.022770	0.02276	0.023089	0.005462	2
3	0.081311	0.070198	0.051616	0.035077	0.005463	0.024048	0.023561	0.02277	0.023149	0.009343	3
4	0.081211	0.072196	0.052516	0.035077	0.005452	0.023048	0.023392	0.02347	0.024089	0.018574	1

Figure 6: An extracted data set from ultrasound images with a total of 11 attributes where classlabel is a dependant variable and the rest of the attributes are the independent variable. The dependant variable holds three values such as 1, 2, and 3.

of complex oscillation. The filtering techniques extract features from ultrasound images. The values of theta, sigma, lambda, and gamma are calculated in the implementation of the Gabor filtering technique. The Robert, Sobel, Prewitt, Gaussian, Scharr, and Median are the building image filtering techniques of the scikit-image (skimage). The system can work both the open-access data set and ultrasound images. The decision tree algorithm (DTA) is used as a base learning algorithm in the ensemble learning architecture. DTA classifies the data set (sample data set or extracted data set from images) for prediction. There are three different groups of ultrasound images such as C1, C2, and C3. Figure 7 illustrates the ultrasound images based on different groups. Two approaches of DTA namely decision tree classifier (DTC) and decision tree regressor (DTR) act like two models. The models split the data set into two different data sets, one is the training data set and the other is the testing data set.

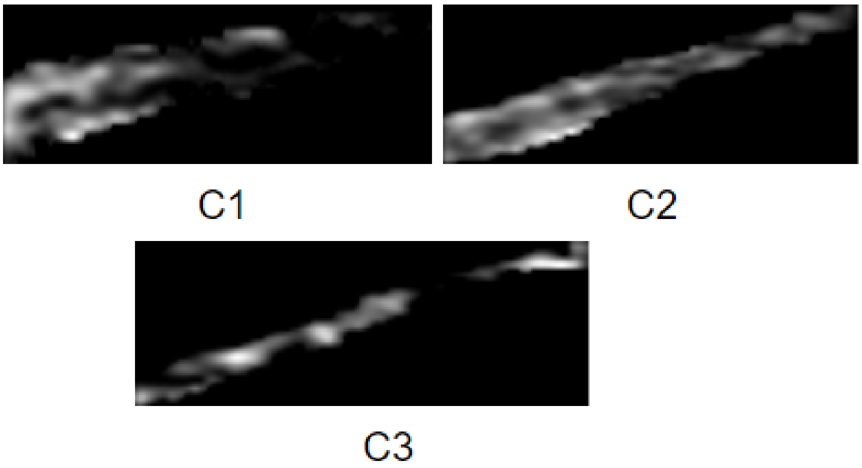


Figure 7: Ultrasound images have three different class such as C1, C2 and C3. Each class represents different information. For instance, C1, C2 and C3 represent the heath conditions of a patient such as good, bad and serious successively.

The 20% of randomly selected data from the sample data set constructs the testing data set and the rest of them construct the training data set. Entropy helps to locate the splitting point of a data set. The algorithm analyzes data from the training data set and produces outputs based on analyzed data. It learns from various situations and updates the training data set. DTC and DTR fit the training and testing data set for prediction. The models predict class labels based on the training data set. The testing data set is used as input. whereas, the training data set works as the knowledge base. Both models namely DTC and DTR take testing data set as input and produce different result sets analyzing the input data set. The testing data set (input data set) is similar for both models. The knowledge base (training data set) helps to analyze the input data set for predicting a result set. The result set contains several class labels regarded as values. The values of two result sets can be similar or not and each value has its own accuracy. DTC and DTR send the result sets to the Probabilistic argumentation framework (PAF) for producing an output. The predicted results from both models argue in PAF. Each result act as an argument. Arguments of both models can attack each other. The accuracy function calculates the accuracy of each argument that also works as a probability in PAF. The attacking relation of arguments depends on the probabilities of arguments. An argument holding high probability attacks an argument that contains low probability. One argument is accepted through the argumentation process. On the other hand, the other is rejected. The accepted argument is regarded as a final output.

5.2 Evaluation of the theoretical framework

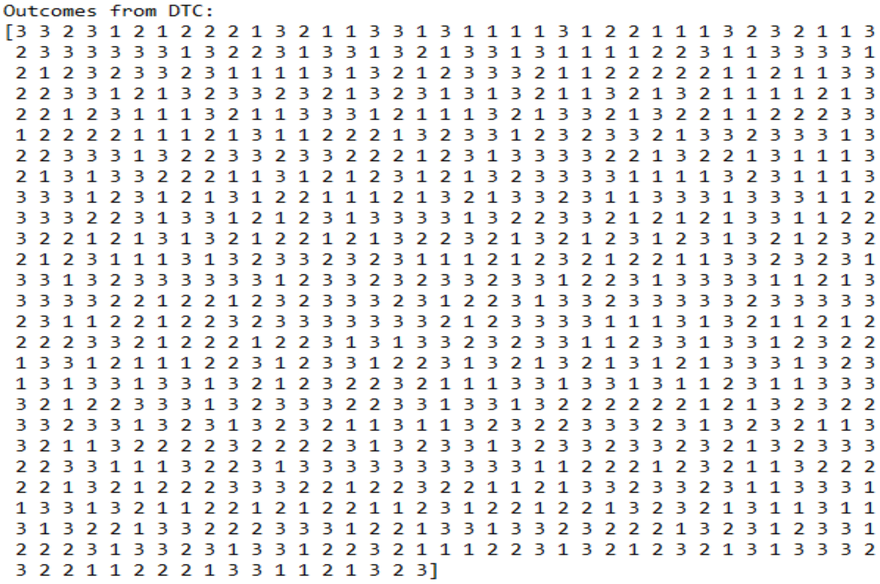
The results sets of the experiment are varied with different testing data sets. In this scenario, the system takes 20% of testing data sets as input and predicts outcomes. The outcomes from different models and PAF are explained below.

Figure 10 illustrates a predicted outcome set and graphical representation of DTC model. The outcome set (Fig.8) contains a total of 980 values and three class labels such as 1, 2, and 3. Outcomes from the outcome set of DTC act as arguments in PAF. For instance, the index positions 0, 1, 2, 3 of the outcome set contain 3, 3, 2, 3 as the outcomes, and they argue with the outcomes at index positions 0, 1, 2, 3 in the outcome set of DTR. The graphical illustration (Fig.9) depicts the graphical view of the outcome set generated from DTC. The X-axis illustrates the index positions of existing outcomes. The Y-axis illustrates the class labels (outcomes 1, 2, 3). The graph represents 980 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 3, 3, 2, 3, 1, 2, 1, 2, 2 respectively.

Figure 13 illustrates a predicted outcome set and graphical representation of DTR model. The outcome set (Fig.11) contains a total of 980 values and three class labels such as 1, 2, and 3. Outcomes from the outcome set of DTR also act as arguments in PAF. For instance, the index positions 0, 1, 2, 3 of the outcome set contain 3, 2, 3, 2 as the outcomes, and they argue with the outcomes at index positions 0, 1, 2, 3 in the outcome set of DTC. The graphical illustration (Fig.12) depicts the graphical view of the outcome set generated from DTR. The X-axis illustrates the index positions of existing outcomes. The Y-axis illustrates the class labels (outcomes 1, 2, 3). The graph represents 980 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 3, 2, 3, 2, 3, 3, 1, 1, 1 respectively.

Figure 16 illustrates the similar index positions and outcomes. Figure 14 illustrates the index positions of the outcome sets of DTC and DTR where they produce similar outcomes. Figure 15 represents similar outcomes according to similar index positions (Fig.14). Both outcome sets contain 346 similar outcomes according to similar index positions. Furthermore, the outcomes will not attack each other. For example, at index positions 0, 6, 9, 10, 12, 14, 15, 19, 20 the outcome sets of DTC and DTR hold the outcomes 3, 1, 2, 1, 2, 1, 2, 1, 1 successively.

Figure 17 illustrates the graphical presentation of both result sets generated from DTC and DTR. The light blue color indicates the outcomes of DTR whereas the red color indicates



```
Outcomes from DTC:
[3 3 2 3 1 2 1 2 2 2 1 3 2 1 1 3 3 1 3 1 1 1 1 3 1 2 2 1 1 1 3 2 3 2 1 1 3
 2 3 3 3 3 3 3 1 3 2 2 3 1 3 3 1 3 2 1 3 3 1 3 1 1 1 1 2 2 3 1 1 3 3 3 3 1
 2 1 2 3 2 3 3 2 3 1 1 1 1 3 1 3 2 1 2 3 3 3 2 1 1 2 2 2 2 2 1 1 2 1 1 3 3
 2 2 3 3 1 2 1 3 2 3 3 2 3 2 1 3 2 3 1 3 1 3 2 1 1 3 2 1 3 2 1 1 1 1 2 1 3
 2 2 1 2 3 1 1 1 3 2 1 1 3 3 3 1 2 1 1 1 3 2 1 3 3 2 1 3 2 2 1 1 2 2 2 3 3
 1 2 2 2 2 1 1 1 2 1 3 1 1 2 2 2 1 3 2 3 3 1 2 3 2 3 3 2 1 3 3 2 3 3 3 1 3
 2 2 3 3 3 1 3 2 2 3 3 2 3 3 2 2 2 1 2 3 1 3 3 3 3 2 2 1 3 2 2 1 3 1 1 1 3
 2 1 3 1 3 3 2 2 2 1 1 3 1 2 1 2 3 1 2 1 3 2 3 3 3 3 1 1 1 1 3 2 3 1 1 1 3
 3 3 3 1 2 3 1 2 1 3 1 2 2 1 1 1 2 1 3 2 1 3 3 2 3 1 1 3 3 3 1 3 3 3 1 1 2
 3 3 3 2 2 3 1 3 3 1 2 1 2 3 1 3 3 3 3 1 3 2 2 3 3 2 1 2 1 2 1 3 3 1 1 2 2
 3 2 2 1 2 1 3 1 3 2 1 2 2 1 2 1 3 2 2 3 2 1 3 2 1 2 3 1 2 3 1 3 2 1 2 3 2
 2 1 2 3 1 1 1 3 1 3 2 3 3 2 3 2 3 1 1 1 2 1 2 3 2 1 2 2 1 1 3 3 2 3 2 3 1
 3 3 1 3 2 3 3 3 3 3 3 1 2 3 3 2 3 2 3 3 2 3 3 1 2 2 3 1 3 3 3 3 1 1 2 1 3
 3 3 3 3 2 2 1 2 2 1 2 3 2 3 3 3 2 3 1 2 2 3 1 3 3 2 3 3 3 3 3 2 3 3 3 3 3
 2 3 1 1 2 2 1 2 2 3 2 3 3 3 3 3 3 3 2 1 2 3 3 3 3 1 1 1 3 1 3 2 1 1 2 1 2
 2 2 2 3 3 2 1 2 2 2 1 2 2 3 1 3 1 3 3 2 3 2 3 3 1 1 2 3 3 1 3 3 1 2 3 2 2
 1 3 3 1 2 1 1 1 2 2 3 1 2 3 3 1 2 2 3 1 3 2 1 3 2 1 3 1 2 1 3 3 3 1 3 2 3
 1 3 1 3 3 1 3 3 1 3 2 1 2 3 2 2 3 2 1 1 1 3 3 1 3 3 1 3 1 1 2 3 1 1 3 3 3
 3 2 1 2 2 3 3 3 1 3 2 3 3 3 2 2 3 3 1 3 3 1 3 2 2 2 2 2 2 1 2 1 3 2 3 2 2
 3 3 2 3 3 1 3 2 3 1 3 2 3 2 1 1 3 1 1 3 2 3 2 2 3 3 3 2 3 1 3 2 3 2 1 1 3
 3 2 1 1 3 2 2 2 2 3 2 2 2 2 3 1 3 2 3 3 1 3 2 3 3 2 3 3 2 3 2 1 3 2 3 3 3
 2 2 3 3 1 1 1 3 2 2 3 1 3 3 3 3 3 3 3 3 3 3 1 1 2 2 2 1 2 3 2 1 1 3 2 2 2
 2 2 1 3 2 1 2 2 2 3 3 3 2 2 1 2 2 3 2 2 1 1 2 1 3 3 2 3 3 2 3 1 1 3 3 3 1
 1 3 3 1 3 2 1 1 2 2 1 2 1 2 2 1 1 2 3 1 2 2 1 2 2 1 3 2 3 2 1 3 1 1 3 1 1
 3 1 3 2 2 1 3 3 2 2 2 3 3 3 1 2 2 1 3 3 1 3 3 2 3 2 2 2 1 3 2 3 1 2 3 3 1
 2 2 2 3 1 3 3 2 3 1 3 3 1 2 2 3 2 1 1 1 2 2 3 1 3 2 1 2 3 2 1 3 1 3 3 3 2
 3 2 2 1 1 2 2 2 1 3 3 1 1 2 1 3 2 3]
```

Figure 8: Represent the generated outcome set from DTC that contains 980 values. The outcome set contains three class labels such as 1, 2, 3.

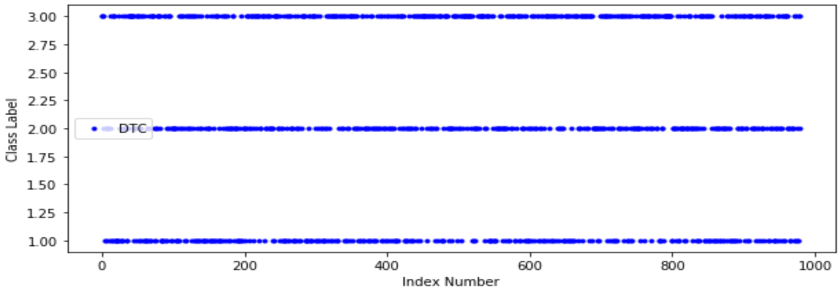
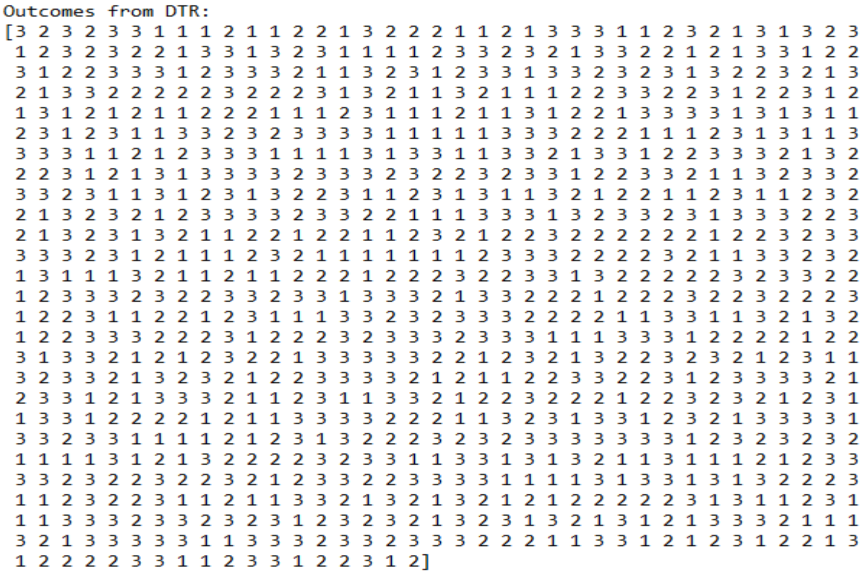


Figure 9: Represent the graphical representation of the generated output set from DTC. X-axis and Y-axis illustrate the index number and the class label of the available outcome successively.

Figure 10: Represent the outcome set of DTC and graphical representation of the outcome set.

the outcomes of DTC. The X-axis and Y-axis depict the index number of existing values in the result sets and the class labels respectively. The two graphs locate some points in the same locations that imply the values of the identical location are similar. Both outcome sets contain the same class labels in the identical location and they will not attack each other. There are 346 similar values that exist in similar positions of both outcome sets. For example, index positions 0, 6, 9, 10, 12, 14, 15, 19, 20 present the class labels 3, 1, 2, 1, 2, 1, 2, 1, 1 respectively for both of the models (DTC and DTR).

Figure 18 illustrates the accepted outcome set of PAF. The outcome set contains a total of 639 values and three class labels such as 1, 2, 3. The outcome set of PAF is different from the outcome from DTC and DTR. The outcomes from two different outcome sets (DTC and DTR) participate in an argument process through PAF. They argue among themselves for finalizing outcomes (accepted outcomes). PAF produces an outcome set that contains only accepted outcomes. For example, The outcome sets of DTC and DTR hold the outcomes 3



```
Outcomes from DTR:
[3 2 3 2 3 3 1 1 1 2 1 1 2 2 1 3 2 2 2 1 1 2 1 3 3 3 1 1 2 3 2 1 3 1 3 2 3
 1 2 3 2 3 2 2 1 3 3 1 3 2 3 1 1 1 1 2 3 3 2 3 2 1 3 3 2 2 1 2 1 3 3 1 2 2
 3 1 2 2 3 3 3 1 2 3 3 3 2 1 1 3 2 3 1 2 3 3 1 3 3 2 3 2 3 1 3 2 2 3 2 1 3
 2 1 3 3 2 2 2 2 2 3 2 2 2 3 1 3 2 1 1 3 2 1 1 1 2 2 3 3 2 2 3 1 2 2 3 1 2
 1 3 1 2 1 2 1 1 2 2 2 1 1 1 2 3 1 1 1 2 1 1 3 1 2 2 1 3 3 3 3 1 3 1 3 1 1
 2 3 1 2 3 1 1 3 3 2 3 2 3 3 3 3 1 1 1 1 1 3 3 3 2 2 2 1 1 1 2 3 1 3 1 1 3
 3 3 3 1 1 2 1 2 3 3 3 1 1 1 1 3 1 3 3 1 1 3 3 2 1 3 3 1 2 2 3 3 3 2 1 3 2
 2 2 3 1 2 1 3 1 3 3 3 3 2 3 3 3 2 3 2 2 3 2 3 3 1 2 2 3 3 2 1 1 3 2 3 3 2
 3 3 2 3 1 1 3 1 2 3 1 3 2 2 3 1 1 2 3 1 3 1 1 3 2 1 2 2 1 1 2 3 1 1 2 3 2
 2 1 3 2 3 2 1 2 3 3 3 3 2 3 3 2 2 1 1 1 3 3 3 1 3 2 3 3 2 3 1 3 3 3 2 2 3
 2 1 3 2 3 1 3 2 1 1 2 2 1 2 2 1 1 2 3 2 1 2 2 3 2 2 2 2 2 2 1 2 2 3 2 3 3
 3 3 3 2 3 1 2 1 1 1 2 3 2 1 1 1 1 1 1 1 2 3 3 3 2 2 2 2 3 2 1 1 3 3 2 3 2
 1 3 1 1 1 3 2 1 1 2 1 1 2 2 2 1 2 2 2 3 2 2 3 3 1 3 2 2 2 2 2 3 2 3 3 2 2
 1 2 3 3 3 2 3 2 2 3 3 2 3 3 1 3 3 3 2 1 3 3 2 2 2 1 2 2 2 3 2 2 3 2 2 2 3
 1 2 2 3 1 1 2 2 1 2 3 1 1 1 3 3 2 3 2 3 3 3 2 2 2 2 1 1 3 3 1 1 3 2 1 3 2
 1 2 2 3 3 3 2 2 2 3 1 2 2 2 3 2 3 3 3 2 3 3 3 1 1 1 3 3 3 1 2 2 2 2 1 2 2
 3 1 3 3 2 1 2 1 2 3 2 2 1 3 3 3 3 3 2 2 1 2 3 2 1 3 2 2 3 2 3 2 1 2 3 1 1
 3 2 3 3 2 1 3 2 3 2 1 2 2 3 3 3 3 2 1 2 1 1 2 2 3 3 2 2 3 1 2 3 3 3 3 2 1
 2 3 3 1 2 1 3 3 3 2 1 1 2 3 1 1 3 3 2 1 2 2 3 2 2 2 1 2 2 3 2 3 2 1 2 3 1
 1 3 3 1 2 2 2 2 1 2 1 1 3 3 3 3 2 2 2 1 1 3 2 3 1 3 3 1 2 3 2 1 3 3 3 3 1
 3 3 2 3 3 1 1 1 1 2 1 2 3 1 3 2 2 2 3 2 3 2 3 3 3 3 3 3 3 1 2 3 2 3 2 3 2
 1 1 1 1 3 1 2 1 3 2 2 2 2 3 2 3 3 1 1 3 3 1 3 1 3 2 1 1 3 1 1 1 2 1 2 3 3
 3 3 2 3 2 2 3 2 2 3 2 1 2 3 3 2 2 3 3 3 3 1 1 1 1 3 1 3 3 1 3 1 3 2 2 2 3
 1 1 2 3 2 2 3 1 1 2 1 1 3 3 2 1 3 2 1 3 2 1 2 1 2 2 2 2 2 3 1 3 1 1 2 3 1
 1 1 3 3 3 2 3 3 2 3 2 3 1 2 3 2 3 2 1 3 2 3 1 3 2 1 3 1 2 1 3 3 3 2 1 1 1
 3 2 1 3 3 3 3 3 1 1 3 3 3 2 3 3 2 3 3 3 2 2 2 1 1 3 3 1 2 1 2 3 1 2 2 1 3
 1 2 2 2 2 3 3 1 1 2 3 3 1 2 2 3 1 2]
```

Figure 11: Represent the generated outcome set from DTR that contains 980 values. The outcome set contains three class labels such as 1, 2, 3.

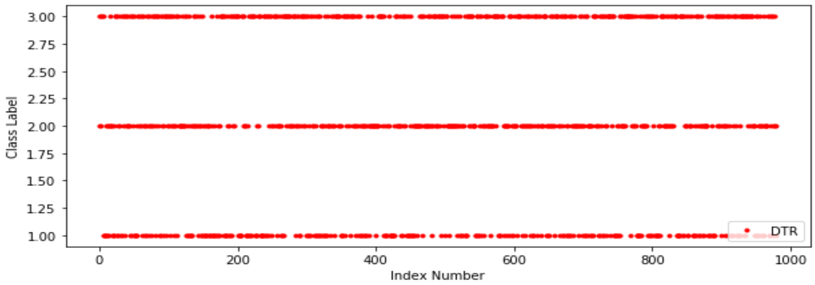
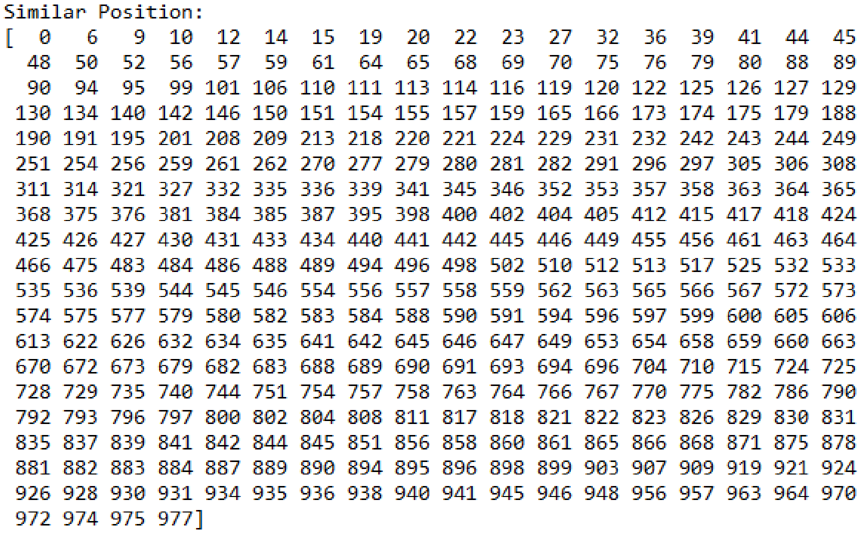


Figure 12: Represent the graphical representation of the generated output set from DTR. X-axis and Y-axis illustrate the index number and the class label of the available outcome successively.

Figure 13: Represent the outcome set of DTR and graphical representation of the outcome set.

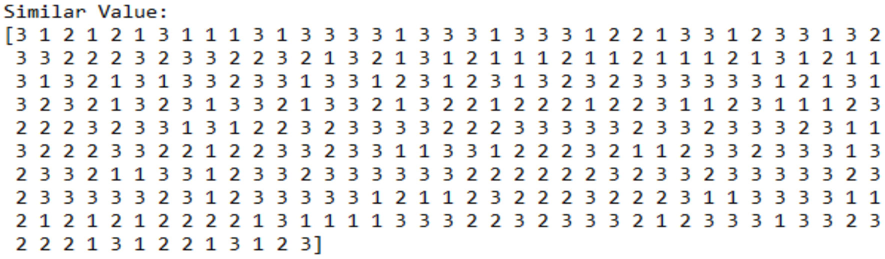
and 2 respectively at index position 1. The outcomes 3 and 2 argue among themselves based on probability. Finally, 3 is considered as an acceptable outcome.The accepted outcome 3 locates the index position 0 in the outcome set of PAF. Figure 19 depicts the graphical view of the accepted outcome set generated from PAF. The X-axis illustrates the index positions. The Y-axis illustrates the class labels. The outcome set has three different class labels such as 1, 2, 3. The graph represents 639 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 3, 2, 3, 1, 2, 2, 2, 3, 1 respectively.

Figure 20 illustrates the attacking relations of the outcomes of different models. It represents only 12 values from the experimented outcome sets (DTC, DTR, PAF) to illustrate the attacking relations based on different probabilities and conditions. The outcome sets contain class labels such as, 1, 2, and 3. The outcomes of both models act as arguments and they attack based on probability. An outcome with a high probability attacks the outcome that contains less probability. PAF decides which argument is accepted. If the same index



```
Similar Position:
[  0   6   9  10  12  14  15  19  20  22  23  27  32  36  39  41  44  45
  48  50  52  56  57  59  61  64  65  68  69  70  75  76  79  80  88  89
  90  94  95  99 101 106 110 111 113 114 116 119 120 122 125 126 127 129
 130 134 140 142 146 150 151 154 155 157 159 165 166 173 174 175 179 188
 190 191 195 201 208 209 213 218 220 221 224 229 231 232 242 243 244 249
 251 254 256 259 261 262 270 277 279 280 281 282 291 296 297 305 306 308
 311 314 321 327 332 335 336 339 341 345 346 352 353 357 358 363 364 365
 368 375 376 381 384 385 387 395 398 400 402 404 405 412 415 417 418 424
 425 426 427 430 431 433 434 440 441 442 445 446 449 455 456 461 463 464
 466 475 483 484 486 488 489 494 496 498 502 510 512 513 517 525 532 533
 535 536 539 544 545 546 554 556 557 558 559 562 563 565 566 567 572 573
 574 575 577 579 580 582 583 584 588 590 591 594 596 597 599 600 605 606
 613 622 626 632 634 635 641 642 645 646 647 649 653 654 658 659 660 663
 670 672 673 679 682 683 688 689 690 691 693 694 696 704 710 715 724 725
 728 729 735 740 744 751 754 757 758 763 764 766 767 770 775 782 786 790
 792 793 796 797 800 802 804 808 811 817 818 821 822 823 826 829 830 831
 835 837 839 841 842 844 845 851 856 858 860 861 865 866 868 871 875 878
 881 882 883 884 887 889 890 894 895 896 898 899 903 907 909 919 921 924
 926 928 930 931 934 935 936 938 940 941 945 946 948 956 957 963 964 970
 972 974 975 977]
```

Figure 14: Represent the index positions of similar outcomes from both outcome sets.



```
Similar Value:
[3 1 2 1 2 1 3 1 1 1 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1 2 2 1 3 3 1 2 3 3 1 3 2
 3 3 2 2 2 3 2 3 3 2 2 3 2 1 3 2 1 3 1 2 1 1 1 2 1 1 2 1 1 1 2 1 3 1 2 1 1
 3 1 3 2 1 3 1 3 3 2 3 3 1 3 3 1 2 3 1 2 3 1 3 2 3 2 3 3 3 3 3 3 1 2 1 3 1
 3 2 3 2 1 3 2 3 1 3 3 2 1 3 3 2 1 3 2 2 1 2 2 2 1 2 2 3 1 1 2 3 1 1 1 2 3
 2 2 2 3 2 3 3 1 3 1 2 2 3 2 3 3 3 3 2 2 2 3 3 3 3 3 2 3 3 2 3 3 3 2 3 1 1
 3 2 2 2 3 3 2 2 1 2 2 3 3 2 3 3 1 1 3 3 1 2 2 2 3 2 1 1 2 3 3 2 3 3 3 1 3
 2 3 3 2 1 1 3 3 1 2 3 3 2 3 3 3 3 3 3 2 2 2 2 2 2 3 2 3 3 2 3 3 3 3 3 2 3
 2 3 3 3 3 3 2 3 1 2 3 3 3 3 3 1 2 1 1 2 3 2 2 2 3 2 2 2 3 1 1 3 3 3 3 1 1
 2 1 2 1 2 1 2 2 2 2 1 3 1 1 1 1 3 3 3 2 2 3 2 3 3 3 2 1 2 3 3 3 1 3 3 2 3
 2 2 2 1 3 1 2 2 1 3 1 2 3]
```

Figure 15: Represent similar outcomes based on the similar index position from both outcome sets.

Figure 16: Represent the sets of similar index positions and similar outcomes from both models (DTC and DTR).

position of both outcome sets holds similar values then they will not attack. For example, the index positions 0, 6, 9, 10 of both outcome sets hold 3, 1, 2, 1 as the similar outcomes respectively and they do not attack. At index position 1, the outcomes sets of DTC and DTR hold 3, 2 as the outcomes. The probability of outcome 3 is 0.72, whereas, it is 0.63 for outcome 2. Outcome 3 attacks outcome 2. Finally, outcome 3 is accepted through an argumentation process. The outcome sets of DTC and DTR hold the outcomes 3 and 2 respectively at index position 3. The probability of outcome 2 is 0.88, whereas, it is 0.67 for outcome 3. The outcome 2 attacks outcome 3 and 2 is regarded as the accepted outcome. In similarly, The outcome sets of DTC and DTR hold the outcomes 2 and 1 respectively at index position 7. The probability of outcome 1 is 0.94, whereas, it is 0.92 for outcome 2. The outcome 1 attacks outcome 2 and 1 is regarded as the accepted outcome. Similarly, the rest of the outcomes of DTC and DTR attack each other based on probability and PAF produces the outcome set of accepted outcomes.

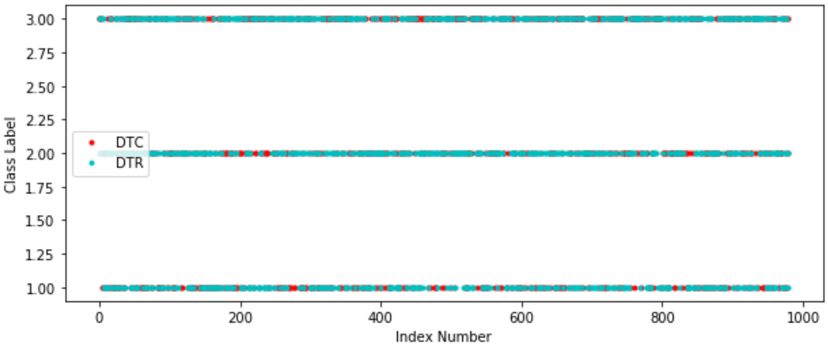
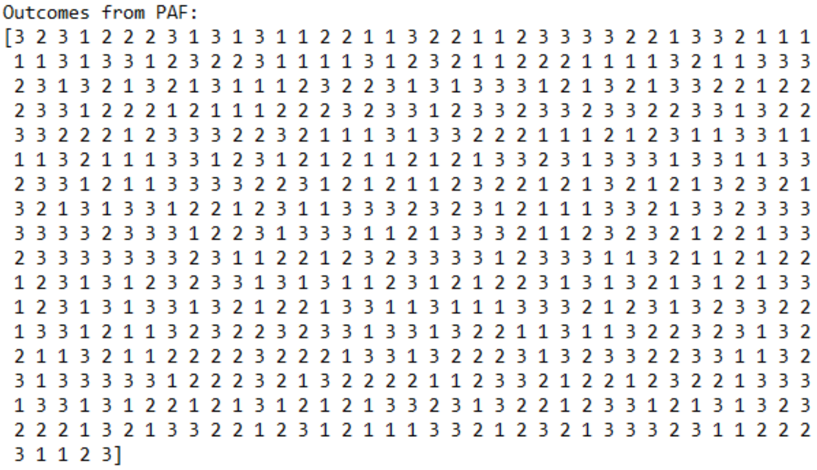


Figure 17: Represent the generated output sets from DTC and DTR. Y-axis and X-axis illustrate the different class labels and the index position of the values successively. Both of the result sets have some similar values in similar index positions.



```
Outcomes from PAF:
[3 2 3 1 2 2 2 3 1 3 1 3 1 1 2 2 1 1 3 2 2 1 1 2 3 3 3 3 2 2 1 3 3 2 1 1 1
 1 1 3 1 3 3 1 2 3 2 2 3 1 1 1 1 3 1 2 3 2 1 1 2 2 2 1 1 1 1 3 2 1 1 3 3 3
 2 3 1 3 2 1 3 2 1 3 1 1 1 2 3 2 2 3 1 3 1 3 3 3 1 2 1 3 2 1 3 3 2 2 1 2 2
 2 3 3 1 2 2 2 1 2 1 1 1 2 2 2 3 2 3 3 1 2 3 3 2 3 3 2 3 3 2 2 3 3 1 3 2 2
 3 3 2 2 2 1 2 3 3 3 2 2 3 2 1 1 1 3 1 3 3 2 2 2 1 1 1 2 1 2 3 1 1 3 3 1 1
 1 1 3 2 1 1 1 3 3 1 2 3 1 2 1 2 1 1 2 1 2 1 3 3 2 3 1 3 3 3 1 3 3 1 1 3 3
 2 3 3 1 2 1 1 3 3 3 3 2 2 3 1 2 1 2 1 1 2 3 2 2 1 2 1 3 2 1 2 1 3 2 3 2 1
 3 2 1 3 1 3 3 1 2 2 1 2 3 1 1 3 3 3 2 3 2 3 1 2 1 1 1 3 3 2 1 3 3 2 3 3 3
 3 3 3 3 2 3 3 3 1 2 2 3 1 3 3 3 1 1 2 1 3 3 3 2 1 1 2 3 2 3 2 1 2 2 1 3 3
 2 3 3 3 3 3 3 3 2 3 1 1 2 2 1 2 3 2 3 3 3 3 1 2 3 3 3 1 1 3 2 1 1 2 1 2 2
 1 2 3 1 3 1 2 3 2 3 3 1 3 1 3 1 1 2 3 1 2 1 2 2 3 1 3 1 3 2 1 3 1 2 1 3 3
 1 2 3 1 3 1 3 3 1 3 2 1 2 2 1 3 3 1 1 3 1 1 1 3 3 3 2 1 2 3 1 3 2 3 3 2 2
 1 3 3 1 2 1 1 3 2 3 2 2 3 2 3 3 1 3 3 1 3 2 2 1 1 3 1 1 3 2 2 3 2 3 1 3 2
 2 1 1 3 2 1 1 2 2 2 2 3 2 2 2 1 3 3 1 3 2 2 2 3 1 3 2 3 3 2 2 3 3 1 1 3 2
 3 1 3 3 3 3 3 1 2 2 2 3 2 1 3 2 2 2 2 1 1 2 3 3 2 1 2 2 1 2 3 2 2 1 3 3 3
 1 3 3 1 3 1 2 2 1 2 1 3 1 2 1 2 1 3 3 2 3 1 3 2 2 1 2 3 3 1 2 1 3 1 3 2 3
 2 2 2 1 3 2 1 3 3 2 2 1 2 3 1 2 1 1 1 3 3 2 1 2 3 2 1 3 3 3 2 3 1 1 2 2 2
 3 1 1 2 3]
```

Figure 18: Represent the accepted outcome set of PAF that contains 639 values. The outcome set contains three class labels such as 1, 2, 3.

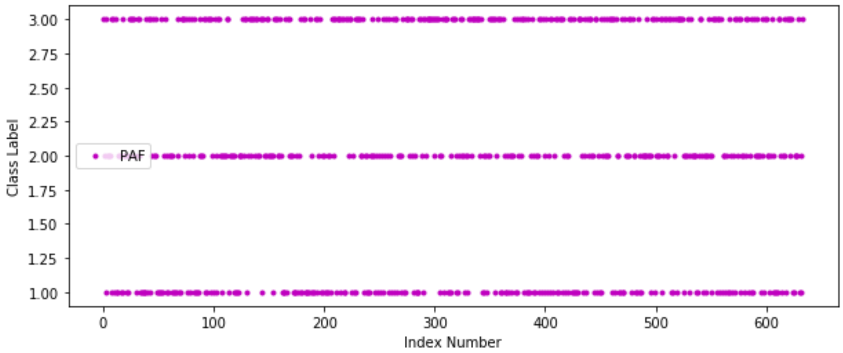


Figure 19: Represent the generated accepted output set from PAF. Y-axis and X-axis illustrate the different class labels and the index number respectively.

5.2.1 Probability function and accuracy

This section discusses two probability equations that are implemented to calculate the probability of an argument.

$$Probability1 : (y, \bar{y}) = (n \div s) \sum_{i=0}^{(s-n)} 1(\bar{y} = y_i) \tag{5.1}$$

n = number of match element

s = number of sample element (a total number of element)

i = index number of element

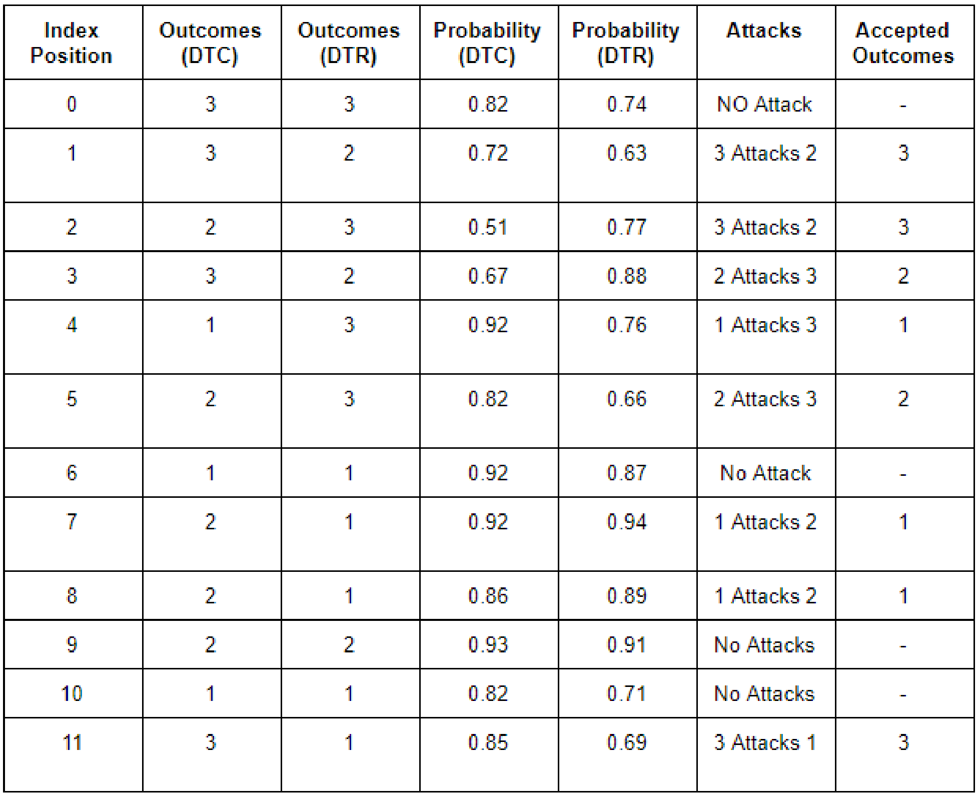
Equation 5.1 illustrates a probability equation where n, s, i define number of match element, number of sample element, and index number of element successively. The probability of result sets from different models is calculated by the probability equation.

$$Probability2 : p = (TP + TN)/(TP + FP + FN + TN) \tag{5.2}$$

TP = true-positive

TN = true-negative

FP = false-positive



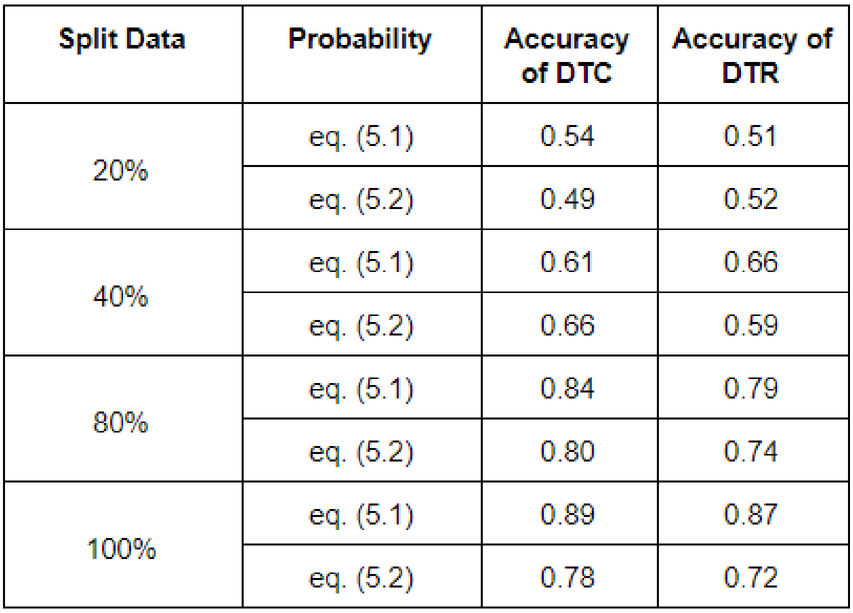
Index Position	Outcomes (DTC)	Outcomes (DTR)	Probability (DTC)	Probability (DTR)	Attacks	Accepted Outcomes
0	3	3	0.82	0.74	NO Attack	-
1	3	2	0.72	0.63	3 Attacks 2	3
2	2	3	0.51	0.77	3 Attacks 2	3
3	3	2	0.67	0.88	2 Attacks 3	2
4	1	3	0.92	0.76	1 Attacks 3	1
5	2	3	0.82	0.66	2 Attacks 3	2
6	1	1	0.92	0.87	No Attack	-
7	2	1	0.92	0.94	1 Attacks 2	1
8	2	1	0.86	0.89	1 Attacks 2	1
9	2	2	0.93	0.91	No Attacks	-
10	1	1	0.82	0.71	No Attacks	-
11	3	1	0.85	0.69	3 Attacks 1	3

Figure 20: Represent the attacking relation between different arguments based on the probability and shows the final accepted outcomes.

FN = false-negative

Equation 5.2 illustrates a probability equation where TP, TN, FP, FN define true-positive, true-negative, false-positive, and false-negative successively. True-positive means the actual condition is positive, but it is truly predicted as positive. Whereas, true-negative defines the actual condition is negative and it is truly predicted as negative. False-positive means the actual condition is negative, but it is falsely predicted as positive. Whereas, false-negative denotes the actual condition is positive, but it is falsely predicted as negative. The probability of result sets from different models is also calculated by the probability equation. It is a common equation for calculating the accuracy of a decision tree algorithm. Both equations are applied for calculating the accuracy of results. The accuracy of a result is regarded as the probability of an argument. The probability of arguments plays a significant role in the probabilistic argumentation framework for data mapping. In this scenario, equation 5.1 performs better than equation 5.2. The accuracy of equation 5.1 is comparably higher than equation 5.2.

Figure 21 explains the accuracy of DTC and DTR based on the amount of splitting data and probability function. Two different probability equations (eq.(5.1) and eq.(5.2)) are applied for calculating the accuracy of different models. Using eq.(5.1) in the case of 20% split data, the accuracy of DTC is 0.54 whereas it is 0.51 for DTR. On the other hand, it is 0.49 for DTC and 0.52 for DTR by using eq.(5.2). Using eq.(5.1) in the case of 100% split data, the accuracy of DTC is 0.89 whereas it is 0.87 for DTR. In contrast, it is 0.78 for DTC and 0.72 for DTR by using eq.(5.2). The accuracy increases with the increasing amount of splitting data. Analyzing the data from Figure 21 it can be said that, the efficiency of eq.(5.1) is better than eq.(5.2) for calculating accuracy.



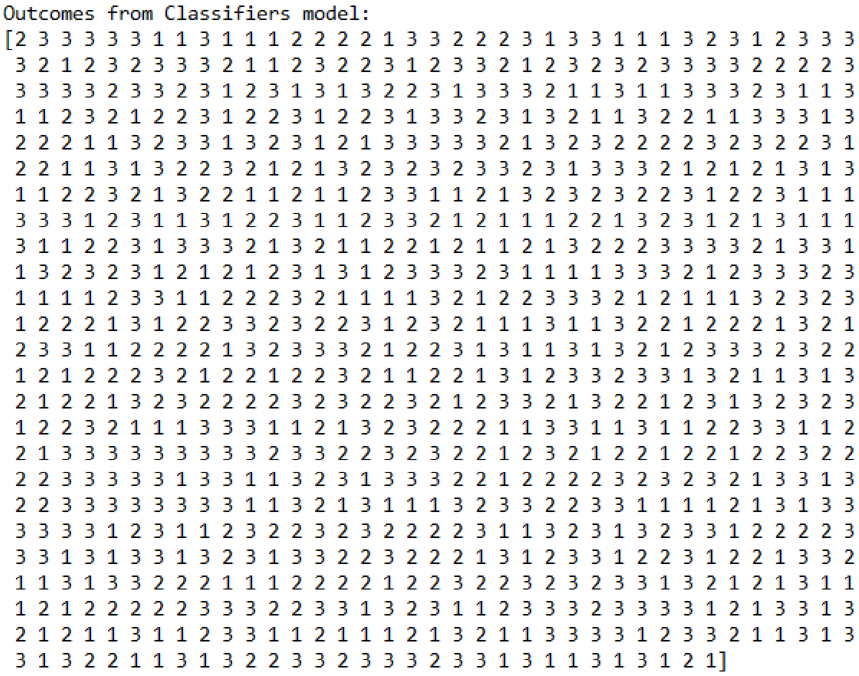
Split Data	Probability	Accuracy of DTC	Accuracy of DTR
20%	eq. (5.1)	0.54	0.51
	eq. (5.2)	0.49	0.52
40%	eq. (5.1)	0.61	0.66
	eq. (5.2)	0.66	0.59
80%	eq. (5.1)	0.84	0.79
	eq. (5.2)	0.80	0.74
100%	eq. (5.1)	0.89	0.87
	eq. (5.2)	0.78	0.72

Figure 21: Represent the accuracy of two models namely DTR and DTC based on different probability functions (eq.(5.1) and eq.(5.2)). The accuracy varies with the amount of splitting data.

5.2.2 Evaluation of the open-access data set

This section presents the evaluation of the open-access data set. In this scenario, the system takes 20% of testing data sets as input and predicts outcomes. The outcomes from different models (DTC and DTR) and PAF are explained below.

Figure 22 illustrates a predicted outcome set of DTC model. The outcome set contains a total of 919 values and three different class labels such as 1, 2, and 3. Outcomes from the outcome set of DTC act as arguments in PAF. For instance, the index positions 0, 1, 2, 3 of the outcome set contain 2, 3, 3, 3 as the outcomes, and they argue with the outcomes at index positions 0, 1, 2, 3 in the outcome set of DTR. Figure 23 depicts the graphical view of the outcome set generated from DTC. The X-axis illustrates the index and the Y-axis illustrates the class labels. The graph represents 919 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 2, 3, 3, 3, 3, 3, 1, 1, 3 respectively.



```
Outcomes from Classifiers model:
[2 3 3 3 3 3 1 1 3 1 1 1 2 2 2 2 1 3 3 2 2 2 3 1 3 3 1 1 1 3 2 3 1 2 3 3 3
 3 2 1 2 3 2 3 3 3 2 1 1 2 3 2 2 3 1 2 3 3 2 1 2 3 2 3 2 3 3 3 3 2 2 2 2 3
 3 3 3 3 2 3 3 2 3 1 2 3 1 3 1 3 2 2 3 1 3 3 3 2 1 1 3 1 1 3 3 3 2 3 1 1 3
 1 1 2 3 2 1 2 2 3 1 2 2 3 1 2 2 3 1 3 3 2 3 1 3 2 1 1 3 2 2 1 1 3 3 3 1 3
 2 2 2 1 1 3 2 3 3 1 3 2 3 1 2 1 3 3 3 3 3 2 1 3 2 3 2 2 2 2 3 2 3 2 2 3 1
 2 2 1 1 3 1 3 2 2 3 2 1 2 1 3 2 3 2 3 2 3 3 2 3 1 3 3 3 2 1 2 1 2 1 3 1 3
 1 1 2 2 3 2 1 3 2 2 1 1 2 1 1 2 3 3 1 1 2 1 3 2 3 2 3 2 2 3 1 2 2 3 1 1 1
 3 3 3 1 2 3 1 1 3 1 2 2 3 1 1 2 3 3 2 1 2 1 1 1 2 2 1 3 2 3 1 2 1 3 1 1 1
 3 1 1 2 2 3 1 3 3 3 2 1 3 2 1 1 2 2 1 2 1 1 2 1 3 2 2 2 3 3 3 3 2 1 3 3 1
 1 3 2 3 2 3 1 2 1 2 1 2 3 1 3 1 2 3 3 3 2 3 1 1 1 1 3 3 3 2 1 2 3 3 3 2 3
 1 1 1 1 2 3 3 1 1 2 2 2 3 2 1 1 1 1 3 2 1 2 2 3 3 3 2 1 2 1 1 1 3 2 3 2 3
 1 2 2 2 1 3 1 2 2 3 3 2 3 2 2 3 1 2 3 2 1 1 1 3 1 1 3 2 2 1 2 2 2 1 3 2 1
 2 3 3 1 1 2 2 2 2 1 3 2 3 3 3 2 1 2 2 3 1 3 1 1 3 1 3 2 1 2 3 3 3 2 3 2 2
 1 2 1 2 2 2 3 2 1 2 2 1 2 2 3 2 1 1 2 2 1 3 1 2 3 3 2 3 3 1 3 2 1 1 3 1 3
 2 1 2 2 1 3 2 3 2 2 2 2 3 2 3 2 2 3 2 1 2 3 3 2 1 3 2 2 1 2 3 1 3 2 3 2 3
 1 2 2 3 2 1 1 1 3 3 3 1 1 2 1 3 2 3 2 2 2 1 1 3 3 1 1 3 1 1 2 2 3 3 1 1 2
 2 1 3 3 3 3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 2 1 2 3 2 1 2 2 1 2 2 1 2 2 3 2 2
 2 2 3 3 3 3 3 1 3 3 1 1 3 2 3 1 3 3 3 2 2 1 2 2 2 2 3 2 3 2 3 2 1 3 3 1 3
 2 2 3 3 3 3 3 3 3 3 1 1 3 2 1 3 1 1 1 3 2 3 3 2 2 3 3 1 1 1 1 2 1 3 1 3 3
 3 3 3 3 1 2 3 1 1 2 3 2 2 3 2 3 2 2 2 2 3 1 1 3 2 3 1 3 2 3 3 1 2 2 2 2 3
 3 3 1 3 1 3 3 1 3 2 3 1 3 3 2 2 3 2 2 2 1 3 1 2 3 3 1 2 2 3 1 2 2 1 3 3 2
 1 1 3 1 3 3 2 2 2 1 1 1 2 2 2 2 1 2 2 3 2 2 3 2 3 2 3 3 1 3 2 1 2 1 3 1 1
 1 2 1 2 2 2 2 2 3 3 3 2 2 3 3 1 3 2 3 1 1 2 3 3 3 2 3 3 3 3 1 2 1 3 3 1 3
 2 1 2 1 1 3 1 1 2 3 3 1 1 2 1 1 1 2 1 3 2 1 1 3 3 3 3 1 2 3 3 2 1 1 3 1 3
 3 1 3 2 2 1 1 3 1 3 2 2 3 3 2 3 3 3 2 3 3 1 3 1 1 3 1 3 1 2 1]
```

Figure 22: Represent the generated outcome set from DTC that contains 919 values. The outcome set contains three class labels such as 1, 2, 3.

Figure 24 illustrates a predicted outcome set of DTR model. The outcome set contains a total of 919 values and three different class labels such as 1, 2, and 3. The outcome set of DTR is different than DTC. Outcomes from the outcome set of DTR also act as arguments in PAF. For instance, the index positions 0, 1, 2, 3 of the outcome set contain 2, 2, 1, 2 as the outcomes, and they argue with the outcomes at index positions 0, 1, 2, 3 in the outcome set of DTC. Figure 25 depicts the graphical view of the outcome set generated from DTR. The X-axis illustrates the index positions of existing values in the outcome set. The Y-axis illustrates the class labels. The outcome set contains three different class labels such as 1, 2, 3. The graph represents 919 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 2, 2, 1, 2, 1, 1, 2, 2, 1 respectively.

Figure 26 illustrates the similar index positions of the outcome sets of DTC and DTR where

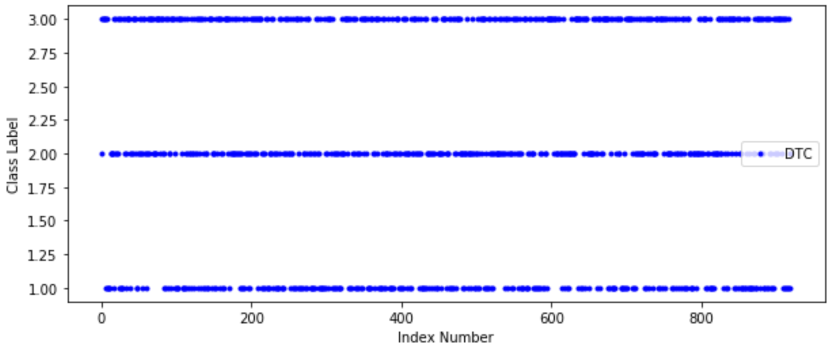
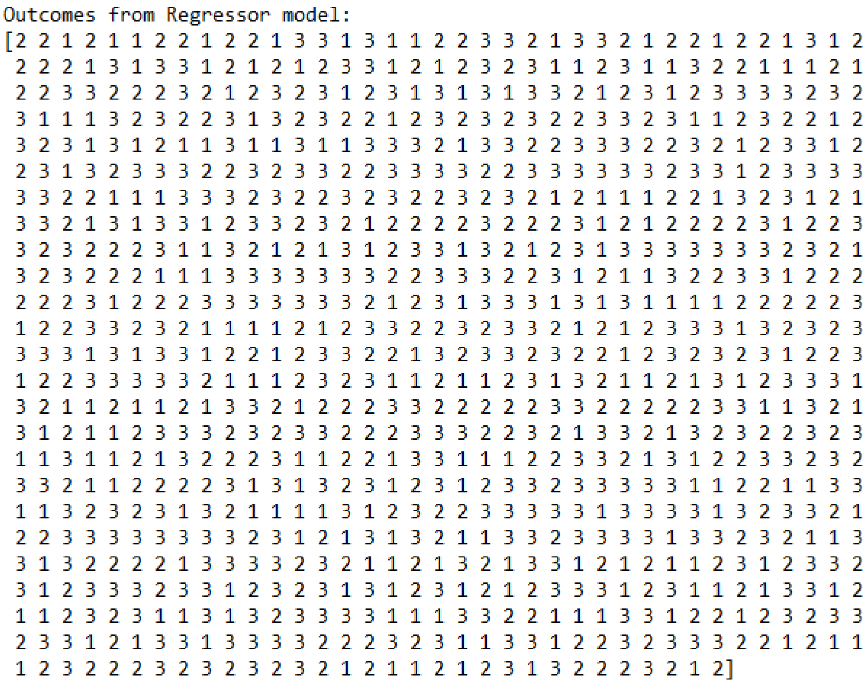


Figure 23: Represent the graphical illustration of the generated output set from DTC. Y-axis and X-axis illustrate the different class labels and the index number successively.



```
Outcomes from Regressor model:
[2 2 1 2 1 1 2 2 1 2 2 1 3 3 1 3 1 1 2 2 3 3 2 1 3 3 2 1 2 2 1 2 2 1 3 1 2
 2 2 2 1 3 1 3 3 1 2 1 2 1 2 3 3 1 2 1 2 3 2 3 1 1 2 3 1 1 3 2 2 1 1 1 2 1
 2 2 3 3 2 2 2 3 2 1 2 3 2 3 1 2 3 1 3 1 3 1 3 3 2 1 2 3 1 2 3 3 3 3 2 3 2
 3 1 1 1 3 2 3 2 2 3 1 3 2 3 2 2 1 2 3 2 3 2 3 2 2 3 3 2 3 1 1 2 3 2 2 1 2
 3 2 3 1 3 1 2 1 1 3 1 1 3 1 1 3 3 3 2 1 3 3 2 2 3 3 3 2 2 3 2 1 2 3 3 1 2
 2 3 1 3 2 3 3 3 2 2 3 2 3 3 2 2 3 3 3 3 2 2 3 3 3 3 3 3 2 3 3 1 2 3 3 3 3
 3 3 2 2 1 1 1 3 3 3 2 3 2 2 3 2 3 2 2 3 2 3 2 1 2 1 1 1 2 2 1 3 2 3 1 2 1
 3 3 2 1 3 1 3 3 1 2 3 3 2 3 2 1 2 2 2 2 3 2 2 2 3 1 2 1 2 2 2 2 3 1 2 2 3
 3 2 3 2 2 2 3 1 1 3 2 1 2 1 3 1 2 3 3 1 3 2 1 2 3 1 3 3 3 3 3 3 3 2 3 2 1
 3 2 3 2 2 2 1 1 1 3 3 3 3 3 3 3 2 2 3 3 3 2 2 3 1 2 1 1 3 2 2 3 3 1 2 2 2
 2 2 2 3 1 2 2 2 3 3 3 3 3 3 3 2 1 2 3 1 3 3 3 1 3 1 3 1 1 1 1 2 2 2 2 2 3
 1 2 2 3 3 2 3 2 1 1 1 1 2 1 2 3 3 2 2 3 2 3 3 2 1 2 1 2 3 3 3 1 3 2 3 2 3
 3 3 3 1 3 1 3 3 1 2 2 1 2 3 3 2 2 1 3 2 3 3 2 3 2 2 1 2 3 2 3 2 3 1 2 2 3
 1 2 2 3 3 3 3 3 2 1 1 1 2 3 2 3 1 1 2 1 1 2 3 1 3 2 1 1 2 1 3 1 2 3 3 3 1
 3 2 1 1 2 1 1 2 1 3 3 2 1 2 2 2 3 3 2 2 2 2 2 3 3 2 2 2 2 2 3 3 1 1 3 2 1
 3 1 2 1 1 2 3 3 3 2 3 2 3 3 2 2 2 3 3 3 2 2 3 2 1 3 3 2 1 3 2 3 2 2 3 2 3
 1 1 3 1 1 2 1 3 2 2 2 3 1 1 2 2 1 3 3 1 1 1 2 2 3 3 2 1 3 1 2 2 3 3 2 3 2
 3 3 2 1 1 2 2 2 2 3 1 3 1 3 2 3 1 2 3 1 2 3 3 2 3 3 3 3 3 1 1 2 2 1 1 3 3
 1 1 3 2 3 2 3 1 3 2 1 1 1 1 3 1 2 3 2 2 3 3 3 3 3 1 3 3 3 3 1 3 2 3 3 2 1
 2 2 3 3 3 3 3 3 3 3 2 3 1 2 1 3 1 3 2 1 1 3 3 2 3 3 3 3 1 3 3 2 3 2 1 1 3
 3 1 3 2 2 2 2 1 3 3 3 3 2 3 2 1 1 2 1 3 2 1 3 3 1 2 1 2 1 1 2 3 1 2 3 3 2
 3 1 2 3 3 3 2 3 3 1 2 3 2 3 1 3 1 2 3 1 2 1 2 3 3 3 1 2 3 1 1 2 1 3 3 1 2
 1 1 2 3 2 3 1 1 3 1 3 2 3 3 3 3 1 1 1 3 3 2 2 1 1 1 3 3 1 2 2 1 2 3 2 3 3
 2 3 3 1 2 1 3 3 1 3 3 3 3 2 2 2 3 2 3 1 1 3 3 1 2 2 3 2 3 3 3 2 2 1 2 1 1
 1 2 3 2 2 2 3 2 3 2 3 2 3 2 1 2 1 1 2 1 2 3 1 3 2 2 2 3 2 1 2]
```

Figure 24: Represent the generated outcome set from DTR that contains 919 values. The outcome set contains three class labels such as 1, 2, 3.

they produce similar values. Figure 27 represents similar values according to similar index positions (Fig.26). Both of the outcome sets contain 285 similar values according to similar index positions. Furthermore, the values are not attack each other. For example, at index positions 0, 11, 16, 19, 23, 24, 25, 27, 34 the outcome sets of DTC and DTR hold the outcomes 2, 1, 1, 2, 1, 3, 3, 1, 3.

Figure 28 illustrates the graphical illustration of both outcome sets generated from DTC and DTR. The light blue color indicates the result set of DTR whereas the red color indicates the result set of DTC. The X-axis and Y-axis depict the index number and the class labels respectively. The two graphs locate some points in the same locations that imply the values

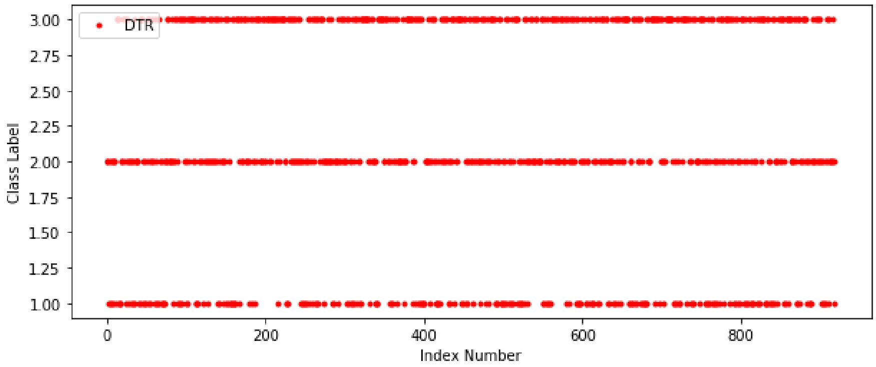
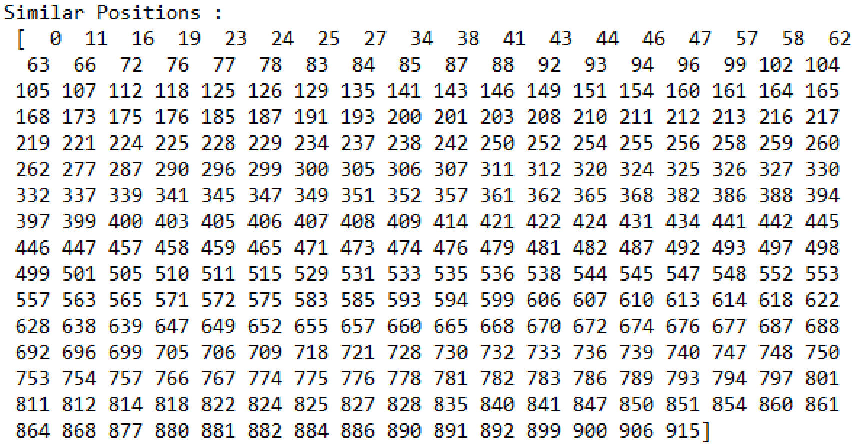
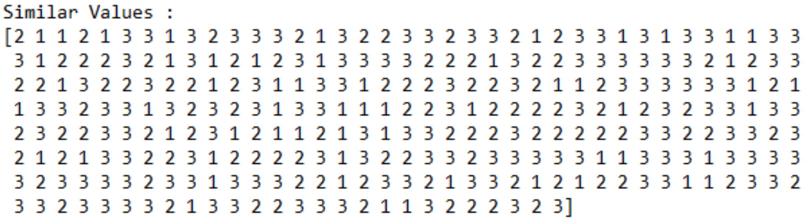


Figure 25: Represent graphical illustration of the generated output set from DTR. Y-axis and X-axis illustrate the different class labels and the index number respectively.



```
Similar Positions :
 [  0  11  16  19  23  24  25  27  34  38  41  43  44  46  47  57  58  62
  63  66  72  76  77  78  83  84  85  87  88  92  93  94  96  99 102 104
 105 107 112 118 125 126 129 135 141 143 146 149 151 154 160 161 164 165
 168 173 175 176 185 187 191 193 200 201 203 208 210 211 212 213 216 217
 219 221 224 225 228 229 234 237 238 242 250 252 254 255 256 258 259 260
 262 277 287 290 296 299 300 305 306 307 311 312 320 324 325 326 327 330
 332 337 339 341 345 347 349 351 352 357 361 362 365 368 382 386 388 394
 397 399 400 403 405 406 407 408 409 414 421 422 424 431 434 441 442 445
 446 447 457 458 459 465 471 473 474 476 479 481 482 487 492 493 497 498
 499 501 505 510 511 515 529 531 533 535 536 538 544 545 547 548 552 553
 557 563 565 571 572 575 583 585 593 594 599 606 607 610 613 614 618 622
 628 638 639 647 649 652 655 657 660 665 668 670 672 674 676 677 687 688
 692 696 699 705 706 709 718 721 728 730 732 733 736 739 740 747 748 750
 753 754 757 766 767 774 775 776 778 781 782 783 786 789 793 794 797 801
 811 812 814 818 822 824 825 827 828 835 840 841 847 850 851 854 860 861
 864 868 877 880 881 882 884 886 890 891 892 899 900 906 915]
```

Figure 26: Represent the index positions of similar values from both outcome sets.



```
Similar Values :
[2 1 1 2 1 3 3 1 3 2 3 3 3 2 1 3 2 2 3 3 2 3 3 2 1 2 3 3 1 3 1 3 3 1 1 3 3
 3 1 2 2 2 3 2 1 3 1 2 1 2 3 1 3 3 3 3 2 2 2 1 3 2 2 3 3 3 3 3 3 2 1 2 3 3
 2 2 1 3 2 2 3 2 2 1 2 3 1 1 3 3 1 2 2 2 3 2 2 3 2 1 1 2 3 3 3 3 3 3 1 2 1
 1 3 3 2 3 3 1 3 2 3 2 3 1 3 3 1 1 1 2 2 3 1 2 2 2 2 3 2 1 2 3 2 3 3 1 3 3
 2 3 2 2 3 3 2 1 2 3 1 2 1 1 2 1 3 1 3 3 2 2 2 3 2 2 2 2 2 3 3 2 2 3 3 2 3
 2 1 2 1 3 3 2 2 3 1 2 2 2 2 3 1 3 2 2 3 3 2 3 3 3 3 3 1 1 3 3 3 1 3 3 3 3
 3 2 3 3 3 3 2 3 3 1 3 3 3 2 2 1 2 3 3 2 1 3 3 2 1 2 1 2 2 3 3 1 1 2 3 3 2
 3 3 2 3 3 3 3 2 1 3 3 2 2 3 3 3 2 1 1 3 2 2 2 3 2 3]
```

Figure 27: Represent similar values based on the similar index positions of both outcome sets.

of the identical location are similar. Both outcome sets contain the same class labels in the identical location and they will not attack each other. There are 285 similar values that exist in similar positions of both outcome sets. For example, index positions 0, 11, 16, 19, 23, 24, 25, 27, 34 present the class labels 2, 1, 1, 2, 1, 3, 3, 1, 3 respectively for both models

(DTC and DTR).

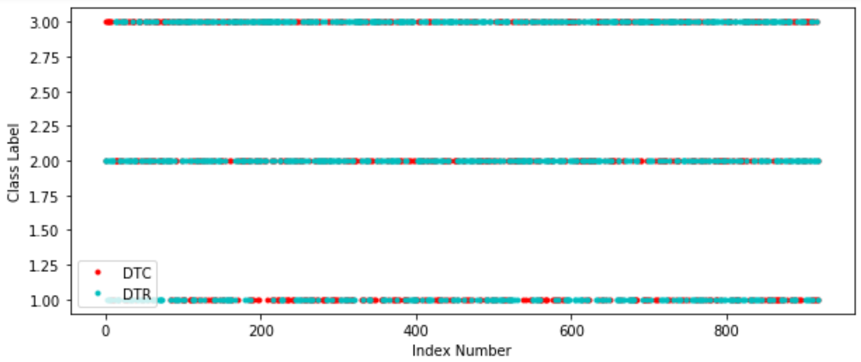
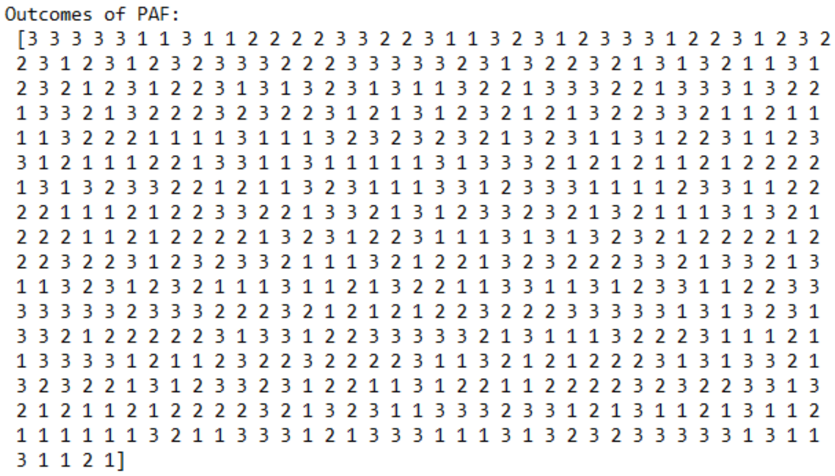


Figure 28: Represent the graphical presentation of the generated output sets from DTC and DTR. Y-axis and X-axis illustrate the different class labels and the index position successively. Both of the result sets have some similar values in some points.

Figure 29 illustrates the accepted outcome set of PAF. The outcome set contains a total of 634 values. The outcome set of PAF is different from the outcome sets of DTC and DTR. The outcomes from two different outcome sets (DTC and DTR) participate in an argument process through PAF. They argue among themselves for finalizing outcomes (accepted outcomes). PAF produces an outcome set that contains only accepted outcomes. For example, The outcome sets of DTC and DTR hold the outcomes 3 and 2 respectively at index position 1. The outcomes 3 and 2 argue among themselves based on probability. Finally, 3 is considered as an acceptable outcome. The accepted outcome 3 locates the index position 0 in the outcome set of PAF. Figure 30 depicts the graphical view of the accepted outcome set generated from PAF. The X-axis illustrates the index positions. The Y-axis illustrates the class labels. The data set has three different class labels such as 1, 2, 3. The graph represents 634 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 3, 3, 3, 3, 3, 1, 1, 3, 1 respectively.

Figure 31 represents only 12 data from the experimented outcome sets of DTC, DTR, and PAF and presents the attacking relations and accepted outcomes. It shows where the arguments attack each other based on probability and different conditions. An argument with high probability attacks the argument that holds less probability. If the similar index positions of both outcome sets (DTC and DTR) hold similar values then they will not attack. For example, at index positions 0, 11, both of the outcome sets hold 2,1 as outcomes and they do not attack. At index position 2, the outcomes sets of DTC and DTR hold 3, 1 as the outcomes. The probability of outcome 3 is 0.77, whereas, it is 0.57 for outcome 1. Outcome 3 attacks outcome 1. Finally, outcome 3 is accepted through an argumentation process. The outcome sets of DTC and DTR hold the outcomes 3 and 1 respectively at index position 4. The probability of outcome 1 is 0.92, whereas, it is 0.76 for outcome 3. The outcome 1 attacks outcome 3 and 1 is regarded as the accepted outcome. In similarly, The outcome sets of DTC and DTR hold the outcomes 3 and 1 respectively at index position 8. The probability of outcome 1 is 0.89, whereas, it is 0.86 for outcome 3. The outcome 1 attacks outcome 3 and 1 is regarded as the accepted outcome. Similarly, the rest of the outcomes of DTC and DTR attack each other based on probability and PAF produces the outcome set of accepted outcomes.



```
Outcomes of PAF:
 [3 3 3 3 3 1 1 3 1 1 2 2 2 2 3 3 2 2 3 1 1 3 2 3 1 2 3 3 3 1 2 2 3 1 2 3 2
  2 3 1 2 3 1 2 3 2 3 3 3 2 2 2 3 3 3 3 3 2 3 1 3 2 2 3 2 1 3 1 3 2 1 1 3 1
  2 3 2 1 2 3 1 2 2 3 1 3 1 3 2 3 1 3 1 1 3 2 2 1 3 3 3 2 2 1 3 3 3 1 3 2 2
  1 3 3 2 1 3 2 2 2 3 2 3 2 2 3 1 2 1 3 1 2 3 2 1 2 1 3 2 2 3 3 2 1 1 2 1 1
  1 1 3 2 2 2 1 1 1 1 3 1 1 1 3 2 3 2 3 2 3 2 1 3 2 3 1 1 3 1 2 2 3 1 1 2 3
  3 1 2 1 1 1 2 2 1 3 3 1 1 3 1 1 1 1 1 3 1 3 3 3 2 1 2 1 2 1 1 2 1 2 2 2 2
  1 3 1 3 2 3 3 2 2 1 2 1 1 3 2 3 1 1 1 3 3 1 2 3 3 3 1 1 1 1 2 3 3 1 1 2 2
  2 2 1 1 1 2 1 2 2 3 3 2 2 1 3 3 2 1 3 1 2 3 3 2 3 2 1 3 2 1 1 1 3 1 3 2 1
  2 2 2 1 1 2 1 2 2 2 2 1 3 2 3 1 2 2 3 1 1 1 3 1 3 1 3 2 3 2 1 2 2 2 2 1 2
  2 2 3 2 2 3 1 2 3 2 3 3 2 1 1 1 3 2 1 2 2 1 3 2 3 2 2 2 3 3 2 1 3 3 2 1 3
  1 1 3 2 3 1 2 3 2 1 1 1 3 1 1 2 1 3 2 2 1 1 3 3 1 1 3 1 2 3 3 1 1 2 2 3 3
  3 3 3 3 3 2 3 3 3 2 2 2 3 2 1 2 1 2 1 2 2 3 2 2 2 3 3 3 3 3 1 3 1 3 2 3 1
  3 3 2 1 2 2 2 2 2 3 1 3 3 1 2 2 3 3 3 3 3 2 1 3 1 1 1 3 2 2 2 3 1 1 1 2 1
  1 3 3 3 3 1 2 1 1 2 3 2 2 3 2 2 2 2 3 1 1 3 2 1 2 1 2 2 2 3 1 3 1 3 3 2 1
  3 2 3 2 2 1 3 1 2 3 3 2 3 1 2 2 1 1 3 1 2 2 1 1 2 2 2 2 3 2 3 2 2 3 3 1 3
  2 1 2 1 1 2 1 2 2 2 2 3 2 1 3 2 3 1 1 3 3 3 2 3 3 1 2 1 3 1 1 2 1 3 1 1 2
  1 1 1 1 1 1 3 2 1 1 3 3 3 1 2 1 3 3 3 1 1 1 3 1 3 2 3 2 3 3 3 3 3 1 3 1 1
  3 1 1 2 1]
```

Figure 29: Represent the accepted outcome set of PAF that contains 634 values. The outcome set contains three class labels such as 1, 2, 3.

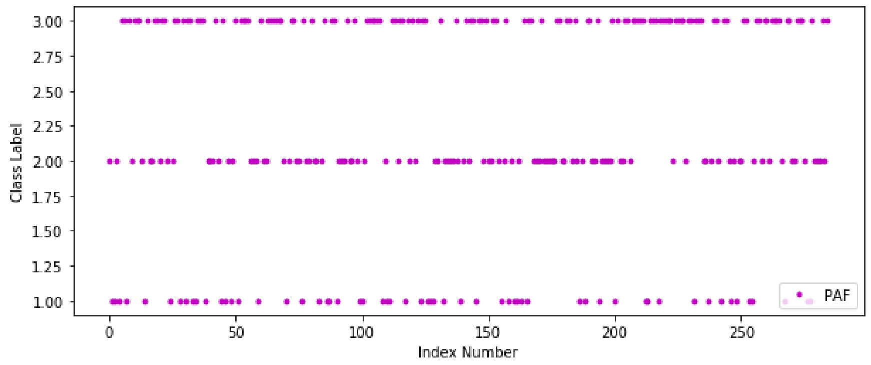
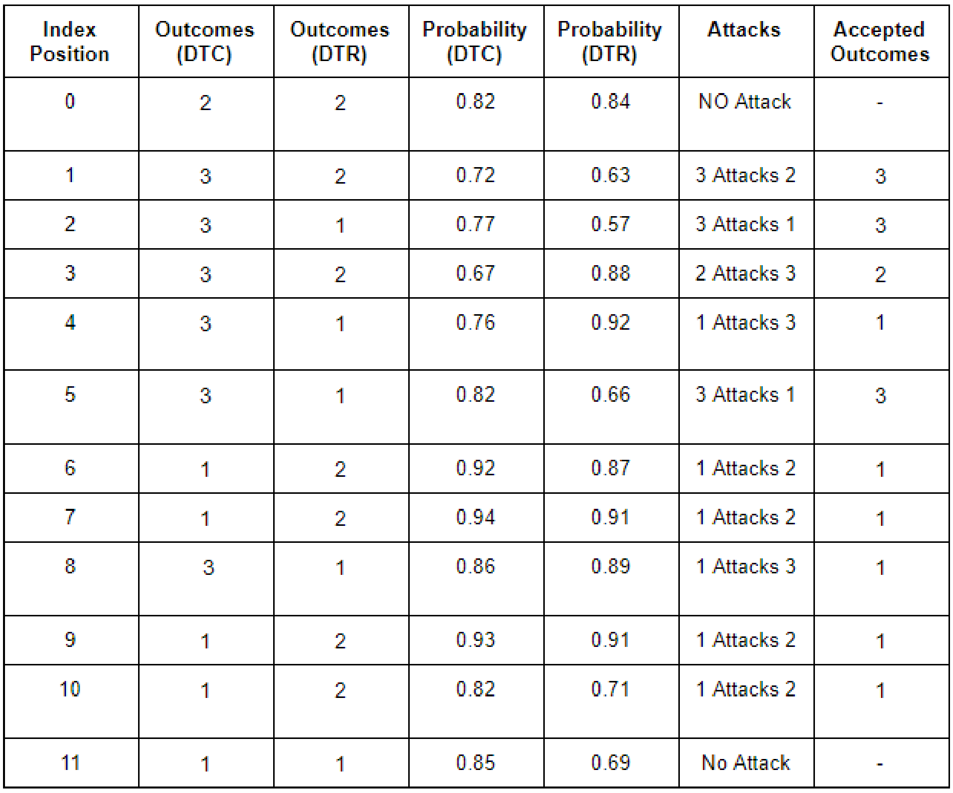


Figure 30: Represent the graphical illustration of the accepted output set of PAF. Y-axis and X-axis illustrate the different class labels and the index number successively.

5.2.3 Evaluation of without an argumentation framework

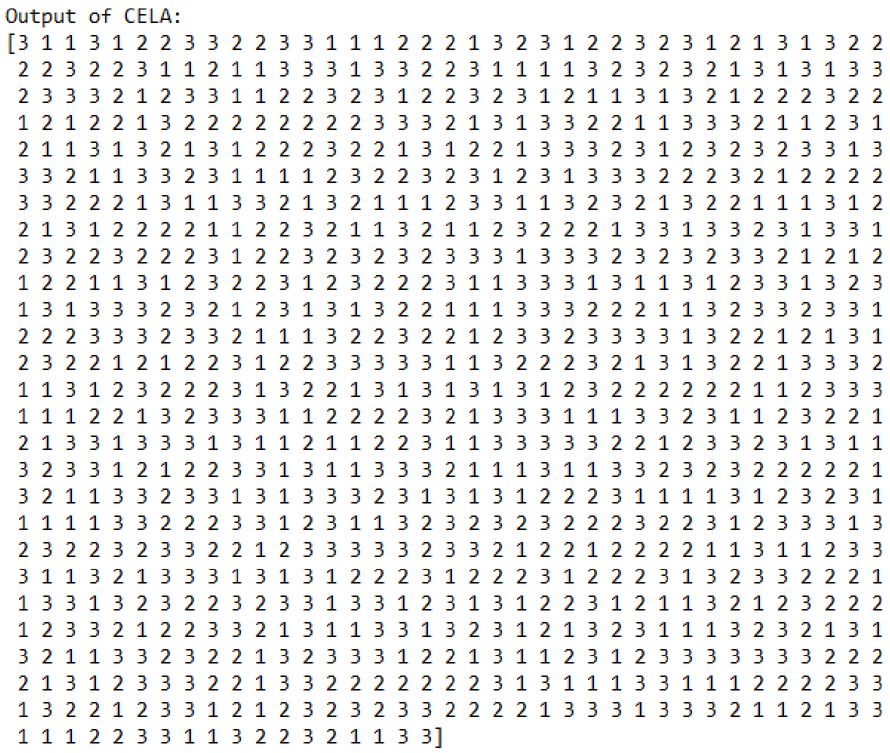
This section represents the evaluation of an ensemble learning algorithm without an argumentation framework. A classical ensemble algorithm (CELA) [21], e.g., a random forest, is evaluated with an open-access data set. The 20% of randomly selected data from the data set constructs the test data set and the rest of the data constructs the training data set. The algorithm has two models (Decision tree classifiers) and it generates a different set of outcomes. Figure 32 illustrates the final outcome set. The outcome set contains 980 values. It contains the class labels 1, 2, and 3. Figure 33 presents the graphical representation of the outcome set, where the X-axis and Y-axis represent the index position and class labels successively. The graph represents 980 outcomes. For example, index positions 0, 1, 2, 3, 4, 5, 6, 7, 8 present the class labels 3, 1, 1, 3, 1, 2, 2, 3, 3 respectively.

The produced outcome set of CELA is different from the outcome set of an ensemble learning algorithm with an argumentation approach. The accuracy of CELA is less than an ensemble learning algorithm with an argumentation approach. The accuracy of CELA is 79%, whereas the accuracy of an ensemble learning algorithm with an argumentation approach is 88%. The evaluation will be extended with different data sets in future work. An argumentation approach works as a decision fusion to combining the outcomes from different models. The integration of an argumentation approach improves the performance of an ensemble learning approach.



Index Position	Outcomes (DTC)	Outcomes (DTR)	Probability (DTC)	Probability (DTR)	Attacks	Accepted Outcomes
0	2	2	0.82	0.84	NO Attack	-
1	3	2	0.72	0.63	3 Attacks 2	3
2	3	1	0.77	0.57	3 Attacks 1	3
3	3	2	0.67	0.88	2 Attacks 3	2
4	3	1	0.76	0.92	1 Attacks 3	1
5	3	1	0.82	0.66	3 Attacks 1	3
6	1	2	0.92	0.87	1 Attacks 2	1
7	1	2	0.94	0.91	1 Attacks 2	1
8	3	1	0.86	0.89	1 Attacks 3	1
9	1	2	0.93	0.91	1 Attacks 2	1
10	1	2	0.82	0.71	1 Attacks 2	1
11	1	1	0.85	0.69	No Attack	-

Figure 31: Represent the attacking relations between different arguments based on the probabilities of the arguments and shows the final accepted values.



```
Output of CELA:
[3 1 1 3 1 2 2 3 3 2 2 3 3 1 1 1 2 2 2 1 3 2 3 1 2 2 3 2 3 1 2 1 3 1 3 2 2
 2 2 3 2 2 3 1 1 2 1 1 3 3 3 1 3 3 2 2 3 1 1 1 1 3 2 3 2 3 2 1 3 1 3 1 3 3
 2 3 3 3 2 1 2 3 3 1 1 2 2 3 2 3 1 2 2 3 2 3 1 2 1 1 3 1 3 2 1 2 2 2 3 2 2
 1 2 1 2 2 1 3 2 2 2 2 2 2 2 2 3 3 3 2 1 3 1 3 3 2 2 1 1 3 3 3 2 1 1 2 3 1
 2 1 1 3 1 3 2 1 3 1 2 2 2 3 2 2 1 3 1 2 2 1 3 3 3 2 3 1 2 3 2 3 2 3 3 1 3
 3 3 2 1 1 3 3 2 3 1 1 1 1 2 3 2 2 3 2 3 1 2 3 1 3 3 3 2 2 2 3 2 1 2 2 2 2
 3 3 2 2 2 1 3 1 1 3 3 2 1 3 2 1 1 1 2 3 3 1 1 3 2 3 2 1 3 2 2 1 1 1 3 1 2
 2 1 3 1 2 2 2 2 1 1 2 2 3 2 1 1 3 2 1 1 2 3 2 2 2 1 3 3 1 3 3 2 3 1 3 3 1
 2 3 2 2 3 2 2 2 3 1 2 2 3 2 3 2 3 2 3 3 3 1 3 3 3 2 3 2 3 2 3 3 2 1 2 1 2
 1 2 2 1 1 3 1 2 3 2 2 3 1 2 3 2 2 2 3 1 1 3 3 3 1 3 1 1 3 1 2 3 3 1 3 2 3
 1 3 1 3 3 3 2 3 2 1 2 3 1 3 1 3 2 2 1 1 1 3 3 3 2 2 2 1 1 3 2 3 3 2 3 3 1
 2 2 2 3 3 3 2 3 3 2 1 1 1 3 2 2 3 2 2 1 2 3 3 2 3 3 3 3 1 3 2 2 1 2 1 3 1
 2 3 2 2 1 2 1 2 2 3 1 2 2 3 3 3 3 3 1 1 3 2 2 2 3 2 1 3 1 3 2 2 1 3 3 3 2
 1 1 3 1 2 3 2 2 2 3 1 3 2 2 1 3 1 3 1 3 1 3 1 2 3 2 2 2 2 2 2 1 1 2 3 3 3
 1 1 1 2 2 1 3 2 3 3 3 1 1 2 2 2 2 3 2 1 3 3 3 1 1 1 3 3 2 3 1 1 2 3 2 2 1
 2 1 3 3 1 3 3 3 1 3 1 1 2 1 1 2 2 3 1 1 3 3 3 3 3 2 2 1 2 3 3 2 3 1 3 1 1
 3 2 3 3 1 2 1 2 2 3 3 1 3 1 1 3 3 3 2 1 1 1 3 1 1 3 3 2 3 2 3 2 2 2 2 2 1
 3 2 1 1 3 3 2 3 3 1 3 1 3 3 3 2 3 1 3 1 3 1 2 2 2 3 1 1 1 1 3 1 2 3 2 3 1
 1 1 1 1 3 3 2 2 2 3 3 1 2 3 1 1 3 2 3 2 3 2 3 2 2 2 3 2 2 3 1 2 3 3 3 1 3
 2 3 2 2 3 2 3 3 2 2 1 2 3 3 3 3 3 2 3 3 2 1 2 2 1 2 2 2 2 1 1 3 1 1 2 3 3
 3 1 1 3 2 1 3 3 3 1 3 1 3 1 2 2 2 3 1 2 2 2 3 1 2 2 2 3 1 3 2 3 3 2 2 2 1
 1 3 3 1 3 2 3 2 2 3 2 3 3 1 3 3 1 2 3 1 3 1 2 2 3 1 2 1 1 3 2 1 2 3 2 2 2
 1 2 3 3 2 1 2 2 3 3 2 1 3 1 1 3 3 1 3 2 3 1 2 1 3 2 3 1 1 1 3 2 3 2 1 3 1
 3 2 1 1 3 3 2 3 2 2 1 3 2 3 3 3 1 2 2 1 3 1 1 2 3 1 2 3 3 3 3 3 3 3 2 2 2
 2 1 3 1 2 3 3 3 2 2 1 3 3 2 2 2 2 2 2 2 3 1 3 1 1 1 3 3 1 1 1 2 2 2 2 3 3
 1 3 2 2 1 2 3 3 1 2 1 2 3 2 3 2 3 3 2 2 2 2 1 3 3 3 1 3 3 3 2 1 1 2 1 3 3
 1 1 1 2 2 3 3 1 1 3 2 2 3 2 1 1 3 3]
```

Figure 32: Represent the outcome set of a classical ensemble learning algorithm. The outcome set contains 980 outcomes.

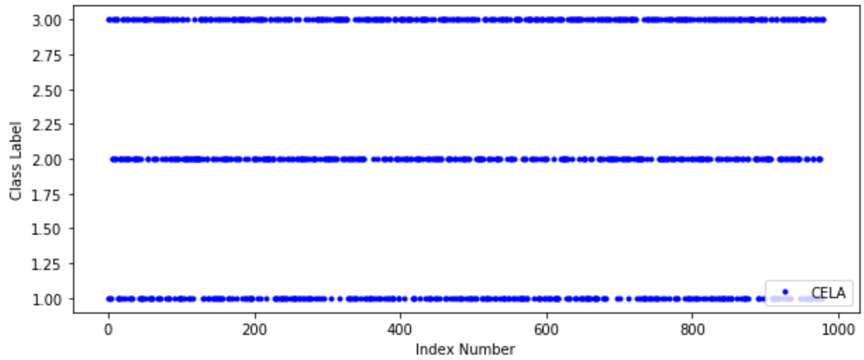


Figure 33: Represent the graphical representation of the outcome set of a classical ensemble learning algorithm, where the X-axis and Y-axis represent the index position and class labels successively.

6 Open Access Library

This chapter introduces a newly developed open-access library of the proposed system namely *Basicensembleargumentation*. The functionality and user manual of the library are explained in this chapter. The library is available at "pypi.org" and "github.com".

Github Link: https://github.com/Istiak1992/BasicEnsembleArgumentation.git

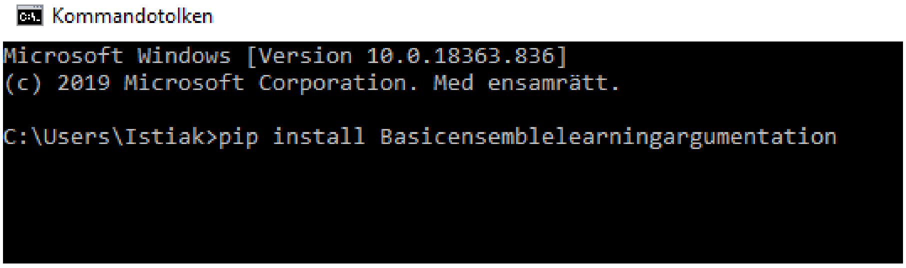
6.1 Functionality

The library has several methods namely *Classifiers*,*Regressors*,*Argumentation*,*andPlotGraph*. *Classifiers* method classifies a data set and predicts outcomes using a decision tree classifier algorithm. *Regressors* method classifies a data set and predicts outcomes using a decision tree regressor algorithm. *Argumentation* method conducts an argumentation process and generates accepted outcomes. It also represents the data set of similar values and index positions. *Classifiers* and *Regressors* methods act like two different models and they produce different sets of outcomes. *Argumentation* method receives the produced sets of outcomes as input parameters and pursues an argumentation process. In the argumentation process, outcomes of different models attack each other based on the probability function. If the outcomes of both models are similar, then they will not attack each other. The probability function also plays a crucial role to select the status of an argument. There are three kinds of status e.g accepted, unaccepted, and undecided. *PlotGraph* method generates a total of four graphs. The first graph represents the outcomes of the decision tree classifier model. The second graph represents the outcomes from the decision tree regressor model. The third graph and fourth graph illustrate the outcomes from both models and the accepted outcomes from PAF respectively. The third graph also presents the attacking relations of arguments that participate in the argument process. Furthermore, the library package contains files such as *setup.txt*,*README.md*,*MANIFEST.in*,*LICENSE.txt*,*CHANGELOG.txt*,*requirement.txt*, and *ensembleArgument.py*. *setup.txt* file contains all kinds of setup and installation requirements. *LICENSE.txt* file contains the license policy provided by MIT. *MANIFEST.in* file helps to work with different types of files (e.g. .txt, .py). *README.md* file explains the user manual and requirements to use the library. *CHANGELOG.txt* file represents information about the release date and version of the library. *requirement.txt* file contains the required Python packages. *ensembleArgument.py* file contains the definition of all of the methods.

6.2 User Manual

The user needs to set up the environment for using the library. Python 3.5 (minimum version), pandas, numpy, matplotlib, and sklearn are the required packages for the environment. A user needs to follow the following steps to use the library.

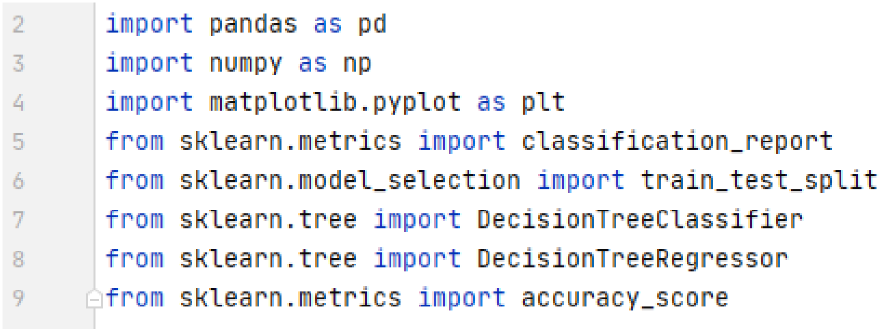
- Set up the environment with the required packages (panda, numpy, matplotlib, and sklearn).
- Install the library package.
  Command Line: **pip install Basicensemblelearningargumentation** (Figure 34).



```
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. Med ensamrätt.

C:\Users\Istiak>pip install Basicensemblelearningargumentation
```

Figure 34: Install the library package typing the command line in the command interpreter.

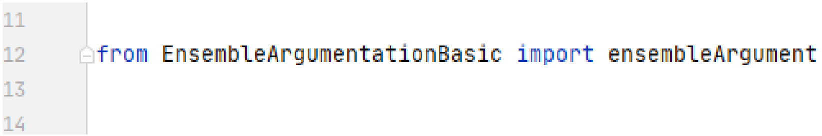
- Import packages in the python file (Figure 35).



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import accuracy_score
```

Figure 35: Import the packages in the test python file (e.g.text.py).

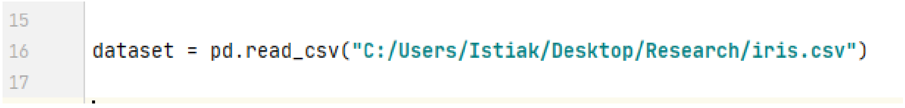
- Import an additional package (Figure 36).



```python
from EnsembleArgumentationBasic import ensembleArgument
```

Figure 36: Import an additional packages in the test python file (e.g.test.py).

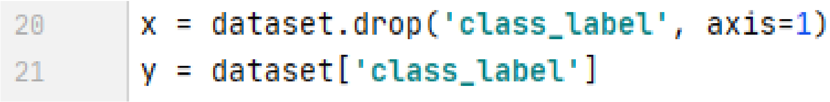
- Upload a data set using pandas (Figure 37).



```python
dataset = pd.read_csv("C:/Users/Istiak/Desktop/Research/iris.csv")
```

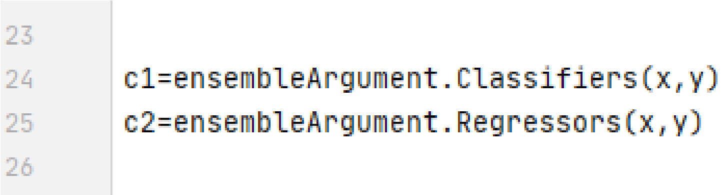
Figure 37: Upload a data set (e.g. variableName = pd.read_csv(“directory/fileName”)).

- Define the independent and dependant variables. (Figure 38).
- Call the two functions namely Classifiers and Regressors and instantiate the two functions into two different variables such as c1 and c2. (Figure 39).



```
x = dataset.drop('class_label', axis=1)
y = dataset['class_label']
```

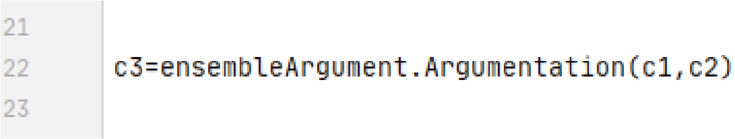
Figure 38: Represent the independent (x) and dependant (y) variables. “class _label” is the last column name of the data set that contains class labels.



```
c1=ensembleArgument.Classifiers(x,y)
c2=ensembleArgument.Regressors(x,y)
```

Figure 39: Assign the outcomes of two models into two different variables e.g. c1, c2.

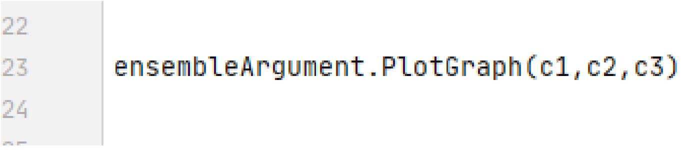
- Call the Argumentation function with the required parameters (e.g.c1, c2). Instantiate the function in a variable (e.g. c3) (Figure 40).



```
c3=ensembleArgument.Argumentation(c1,c2)
```

Figure 40: Represent how to call the Argumentation function with the required parameters and instantiate in a variable.

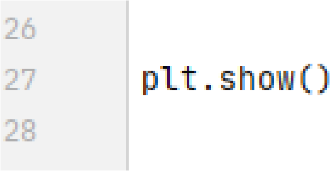
- Call the PlotGraph function with the required parameters (e.g.c1, c2, c3) (Figure 41).



```
ensembleArgument.PlotGraph(c1,c2,c3)
```

Figure 41: Represent how to call the PlotGraph function with the required parameters.

- Call the show() function to represent the graphical views (Figure 42).



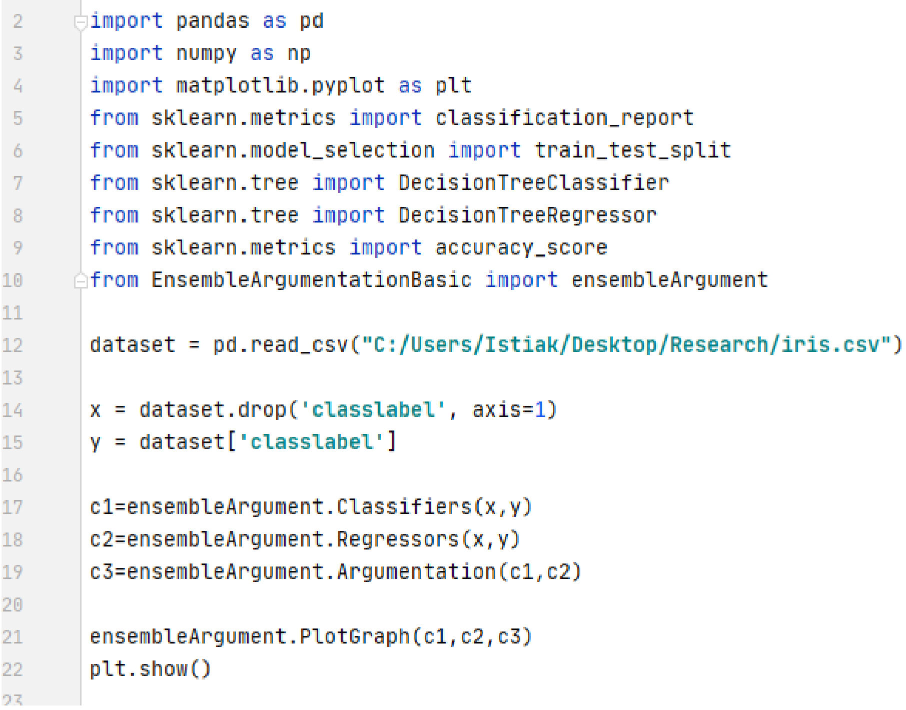
```
plt.show()
```

Figure 42: Represent how to call the show() function to represent the graphical views.

6.3 Example

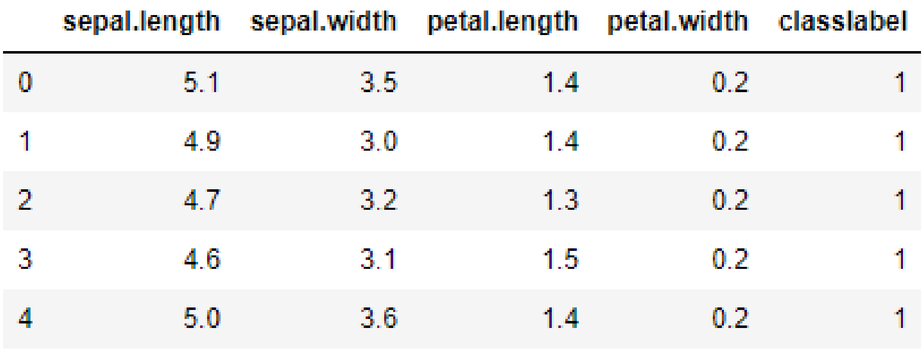
The following example (Figure 43) helps the user about using the library. To use the library user should install the required packages (section User Manual). The user should create a python file e.g. test.py and import all of the required packages in the python file (test.py). In the next step, the user should upload a data set (Figure 37) by using pandas and define the dependant and independent variables. Call two building functions namely Classifiers and Regressors with the required parameters. Then the user should call Argumentation and PlotGraph functions with the required parameters to generate reliable outcomes and graphical representation of the outcomes successively. The user should call the show() function for the sake of representing the graphical views. The outcomes of Classifiers and Regressors functions should be placed as parameters in the Argumentation function.

Example 1 (Fig.43) explains: how to implement the library in a python file. An open access data set (Fig.44) is used for the example.



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import accuracy_score
from EnsembleArgumentationBasic import ensembleArgument

dataset = pd.read_csv("C:/Users/Istiak/Desktop/Research/iris.csv")

x = dataset.drop('classlabel', axis=1)
y = dataset['classlabel']

c1=ensembleArgument.Classifiers(x,y)
c2=ensembleArgument.Regressors(x,y)
c3=ensembleArgument.Argumentation(c1,c2)

ensembleArgument.PlotGraph(c1,c2,c3)
plt.show()
```

Figure 43: The figure represents an example using the library.



	sepal.length	sepal.width	petal.length	petal.width	classlabel
0	5.1	3.5	1.4	0.2	1
1	4.9	3.0	1.4	0.2	1
2	4.7	3.2	1.3	0.2	1
3	4.6	3.1	1.5	0.2	1
4	5.0	3.6	1.4	0.2	1

Figure 44: Represent a data set that is used for example.1.

6.4 Evaluation

This section discusses the generated outcomes by executing example.1 (Figure 43). In this scenario, the Classifiers model (First model) generates the following result set.
Outcomes of Classifiers model: [2 3 2 3 1 1 1 3 3 3 2 2 3 1 3 3 3 2 1 3 3 2 3 1 3 3 1 2 1 1 1 2 3 1 1 1 2 2 2 1 2 2 2 3 2 2 3 2 1 1 3 3 1 3 3 2 1 3 2 3 3 1 2 2 1 1 3 3 1 2 2 1 3 2 3 3 2 3 3 2 1 3 3 1 2 1 3 3 3 1]

Whereas, the Regressors model (Second model) generates a different result set.
Outcomes of Regressors model: [2 3 1 3 3 1 2 2 2 2 1 3 1 2 2 3 3 2 3 3 1 2 3 2 2 2 3 1 3 1 2 1 3 2 1 2 2 1 3 1 1 2 3 1 3 3 1 2 2 2 3 3 1 2 2 1 1 1 3 2 2 2 2 1 1 1 3 2 2 2 2 2 1 2 2 3 1 1 3 1 2 2 1 1 3 1 2 1 2 3]
Both models generate some similar results in the same index positions. Similar results will not attack each other. The similar result set and the index positions of the similar values are presented below. Similar Values: [2 3 3 1 3 3 2 3 2 3 1 3 1 2 1 2 2 3 3 1 1 2 1 1 3 2 2 2 3 3 1 1]

Positions of the similar values: [0 1 3 5 15 16 17 19 21 22 29 32 34 36 39 41 47 50 51 52 56 62 64 65 66 69 70 73 75 78 83 85]

Argumentation function produces a result set that is different from the result sets of *Classifiers* and *Regressors* models. The result set of *Argumentation* function illustrates only the accepted arguments. The result set is presented below.
Accepted outcomes of PAF: [2 1 1 3 3 3 2 2 3 1 3 1 3 1 3 3 1 2 1 1 2 1 1 2 2 2 2 3 2 2 3 1 1 3 3 2 3 2 3 3 1 2 3 1 1 3 3 2 3 2 1 3 3 2 3 3 3 1]

6.4.1 Graphical Representation

This section illustrates the graphical representations of different result sets. In the graph, X-axis and Y-axis define the index position and the value successively. Figure 45 illustrates the graphical representation of the result set of *Classifiers* model. Figure 46 illustrates the graphical representation of the result set of *Regressors* model. Figure 47 illustrates the graphical representation of the result sets of both models. The intersecting points denote that they will not attack. Figure 48 represents the graphical representation of the accepted result set generated by an argumentation approach.

6.5 Target User

A user such as a researcher, student, and professional personnel can use the library for several purposes (data classification, data analysis, prediction, and so on). The user can generate accurate and reliable outcomes using the library. It saves a lot of time of a user because a user does not need to write a lot of lines of code. It is necessary, a user should have a mature skill in Python language or other programming languages to use the library.

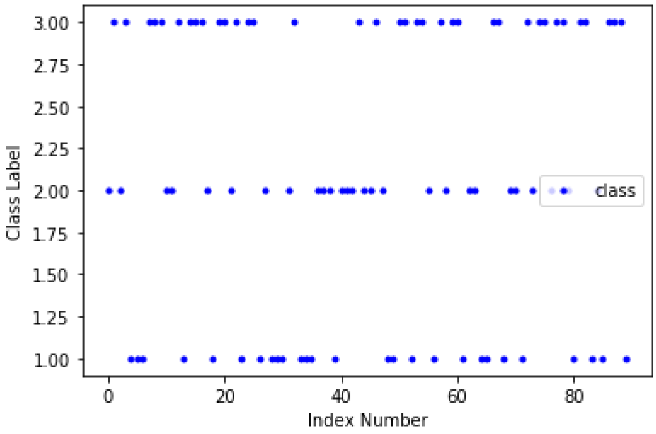


Figure 45: The figure represents a result set of $Classifiers$ model. X-axis and Y-axis represent index position and class label successively.

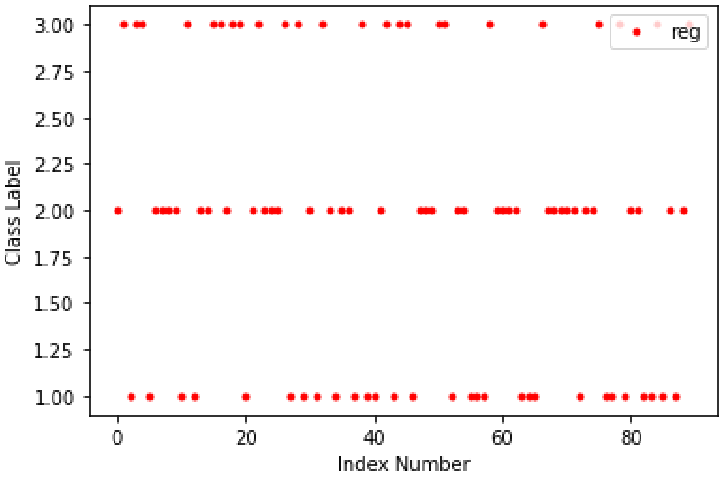


Figure 46: The figure represents a result set of *Regressors* model. X-axis and Y-axis represent index position and class label successively.

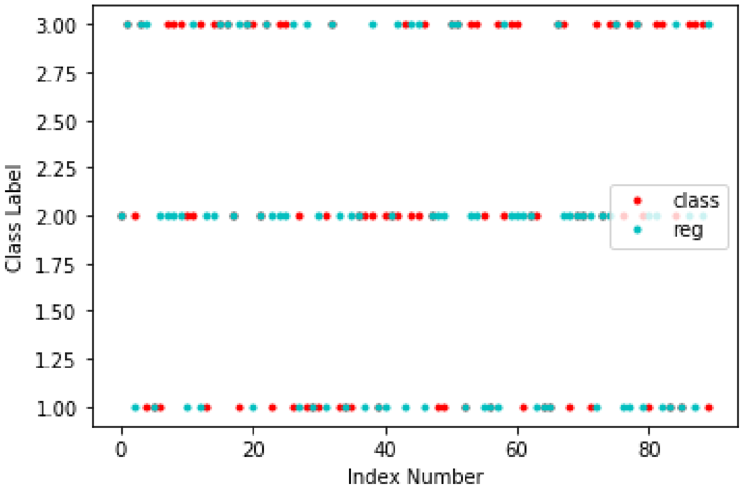


Figure 47: The figure represents a result sets of both models. X-axis and Y-axis represent index position and class label successively.The intersecting points denote that they will not attack.

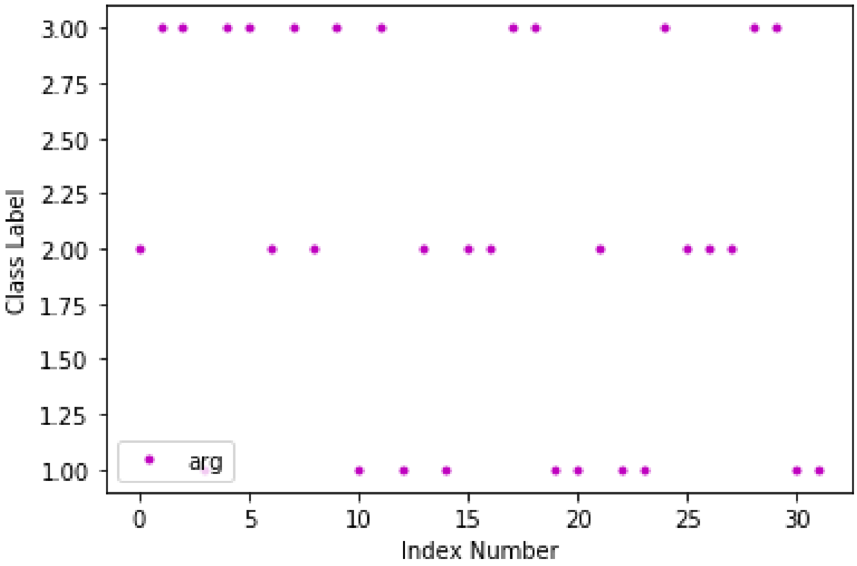


Figure 48: The figure represents the result set generated by an argumentation approach. X-axis and Y-axis represent index position and class label successively.

7 Discussion

This study has investigated an approach for classifying the data sets using different models of learning algorithms (an ensemble learning architecture) and the probabilistic argumentation framework generates reliable outputs by using the outcomes from different models. ELA usually executes a base learning algorithm multiple times and creates a voting classifier for the resulting hypothesis [5]. In classical ensemble learning architecture, the data source provides data sets to the different models of a learning algorithm. Each model generates a particular result that will be considered a vote. Then the voting classifier combines all the votes from several models. The result from the voting classifier is finalized based on the majority of the vote. For example, the algorithm consists of 10 models and the models generate a total of 10 votes. 6 models produce 1 and 4 models produce 0 as the vote. In this scenario, the voting classifiers produce 1 as the final output. On the other hand, the proposed architecture is different from the classical ensemble learning architecture. It does not mention the voting classifier. It implements a probabilistic argumentation approach instead of the voting classifier. A data source transmits data sets to the different models of a decision tree algorithm. There are two models namely decision tree classifier and decision tree regressor. Both of the models produce results individually and transmit the results to a probabilistic argumentation framework for finalizing a reliable result.

The classical probabilistic argumentation refers to two aspects namely qualitative and quantitative. The qualitative aspects can be captured by the logic and the quantitative aspects can be accounted for by probabilistic measures. The proposed system uses the concepts of the classical probabilistic argumentation. It includes simple rules of a probabilistic argumentation framework in the settings of ELA for enhancing performance. That is why the implemented probability argumentation framework is a little different than the classical probabilities argumentation framework. The outcomes from two models of decision tree algorithms are considered as arguments. Each argument has qualitative aspects. The accuracy of outcome acts as the probability. The probability decides the attacking relation of the arguments. It also determines the status of an argument such as accepted or not accepted.

The previous studies discussed the experimental approaches of data classification in different scenarios. Some studies use only learning algorithms and some use neural network architecture to predict outcomes by classifying data [13] [3] [19] [10] [4]. The learning algorithms produce approximate outcomes in complex situations (e.g. data conflicting and overlapping). An accurate outcome is more desirable rather than an approximate outcome for practical usage (e.g. medical decision support system). Furthermore, the working procedure of a neural network is close to classical ensemble architecture [4]. The architecture uses learning algorithms for prediction and the predicted outcomes from different networks are merged to generate an outcome. In contrast, the proposed research uses an architecture which includes ELA and PAF. ELA uses learning algorithms for classifying data and prediction. PAF is an environment where multiple predicted outcomes can argue to finalize a outcome. PAF produces accurate outcome resolving data conflicting and overlapping in complex situations. The proposed architecture can produce a satisfactory outcome in normal and complex situations and it is suitable for the medical decision support system.

PAF helps to produce output that is relevant to the ground truth. The produced output from PAF is different than the output from DTC and DTR. The combined effort of learning algorithms and argumentation generates fruitful outcomes. For example, DTC and DTR produce C1 and C2 as outputs successively. C1 means the condition of patient health is normal and the patient does not require any diagnosis. On the other hand, C2 signifies the condition of patient health is bad and the patient requires some diagnoses. Then C1 and C2 argue through PAF and finally, C2 wins in the argument process. C2 is considered the more reliable output in this scenario. This implementation and evaluation of an ensemble learning approach based on an argumentation framework provide an answer to the research question. An argumentation approach successfully works as a decision fusion in an ensemble architecture. A probability argumentation framework produces outcomes by combining outcomes of DTC and DTR models. The more structured and large data set and extending the number of models will be increased the accuracy of the proposed system. PAF performs efficiently with an ensemble learning architecture and the contribution of PAF is more advantageous than voting classifiers.

The research presented by (Bar-Haim et al. [25]) represents the results on testing data set using coverage (e.g. the number of claims picked over a total of claims). The research implements a support vector machine (SVM) for classifying data. In the case of 10% coverage, the accuracy is 0.849. The accuracy is 0.740 for 60% coverage whereas it is 0.632 for 100% coverage. The research presented by (M Khalilia et al. [11]) presents a medical diagnosis system implementing a random forest algorithm for classifying data sets and predicting outcomes based on classified data. The overall accuracy of the system is 89.05% . The research presented by (Chi et al. [3]) represents the total accuracy based on different thresholds. The research implements a decision tree algorithm for classifying data sets. The total accuracies for thresholds 0.74, 0.85, and 0.95 are 0.813, 0.852, and 0.847 successively. The average accuracy of the proposed system with an argumentation approach is 88%. The overall accuracy of the proposed system is acceptable comparing to the accuracy of other researches.

The combined effort of the different models of decision tree algorithms and the probabilistic argumentation framework provides reliable outcomes because the outcomes are relevant to the ground truth. Moreover, the outcomes are produced by an argumentation approach that implements the probability function and an argumentation process to generate the outcomes instead of applying the max function (e.g. voting classifiers). The argumentation framework ensures reliable outcomes through an argumentation approach. The reliability of outcomes depends on the number of models of a learning algorithm. The number of arguments depends on the number of models. The increasing number of arguments and models will require to consider more sophisticated probabilistic argumentation semantics [4]. A more developed prototype of the system could further test the concept, increasing the number of models that could have an impact on outcomes.

8 Conclusion and Future Work

This study introduces a pipeline architecture that contains ensemble learning and argumentation approach. The combined work of ELA and PAF produces outcome and the accuracy of the outcome is satisfactory. Both approaches play a significant role to generate reliable outcomes. The different models of ELA produce different sets of outcomes. Then the produced different sets of outcomes act like different agents in the argumentation system. The different agents decide the final decision through a decision-making procedure that includes deliberation and negotiation. The final decision from the argumentation system is considered as a reliable outcome. The implemented architecture provides a satisfactory result regarding the specific problem, with an overall accuracy of 88%. Whereas the accuracy of ELA without an argumentation approach is 79%. Further studies are needed to increase accuracy and reduce time complexity, as well as increasing the models of ELA with different approaches that could have an impact on producing more and more reliable outcomes. Furthermore, the algorithm should be compared with other existing methods to explore the possibilities of this approach. A real-time application will be developed where the user can input data set or ultrasound images and the application produces the outputs based on input. Different kinds of argumentation semantics can be considered to deal with the probability argumentation process.

Bibliography

[1] Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[2] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] Chih-Lin Chi. Medical decision support systems based on machine learning. 2009.

[4] Oana Cocarascu, Elena Cabrio, Serena Villata, and Francesca Toni. A dataset independent set of baselines for relation prediction in argument mining. *arXiv preprint arXiv:2003.04970*, 2020.

[5] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002.

[6] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming. In *IJCAI*, volume 93, pages 852–857, 1993.

[7] Artur Garcez and Luis C Lamb. Reasoning about time and knowledge in neural symbolic learning systems. In *Advances in neural information processing systems*, pages 921–928, 2004.

[8] Simon C Harris. Travelling-waves for the fkpp equation via probabilistic arguments. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 129(3):503–517, 1999.

[9] Anthony Hunter and Matthias Thimm. Probabilistic reasoning with abstract argumentation frameworks. *Journal of Artificial Intelligence Research*, 59:565–611, 2017.

[10] Ilham Kadi and Ali Idri. A decision tree-based approach for cardiovascular dysautonomias diagnosis: A case study. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 816–823. IEEE, 2015.

[11] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):51, 2011.

[12] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[13] D Vijaya Kumar and VV Jaya Rama Krishniah. An automated framework for stroke and hemorrhage detection using decision tree classifier. In *2016 International Conference on Communication and Electronics Systems (ICCES)*, pages 1–6. IEEE, 2016.

[14] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[15] SOZAN SULAIMAN MAGHDID. *INTELLIGENT SYSTEM FOR IDENTIFICATION HEART DISEASES*. PhD thesis, NEAR EAST UNIVERSITY, 2019.

[16] Manzoor Ahmad Mir and Raid Saleem Albaradie. *Recent advances in stroke therapeutics*. Nova Science Publishers, Incorporated, 2014.

[17] Carey K Morewedge and Daniel Kahneman. Associative processes in intuitive judgment. *Trends in cognitive sciences*, 14(10):435–440, 2010.

[18] Wonhong Nam and Rajeev Alur. Learning-based symbolic assume-guarantee reasoning with automatic decomposition. In *International Symposium on Automated Technology for Verification and Analysis*, pages 170–185. Springer, 2006.

[19] Søren Holbech Nielsen and Simon Parsons. An application of formal argumentation: Fusing bayesian networks in multi-agent systems. *Artificial Intelligence*, 171(10-15):754–775, 2007.

[20] Juan Carlos Nieves and Roberto Confalonieri. A possibilistic argumentation decision making framework with default reasoning. *Fundamenta Informaticae*, 113(1):41–61, 2011.

[21] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.

[22] Kanak Saxena, Richa Sharma, et al. Efficient heart disease prediction system using decision tree. In *International Conference on Computing, Communication & Automation*, pages 72–77. IEEE, 2015.

[23] Anurag Srivastava, Eui-Hong Han, Vipin Kumar, and Vineet Singh. Parallel formulations of decision-tree classification algorithms. In *High Performance Data Mining*, pages 237–261. Springer, 1999.

[24] Shuangling Wang, Yilong Yin, Guibao Cao, Benzheng Wei, Yuanjie Zheng, and Gongping Yang. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. *Neurocomputing*, 149:708–717, 2015.

[25] Lauren K White, Kathryn A Degnan, Heather A Henderson, Koraly Pérez-Edgar, Olga L Walker, Tomer Shechner, Ellen Leibenluft, Yair Bar-Haim, Daniel S Pine, and Nathan A Fox. Developmental relations among behavioral inhibition, anxiety, and attention biases to threat and positive information. *Child Development*, 88(1):141–155, 2017.

[26] Andrew C Yao. Lower bounds by probabilistic arguments. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 420–428. IEEE, 1983.

[27] Yao Yu, Fu Zhong-liang, Zhao Xiang-hui, and Cheng Wen-fang. Combining classifier based on decision tree. In *2009 WASE International Conference on Information Engineering*, volume 2, pages 37–40. IEEE, 2009.

[28] Zhi-Hua Zhou. Ensemble learning. *Encyclopedia of biometrics*, 1:270–273, 2009.