Qubahan Academic Journal
گوڤارا قبهان يا ئەكاديمى
مجلة قبهان الأكاديمية
Doi: 10.48161/Issn.2709-8206

Machine Learning Classifiers Based Classification For IRIS Recognition

1st Bahzad Taha Chicho
Master Student
Duhok Polytechnic University
Duhok, Iraq
Bahzad.taha@dpu.edu.krd

2nd Adnan Mohsin Abdulazeez
President of
Duhok Polytechnic University
Duhok, Iraq
adnan.mohsin@dpu.edu.krd

3rd Diyar Qader Zeebaree
Research Center
Duhok Polytechnic University
Duhok, Iraq
Dqszeebaree@dpu.edu.krd

4rd Dilovan Assad Zebari
Research Center
Duhok Polytechnic University
Duhok, Iraq
Diloven.zebari@dpu.edu.krd

https://doi.org/10.48161/qaj.v1n2a48

***Abstract*—Classification is the most widely applied machine learning problem today, with implementations in face recognition, flower classification, clustering, and other fields. The goal of this paper is to organize and identify a set of data objects. The study employs K-nearest neighbors, decision tree (j48), and random forest algorithms, and then compares their performance using the IRIS dataset. The results of the comparison analysis showed that the K-nearest neighbors outperformed the other classifiers. Also, the random forest classifier worked better than the decision tree (j48). Finally, the best result obtained by this study is 100% and there is no error rate for the classifier that was obtained.**

Keywords—Data Mining, Classification, Decision Tree, Random Forest, K-nearest neighbors

I. Introduction

Nowadays, the data online is massive, and it is growing on a daily basis. It is essential to handle such vast amounts of data and to view the most relevant queries on the user's computer. Since manually analyzing and retrieving relevant data from vast databases is impossible, automatic extraction tools are needed, which enable user-queried data to be retrieved from billions of sites on the internet and relevant knowledge to be discovered. Search engines such as Yahoo, Bing, MSN, and Google are commonly used by users to obtain data from the World Wide Web [1], [2]. Data mining is also used to explore and derive information from data warehouses.

Data mining is a method of processing user data and extracting data from vast data warehouses that employ a variety of trends, intelligent processes, algorithms, and software. This approach will assist companies in evaluating results, forecasting potential patterns, and predicting user behavior. For relevant data extraction, data mining involves four techniques phases [3], [4], [5]. A data base is a set of information from different sources, a vast database that can include issue definitions. Data discovery is the method of collecting valuable knowledge from vast volumes of unfamiliar data [6]. The third stage is modeling, which entails creating and evaluating various templates. Finally, in the final phase of data mining techniques, validated models are implemented [7]. Data mining methods may be used by businesses to turn raw data into useful facts. By understanding all about consumer actions, it will also assist companies in enhancing their communication campaigns and growing revenues [8], [9]. Moreover, this data should be properly classified to benefit from its great use.

Classification tries to predict the target category with the highest accuracy. The classification algorithm establishes a connection through the input and output attributes in order to build a model [10], [11]. The volume of data collected in data mining environments is massive. Using the decision tree method is optimal if the data set is properly classified and contains the fewest number of nodes [12], [13].

A Decision Tree (DT) is a tree-based strategy in which every direction between the root and the leaf node is represented by a data separating series before a Boolean outcome is obtained [14], [15]. It is a hierarchical exemplification of nodes and links in information relationships. Nodes reflect uses as ties are used to distinguish [16]. DT is a form of ML algorithm that is applicable to both classification and regression. It typically makes use of the shape of a binary tree, with each node making a decision by comparing a function to a threshold and dividing the decision route there. Depending on whether the task is grouping or regression, leaf nodes include choices, actual values, or class names [17], [18]. Random Forest (RF) utilizes an ensemble of trees to create trees at random using the training input vector to estimate the output vector, equivalent to producing a random range of weights that is unchanged by previous weight sequences [19]. The best tree is then voted in, and the procedure is replicated a certain number of times, with the best tree being chosen as the corresponding classifier [20]. The K-Nearest Neighbors (K-NNs) classifier, also known as the Nearest Neighbor Classifier, is a kind of supervised ML method that is used to classify or predict data. K-NN is incredibly easy to set up and use, yet it excels at specific grouping tasks like economic forecasting [21], [22]. Since it is a non-parametric approach, it does not have a particular training phase. Instead of classifying a question data point, it observes all of the data. K-NN can no longer make any assumptions regarding the underlying results. This property corresponds to the underlying trend in the vast majority of real-world datasets [23], [24]. The aim of this study is to evaluate the efficiency of the used methods that are based on classification. Besides, the researchers have highlighted the most widely employed techniques as well as the strategies with the best precision.

The remainder of the paper is structured as follows: Section II includes a related work on the used classification algorithms; Section III contains supplementary details regarding the IRIS datasets; Section IV explains the three approaches used in this study; Section V illustrates the experimental results and discussion; Section VI comparative studies on the mentioned techniques; and Section VII concludes the research work.

II. RELATED WORK

The term data mining is a process of assigning individual objects in a database to one or set of categories or groups. In the phase of classification, the aim is to correctly classify the target class for each instance. This portion offers a survey of the most current and useful approaches to classification in different fields of ML that have been established by researchers in the last two years. Also, it only focuses on decision trees, random forests, and k-Nearest Neighbors as classifiers.

Lakhdoura and Elayachi [25] compared the performance of two classifiers methods: J48 (c4.5) and RF on the IRIS features, and the test was executed by the WEKA 3.9. Therefore, the IRIS plant dataset, one of the most common databases for classification issues, is gained from the ML library at the University of California, Irvine (UCI). In addition, the investigators compared the results of both classifiers on various efficacy assessment measures. The findings showed that the J48 classifier outperforms the Random Forest (RF) classifier for IRIS variety prediction using various metrics such as classification precision, mean absolute error, and time to construct the technique. The J48 classifier has an accuracy of 95.83%, while the Random Forest has an accuracy of 95.55%.

Mijwil and Abttan [26] proposed using a C4.5 decision tree to reduce the effects of overfitting. The datasets used were IRIS, Car Assessment, Bottle, and WINE, both of which may be included in the UCI ML library. The trouble with this classifier is that it has so many nodes and divisions, which contributes to overfitting. This overfitting has the potential to sabotage the classification mechanism. The experimental findings showed that the genetic algorithm was efficient in pruning the impact of overfitting on the four datasets and maximizing the trust Confidence Factor (CF) of the C4.5 decision tree, with an accuracy of about 92%.

Rana et al. [27] performed the comparison between Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) with Linear Regression (LR) and Random Forest (RF) for IRIS Flower Classification. The suggested distinction between the results of both machine learning and the dimensionality reduction processes. According to the findings, both approaches provide decent classification performance, though the accuracy varies depending on the number of principal components chosen. LDA, on the other hand, outperforms PCA for a defined collection of principal components. The analysis also showed that when the percentage of training data improves, so does the degree of precision. LR and RF were used to classify the data. Both the RF and PCA approaches behave similarly to PCA and LDA. In comparison to the 86% of results provided by PCA, the LDA performs much better, providing 100% accuracy.

Gong et al. [28] presented a new evidential clustering algorithm centered on the discovery of "cumulative belief peaks" and the application of the irrefutable K-NN principle. This method's basic assumption is that a cluster center in its neighborhood has the greatest accumulated probability of becoming a cluster center, and that its neighborhood is relatively big. Iris, Pima, Seeds, Waveform, WDBC, Wine, and Pen-based datasets were all included in the analysis. In the context of belief functions, a new notion of accumulated belief is proposed to quantify such cumulative probability. The scale of the comparatively wide neighborhood is calculated by optimizing an objective function. The cluster centers are then immediately detected as the objects with the highest collective confidence among their own neighborhood of this magnitude. Finally, a creedal partition is formed using the evidential K-NN base and the constant cluster core. Experiment results show that when working with datasets with a limited number of data items and measurements, in a reasonable amount of time, the suggested evidence gathering method will easily classify cluster nodes and reveal data structure in the form of doctrinal sections. When using seeds as a dataset, the best accuracy is obtained, which is 95.26%.

Shukla et al. [29] focused around how machine learning algorithms can automatically identify the flower class with a high degree of precision rather than roughly. They used the IRIS dataset, and it is divided into three groups, each with 50 instances. The Iris dataset utilizes deep learning to classify the subclasses of Iris flower. Segmentation, function extraction, and classification are the three steps of this method's implementation. To identify the flower class, Neural Networks (NN), Logistic Regression (LR), Support Vector Machine (SVM), K-NN are utilized. The results showed that the accuracy achieved by each algorithm was as follows: Both NN, LR, and K-NN have an equal precision of 96.67% while a SVM has higher precision than all of which is 98%.

Sugiharti and Putra [30] analyzed the system of Two-Dimensional Principal Component Analysis (2DPCA) paired with K-NN is used for facial image recognition. The study employs the 2DPCA system for extraction of features and the K-NN classification techniques for data classification, resulting in the required accuracy score. The image archive from the UCI repository is used by the test participants, and it includes 190 black and white facial images of individuals in different positions (straight, left, correct, up), expressions (neutral, positive, sad, angry), and sizes. The results showed that the output review of Facial Image Recognition was focused on the 2DPCA process, which was combined with K-NN. Moreover, the accuracy of 2DPCA is equal to 94.74%, while the K-NN obtained the best accuracy which is 97.37%, when the values of k = 1 and k = 2, with the smallest recognition errors.

Quist et al. [31] presented a permutation-based model for RF approaches that allows for impartial mixed-type data incorporation while still determining relative function significance. The system is adaptable, modular, and can be used across a wide range of studies. They chose breast cancer as a dataset since the causes of certain diseases are complex and include more than one biological agent. The approach's output was measured using modeling experiments and machine learning datasets. There was very little multicollinearity and very little over fitting in the results. The permutation-based approach was extended to multidimensional high-dimensional different datasets from two separate breast cancer cohorts to further evaluate accuracy. The concordance in relative feature value between the cohorts, as well as accuracy in clustering profiles, illustrated the reproducibility and robustness of our methodology. One of the newly identified clusters has been demonstrated to be predictive of clinical results during standard-of-care adjuvant chemotherapy, outperforming conventional intrinsic molecular breast cancer classifications. Also, 95% of the cases in the International Cancer Genome Consortium (ICGC) Cluster i5 were Estrogen Receptor-positive (ER-positive).

KADHM et al. [32] suggested a Palmprint Recognition System (PRS) that is both precise and effective. The framework used path, Local Binary Pattern (LBP) features, DT (C5.0), and K-NN to isolate and classify features. The College of Engineering Pune (COEP) and the Chinese Academy of Sciences provided palmprint image datasets for the method (CASIA). The method became more efficient and reliable after going through all of the steps, which included preprocessing, segmentation, feature extraction, and classification. The findings of the comparison show that the device outperforms current processes and procedures. The method can also work accurately in an online recognition manner by using a scanning device to interpret the palmprint images directly, thanks to the efficient recognition stages, especially the classification stage. The PRS had a strong identification accuracy of 99.7% and a low error matching rate of 0.009%. Also, the accuracy of LBP, DT (C5.0), and K-NN are equal to 92%, 70.25%, and 95%, respectively.

Ogundokun et al. [33] investigated the diagnosis of long-sightedness employing three techniques, namely NN, DT, and Back Propagation, resulting in the creation of an Expert System. Furthermore, the information area was extracted from detailed discussions with specialists in the area of eye examination (ophthalmologists) as well as various studies of the literature. The specialist framework was built from the ground up using the C# programming language and MySQL as the database. The NN was trained using back propagation and DT algorithms. According to the signs of the patient, a DT was used to identify and categorize the illness using an information extraction rule. The designed system's outcome demonstrated how the illness was detected in order to eliminate the neural network's impenetrability. Also, they showed that the hybridization of the three algorithms made the system model accurate and efficient, and eventually, the strategy was validated after implementation.

Sarpatwar et al. [34] offered an end-to-end method to support privacy-enhanced decision tree classification using an open-source Homomorphic Encryption Library (HELib). They demonstrated the classification use case for decision trees with the iris dataset (150 samples, 4 functions, and 3 classes). The comparator and other associated processing in the first stage enable the function values to be within a certain range. Use a number of options to create a decision node, in addition to the ignorant accounts and the argmax feature in g Fully Homomorphic Encryption (FHE). The findings revealed that a highly stable and trustworthy decision tree service can be implemented, and the achieved precision was 98%, meaning that the private solution suited the non-private variant nearly exactly.

III. DATASET

In this article, three data mining algorithms on classifications are applied to the IRIS dataset from the UCI ML library. There are five characteristics in the data collection, each of which corresponds to a different iris flower species. Class (Species), Petal Length, Petal Width, Sepal Width, and Sepal Length are the characteristics [35]. There were 50 samples of each genus, totaling 150 examples. For the four non-species defining characteristics, this data form is broken down numerically in (cm) volume. Furthermore, it offers a clear and easy-to-manage presentation [36]. Data mining and deep learning have been extensively applied to clustering for several years for the iris dataset. It was postulated by the British statistician and evolutionary biologist Ronald Fisher in his publication, "On the Analysis of Covariance of Taxonomic Studies," in which he argued that multiple measure testing ought to be preferred to one over one measure for character classification. The IRIS flower types are shown in "Fig. 1", while sample instances of the IRIS dataset are illustrated in "Table 1".

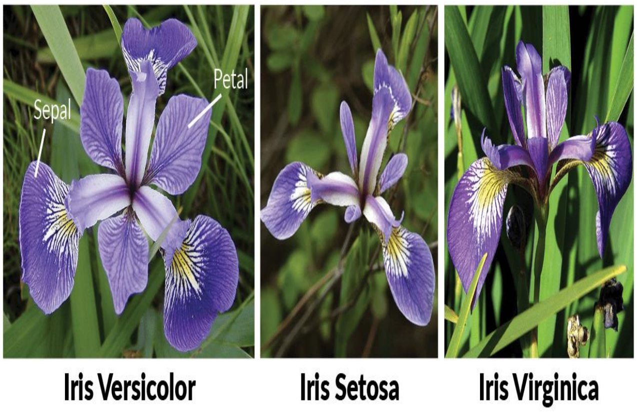

Fig. 1: IRIS flower types

TABLE 1: SAMPLE INSTANCES FROM IRIS DATASET

	Sepal length	Sepal width	petal length	petal width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

IV. METHODOLOGY

Classification is a data mining strategy for categorizing data instances into one of a few classes. Machine learning classification algorithms are made up of many algorithms that have been designed to outperform one another [37], [38]. They all use statistical methods such as decision trees, linear programming, support machine vectors, and neural networks, among others. To make a guess, these methods examine the available data in a variety of ways [39].

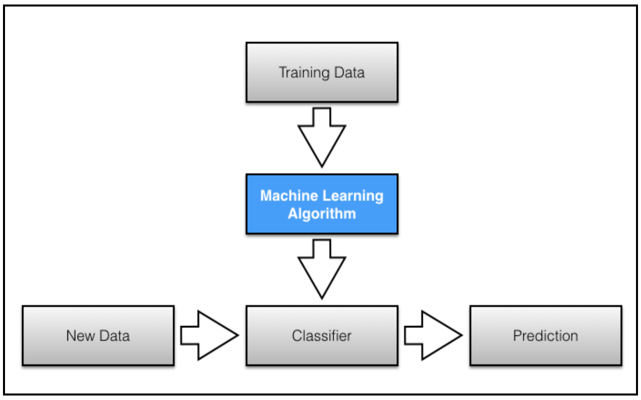

Fig. 2: Simplified diagram of the procedures for building the general pattern classification model [40].

This work focused on decision tree, random forest, and k-Nearest neighborhoods algorithms in general, and they are implemented by the data mining tool known as Weka. “Fig. 2” depicts a simplistic illustration of the procedures for building the general pattern classification technique.

A. *Decision Tree Classifier*

One of the techniques widely used in data mining is the systems that create classifiers [41]. DT is a text and data mining classification algorithm that was used previously. Decision Tree classifiers (DTCs) have been shown to be effective in a variety of classification applications. A hierarchical decomposition of the data space is the framework of this methodology. D. Morgan first suggested, and J.R. Quinlan established the DT as a classification task. The basic concept is to create a tree with classified data points dependent on attributes, but the main problem of a DT is deciding which attributes or features should be at the parent level and which should be at the child level. De Mántaras suggested statistical modeling for feature selection in trees as a solution to this issue [42]. The structure of DT is illustrated in “Fig. 3”.

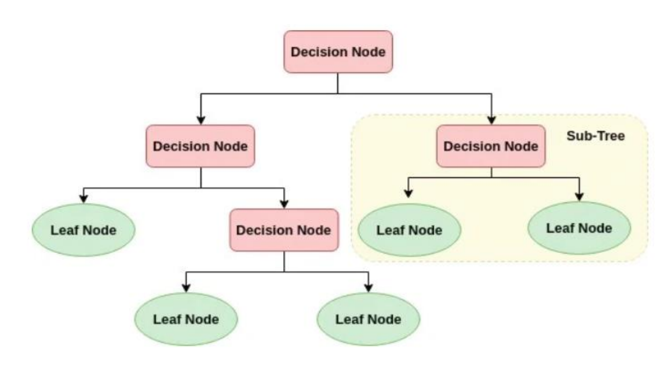

Fig. 3: Structure of DT

There are several kinds of DT techniques, that contain Iterative Dichotomies 3 (ID3), Successor of ID3 (C4.5), Classification and Regression Tree (CART) [44], CHi-squared Automatic Interaction Detector (CHAID) [45], Multivariate Adaptive Regression Splines (MARS) [46], Generalized, Unbiased, Interaction Detection and Estimation (GUIDE), Conditional Inference Trees (CTREE) [43]. The DT method is a supervised linear classifier whose main goal is to provide a training scheme that can be employed to infer judgment principle from a dataset in order to predict the class or value of target variables [44]. The DT algorithm can be used to overcome regression and classification issues, but it has a range of benefits and drawbacks, which are described in “TABLE 2”.

TABLE 2: DT BENEFITS AND DRAWBACKS [45]

Benefits	Drawbacks
1) Easy to understand. 2) Easily converted into a series of production principles. 3) May distinguish both categorical and numerical results, but only categorical attributes can be produced. 4) No a priori hypotheses are considered when evaluating the quality of the findings.	1) The desired decision-making process may be thwarted, resulting in erroneous judgments. 2) The DT has a number of layers, which makes it fascinating. 3) The DT's estimation difficulty may increase when further training samples are added.

The split of a DT is based on the computation of both entropy and knowledge gain. The impurity or randomness of a dataset is calculated using entropy [46]. The entropy value is always between 0 and 1. Its meaning is higher when it equals 0, and it is bad when it equals 0, i.e., the closest it is to 0, the better. As shown in "Fig. 4." The entropy of the grouping of set S with respect to c states if the objective is G with separate attribute values. As shown in “ Equation (1)”.

$$Entropy(S) = \sum_{i=1}^{c} P_i \log 2^{P_i} \qquad (1)$$

Where P_i is the ratio of the subset's sample number to the sum of the i-th attribute.

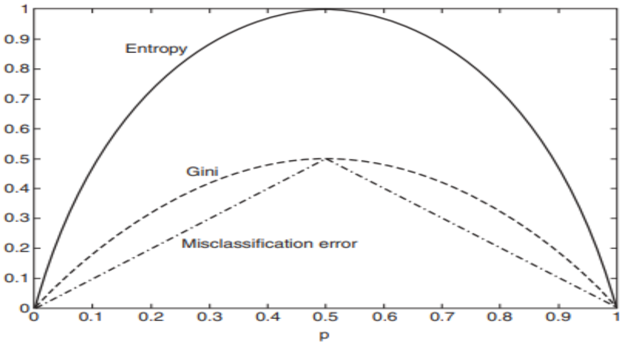

Fig. 4: The entropy value is shown [47]

Mutual information is another term for information gain, which is a metric used for segmentation. This tells you how much you do about the meaning of a random variable [48]. It's the inverse of entropy, and the higher the frequency, the greater. On the basis of the concept of entropy, the data $Gain(S, A)$ is specified as follows: “Equation (2)” shows this [49], [50].

$$Gain(S, A) = \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (2)$$

Where V(A) represents the spectrum of attribute A, and SV represents a subset of set S equal to the attribute value of attribute V.

There are several DT approaches, such as ensemble processes, that are used to create multiple DTs [51]:

- **Bagging:** Bagging is a strategy for constructing a large number of DTs by resampling guided information with alternates and determining the tree for a consent measurement.
- **Random forest:** To improve the classification rate, this sort of classifier chooses the numerals of DTs at random.
- **Boosted trees**: Boosted trees are a kind of tree that can be used to reflect classification and regression problems.
- **Rotation forest:** In this ensemble approach, each DT is first subjected to Principal Component Analysis (PCA).

B. *Random Forests Classifier*

Random Forest is a classification method consisting of a set of tree-structured algorithms with identically distributed separate random values, each tree casting a single vote for the most popular class at input x.

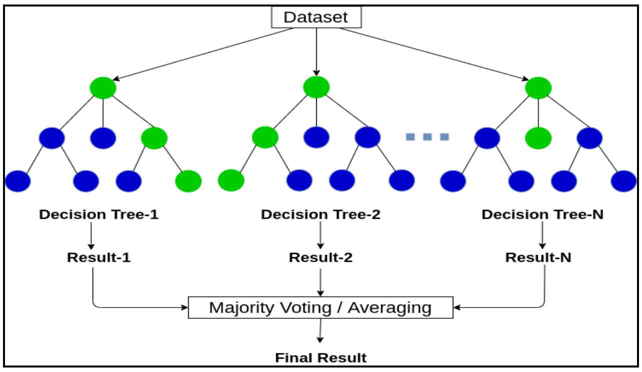

Fig. 5: The RF structure

A tree is developed utilizing the training test [52], which is able to generate vector that is independent of previous random vectors of the same distribution. In terms of two parameters: exactitude and interdependence of individual classifiers, an upper bound is extracted for RF to get the generalization error [53]. The RF structure is shown in “Fig. 5”. In 2001, Breman presented the learning model integrated with the primary classifier DT being RF. It uses the bootstrap approach to collect several subsets of samples, then generates a DT from every subclass of items, and then merges those DTs into an RF. When the classification tests are reached, the classification's final result is determined by a ballot on the DT. Scholars usually start by raising the accuracy of the classifier and then decreasing the interaction among classification models [54]. The final reduction of the classification effect is achieved using the RF method in the classifier, where the outcomes of each base classifier's classification have a similar error distribution. Takes the properties of the test and predicts the result based on the rules of each randomly generated DT and stores the predicted result (target). Calculate the number of votes for each projected target [55]. Consider the expected high-voted target to be the RF algorithm's final prediction. “Fig. 6” showed the RF's training phase.

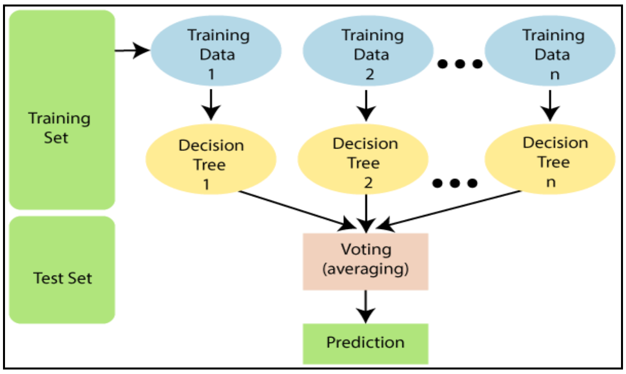

Fig. 6: Random Forest training flowchart

Random forests are a method for classification, regression and other functions, often called random decision forests, which work by building a large amount of DT during training and then producing the class that is the mode of classification (speciation) or average estimate (regression) of the different trees. Random decision forests compensate for

DTs' proclivity to overfit their training collection [56]. RFs outperform DT in general, but they have benefits and drawbacks, as seen in “TABLE 3”.

TABLE 3: RF BENEFITS AND DRAWBACKS [57]

Benefits	Drawbacks
1) There is better precision. 2) Capable of interacting with huge datasets. 3) It efficiently and easily handles thousands of input variables. 4) Provides detail on critical factors that are not used in the Classifying. 5) Handles missing data while preserving precision. 6) Prototypes are employed to include details or Meta data on the interaction among multiple variables.	1) One of the more common issues discovered is Oversize the selection of a single feature, particularly with regression problems. 2) RF struggle with multiple values and multiple values characteristics, in several dimensions. 3) They favor categorical variables of several levels.

Another excellent aspect of the RF algorithm is how simple it is to assess the relative value of each feature in the forecast. Sklearn has a fantastic method for measuring the value of a function by looking at how often the tree nodes that use it decrease impurity in the whole forest. After preparation, it calculates this score for each element and scales the scores such that the total importance equals one. You will determine the functionality to remove based on their value since they don't add sufficiently (or even none at all) to the prediction method. This is significant because, in machine learning, the more capabilities you have, the more likely your model is to suffer from overfitting, and vice versa [58].

C. K- Nearest Neighbors Classifier

K-NN is a non-parametric of both classification and regression approach. K-NN is one of the methods utilized in the guided learning process. The fundamental idea behind this approach is to identify data by computing the k nearest neighbors to a data point. In other terms, calculate the gap between the test data and the feedback and make the appropriate prediction. Furthermore, the point is the most common class allocated to those k neighbors [59]. As shown in “Fig. 7”.

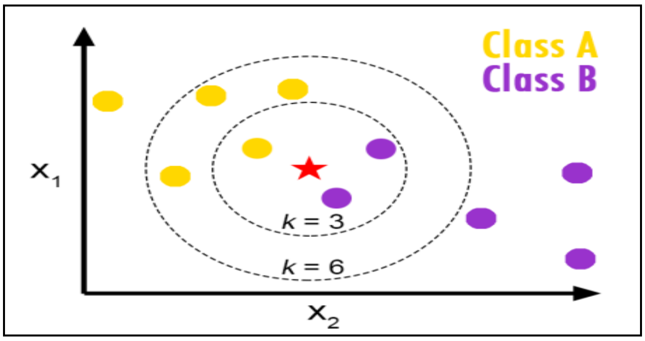

Fig. 7: K- Nearest Neighbors Classifier

There are some different metrics for measuring the disparity between the two samples. The Euclidean distance, which defines the element-wise distance between corresponding elements of two objects to be compared, is one of the extreme commonly utilized metrics. The K-NN rule classifies each unknown instance in the training set based on a plurality vote from its closest K-NN neighbors [60]. Its efficiency is often heavily influenced by the distance metric used to describe the closest neighbors. In the absence of prior knowledge, most K-NN classifiers use basic Euclidean metrics to measure the gap between examples described as vector inputs [61], [62]. Other proposed distance estimation formulas involve Xing distance measurements, in addition to the traditional distance approaches such as Minkowski and Chebyshev. The Euclidean distance is determined using the formula shown below in “ Equation (3)”.

$$d(x_i\ ,\ x_j) = \sqrt{\sum_{r=1}^{n} w_r\,(\ a_r(x_i) - a_r(x_j))^2} \qquad (3)$$

When an example is given as a vector $x = (\ a1, a2, a3, \dots\dots, an)$, n is the amount of example attributes in the input vector's dimensionality. a_n is an example r-th attribute, w_r is the weight of the r-th attribute, r ranges from 1 to n, and the smaller the $d(x_i\ , x_j)$, Which two cases is more important. The class mark allocated to the test example must be determined by a plurality vote of its closest k neighbors.

$$y(d_i) = argmax \sum_{x_j \in kNN} y(\,x_j\ , c_k) \qquad (4)$$

Where d_i is an indicator of a test, x_j is one of the training set's closest neighbors, and $y(x_{ij}\ , C_k)$ indicates if x_j belongs to class C_k . According to “ Equation (4)”, the indicator is a class with the bulk of its representatives in the closest k neighbors. For example, if the 5-nearest neighbor algorithm is converted into a classifier, three of the five closest neighbors in the case belong to category One, while the other two belong to category Two [63]. We should conclude that the test case is from class one. If the class mark of a sample is achieved solely by defining its nearest neighbors (NN), the closest neighbor’s algorithm is used [64]. K-NN claims that the class's conditional probabilities are globally stable and that big dimensions profit from bias. K-NN is an exceptionally scalable classification scheme that does not involve any pre-processing of training results [65]. It is not a good idea to use the same K-NN algorithm to choose the class labels for all test examples by choosing the same number of near neighbors. The improved k-NN algorithm should then concentrate on determining the necessary k, or the number of its nearest neighbors, in order to decide the possible class mark for each test example. “Fig. 8” showed an overview on the method in general.

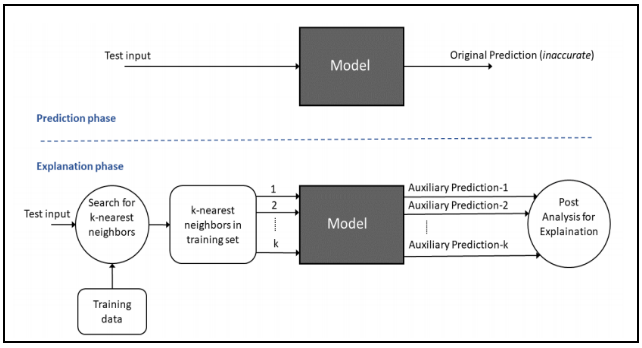

Fig. 8: Overview of the K-NN

The following measures are followed by the K-NN method: To begin, enter the data set and divide it into training and testing sets. Then, pick an instance from the test sets and determine its distance from the training collection [66]. After that, list the distances in ascending order. Finally, the instance's class is the most unique class of the first three teaching instances (k=3).

The K-NN technique has been implemented in many aspects of the classification, including findings that are both optimistic and unfavorable. Even though the K-Nearest Neighbor approach is an impressive classifier, it has its benefits and drawbacks like most classifiers [67], as seen in "TABLE 4".

TABLE 4: K-NN BENEFITS AND DRAWBACKS

Benefits	Drawbacks
1) The preparation phase is fast and free of charge. 2) Simplicity of use and speed of execution. 3) It can deal with noisy data. 4) Even compelling if the training data is massive. 5) The algorithm successfully computes several class labels for an unknown case.	1) It is very expensive computationally. 2) It is highly susceptible to characteristics that are irrelevant. 3) It is a slow algorithm that takes longer to implement. 4) A significant amount of memory is needed to store all of the training examples. 5) The expected cost is large since a device is expected to move the distance between each instance and all training exercises, and the value of K must be calculated.

The feature extraction in the K-NN, When an algorithm's input data is too big to process and is accused of being repetitive (for example, the same calculation in feet and meters), the data is converted into a reduced representation collection of features. Feature extraction is the process of transforming input data into a series of functions [68]. It is assumed that the features collection would retrieve the necessary details from the input data in order to execute the desired role utilizing this reduced representation instead of the full size input if the features collected are carefully selected. Until applying the k-NN algorithm to the transformed data in feature space, feature extraction is done on the raw data. Function extraction and dimension reduction pre-processing measures are included in a standard computer vision computing pipeline for face recognition using k-NN:

1. Face recognition by Haar, which is a wavelet family or base is composed of a series of rescaled "square-shaped" functions [69].
2. Study of mean-shift monitoring.
3. Following a PCA or Fisher LDA projection into feature space, k-NN classification is used.

D. WEKA Tool

Orderly to perform experiments and applications, WEKA was employed as the data mining tool. WEKA (Waikato Environment for Knowledge Analysis) is a Java-based data mining tool built at Waikato University. In the field of bioinformatics, WEKA is an excellent data mining method that helps users to identify the precision of datasets by contrasting various algorithmic approaches. Researchers have used the Explorer, Experimenter, Workbench, and Knowledge Flow interfaces in WEKA [70].

The WEKA suite provides data mining and predictive analytics visualization tools and techniques, as well as immersive user interfaces for easy access to this application. It includes a plethora of algorithms for data processing and deep learning [71]. WEKA is open source and free to use. It is also network agnostic. WEKA facilitates many capabilities for its users, and among its main capabilities are preprocess, classify, cluster, associate, select attributes, and visualize [72], and they are illustrated as follows:

- **Preprocess**: WEKA supports a native file format (ARFF) as well as a variety of database connectivity by JDBC and other formats (for example, CSV and Matlab ASCII files). Data may also be handled in a number of forms (over 75), from excluding individual attributes to doing more complex processes like PCA.
- **Classify**: WEKA's more than 100 grouping strategies are one of its selling points. Classifiers are classified as "Bayesian" learners (Naive Bayes, Bayesian networks, etc.), "Lazy" learners (nearest neighbor and variants), rule-based (decision tables, OneR, RIPPER), tree learners (C4.5, Naive Bayes trees, M5), function-based learners (linear regression, SVMs, Gaussian processes), and miscellaneous. WEKA also includes meta-classifiers such as bagging, boosting, and piling, as well as various instance classifiers and interfaces for Groovy and Jython classifiers.
- **Cluster**: Several clustering systems, EM-based mixture structures, k-means, and a variety of hierarchical clustering techniques are among them, enable unsupervised learning. Most of the classic algorithms are included, despite the fact that there are not as many as there are for sorting.
- **Select attributes**: For classification outcomes, the characteristics used are crucial. There are a variety of classification criteria and search tools to choose from.
- **Visualize**: Plotting attribute values on a graph allows for visual analysis of results against the class or against other attribute values. To detect outliers and analyze classifier characteristics and judgment limits, classifier performance can be compared to training results. There are advanced visualization applications for particular approaches, such as a tree viewer for any system that generates classification trees, a Bayes network viewer with automated layout, and a dendrogram viewer for hierarchical clustering [73].

V. EXPERIMENTAL RESULTS AND DISCUSSION

The results of applying the DT, RF, and K-NN algorithms to the IRIS datasets, which are explained in detail in the ”TABLE 1”, were discovered using a web-based framework developed with Weka. The simulation was performed on a laptop with a Core-I3 processor operating at 2.20 GHz and 4 GB of RAM.

A. *Performance of the Decision Tree (j48) classifier*

First, the author applies the decision tree and employs the J48 kind, which is an algorithm for generating a decision tree using C4.5 (an extension of ID3). It may also be referred to as a mathematical classifier. Several tests are carried out in order to test the chosen method utilizing the generated dataset. The test mode for evaluation is k-fold Cross-Validation (k-fold CV). The k-fold CV is an experimental research technique in which the database is randomly divided into k disjoints entity fragments, the data mining classifier is run using k-1 items, and the remaining block is used to evaluate the method's accuracy. This procedure is repeated k times total. Finally, the reported measurements are averaged. It is normal to use k=10 or some other size based on the initial dataset size. After the experiments are completed with the chosen dataset, the findings are gathered, and an average comparison is performed utilizing the classification and testing modes that are accessible. There are many factors that affect the results. For starters, the higher the value of cross-validation (k-fold cv), the greater its accuracy, that is, its positive relationship with the other; for example, when the cross-verification process is equal to 10, the accuracy is equal to 94%, while mutual verification equals 50, the accuracy is equal to 96%. The precision of the percentage of split improves as the separation of the tree is increased. When the percentage amount is equivalent to 50% of training, the accuracy obtained is 94.66%, while the attained accuracy is equal to 95.55%, which is 70%. The confidence factor, which reflects a threshold of permitted inherent error in data when pruning the decision tree, is another environment that affects the pruning method. It affects the size of the tree and the number of leaves when the value of pruning changes, affecting the size of the tree and its leaves. Besides, the amount of time required to construct the model is equal to (0 seconds) in all the above cases. This means that these factors are not affected at implementation time to build the model. Furthermore, accuracy by class was detailed in “TABLE 6” when cross validation, and confidence factor were set to 10, and 0.25, respectively. Also, the performance metric of DT (j48) is illustrated in “TABLE 5”. The weighted average accuracy is 98%, and the amount of time required to construct the model is 0 seconds.

TABLE 5: PERFORMANCE METRIC OF DT (J48)

Attribute	Value
Correctly Classified Instances	147 (98 %)
Incorrectly Classified Instances	3 (2%)
Kappa statistic	0.97
Mean absolute error	0.0133
Root mean squared error	0.1155
Relative absolute error	3%
Root relative squared error	24.4949 %
Total Number of Instances	150

TABLE 6: OBTAINED RESULTS BASED DT (J48) BY CLASS

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.980	0.000	1.000	0.980	0.990	0.985	0.990	0.987	Iris-setosa
	0.960	0.010	0.980	0.960	0.970	0.955	0.975	0.954	Iris-versicolor
	1.000	0.020	0.962	1.000	0.980	0.971	0.990	0.962	Iris-virginica
Weighted Avg.	0.980	0.010	0.980	0.980	0.980	0.970	0.985	0.967	

B. *Performance of the Random Forest classifier*

In classification, Cross validation tests of various kinds are used. The researcher evaluated the Random Forest's success using a 10-fold CV test in this scenario. The RF is checked on one fold, while the other folds are used for training. The entire test is replicated five times, with the findings eventually being combined. In all cases of implementation, the results do not change. Only the time spent in the process changes. Where the first implementation process takes longer than in other cases, and the average absorption in all cases is equal to 0.028 seconds. The accuracy in all cases does not change and is equal to 99.33%. Cross validation is impacted by the time taken for building the model and the accuracy of the process. The execution of the process is repeated more than five times and each time the cross-verification value changes, its value ranges between 10 and 60, so the time elapsed and the accuracy of the process varies each time. The time taken to build the model reaches 0.032 seconds as an average and the accuracy reaches 99.73% as an average. Also, the percentage split affects the time it takes to build the model and changes its value from 40 to 90, as the time taken varies and ranges from 0.02 seconds to 0.1 seconds. However, the accuracy in all cases tested in the process remains unchanged and remains the same. In addition, when cross validation is set to 10, accuracy by class is detailed in "TABLE 8". In contrast, "TABLE 7" shows the RF efficiency metric. The weighted average accuracy is 99.33%, and it takes 0.02 seconds to build a model.

TABLE 7: PERFORMANCE METRIC OF RF

Attribute	Value
Correctly Classified Instances	149 (99.33%)
Incorrectly Classified Instances	1 (0.6667 %)
Kappa statistic	0.99
Mean absolute error	0.0143
Root mean squared error	0.0691
Relative absolute error	3.22 %
Root relative squared error	14.6479 %
Total Number of Instances	150

TABLE 8: OBTAINED RESULTS BASED RF BY CLASS

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-setosa
	0.980	0.000	1.000	0.980	0.990	0.985	1.000	0.999	Iris-versicolor
	1.000	0.010	0.980	1.000	0.990	0.985	1.000	1.000	Iris-virginica
Weighted Avg.	0.993	0.003	0.993	0.993	0.993	0.990	1.000	1.000	

C. Performance of the K- Nearest Neighbors classifier

On the IRIS databases, the analysis used the 10-fold cross-validation technique. This study divided the entire dataset into ten subsets at random, and then chose one subset for research and the other nine subsets for preparation. To prevent prejudice during dataset partitioning for cross-validation, the researchers replicated the procedure ten times and nearest neighbors equal to one. The final outcome was calculated by comparing the outcomes of all tests; in other words, the investigator replicated the experiments 10 times on each dataset and took the average score as the recorded results. For the classification task, the analysis used classification accuracy as the criteria. The higher the algorithm's precision, the better the classification's efficiency. Furthermore, the precision obtained after repeating the test ten times does not change and stays stable at 100% in all situations. Additionally, repetition has no effect on the time it takes to build the model, where time elapsed in all cases is equal to 0 second. Also, the percentage split affects the accuracy of the operation when it changes its value from 30 to 90. By executing the operation, it showed that the greater the split percentage improved the accuracy, and the accuracy value increased from 98% to 100%. However, the time it takes to build the model does not affect when the percentage split changes. The researcher then increased the value of nearest neighbor (s) for classification from 1 to 5, and it was discovered that the accuracy improved and had equal value during implementation. In addition, the amount of time required to construct the model does not affect and remains the same when the nearest neighbor value changes. In contrast, each of the, Mean Absolute Error, Root Mean Squared Error, Relative Error, Root Relative Squared Error and Root, Relative Error were affected by changing the nearest neighbor (s) value, and all them decreased when increased nearest neighbor. Moreover, when cross validation is set to 10 and nearest neighbor to 1, accuracy by class is detailed in "TABLE 10". In contrast, "TABLE 9" shows the K-NN efficiency metric. The weighted average accuracy is 100%, and it takes 0 second to build a model.

TABLE 9: PERFORMANCE METRIC OF K-NN

Attribute	Value
Correctly Classified Instances	100%
Incorrectly Classified Instances	0%
Kappa statistic	1
Mean absolute error	0.0097
Root mean squared error	0.0102
Relative absolute error	2.1739 %
Root relative squared error	2.1739 %
Total Number of Instances	150

TABLE 10: OBTAINED RESULTS OF K-NN BY CLASS

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-setosa
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-versicolor
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-virginica
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	

The researchers evaluated numerous classifiers on IRIS datasets, including DT (j48), RF, and K-NN. The results indicate that due to their differences in functionality, the classifiers offer different resolutions on different datasets. "Tab 11" explained the accuracy, error rate, and time to construct the model. In comparison to Random forests and J48, the K-Nearest Neighbor algorithm performs exceptionally well. The ultimate conclusion of this paper is that K-Nearest Neighbor has the maximum accuracy, minimum error rate, and takes less time to build the model than other classifiers. Also, the Random forest is graded second in comparison to the DT in terms of consistency, which is ranked last, even though the duration of the model's construction is less than that of the random forest, as seen in the table below, and in the “chart 1”. The accuracy was determined using the " Equation (5, 6)" as seen below.

$$Accuracy = \frac{TP+TN}{P+N} \qquad (5)$$

$$\text{Error Rate} = \frac{FP+FN}{P+N} \qquad (6)$$

Here are the explained details of the above equation terminology found in the confusion matrix: True Positive (TP) is a consequence of which the sample forecasts the positive class correctly; True Negative (TN) is an outcome which the sample forecasts the negative class accurately; False Positive (FP) is an outcome under which the sample forecasts the positive class incorrectly; False Negative (FN) is an outcome under which the sample forecasts the negative class wrongly; condition Positive (P) the number of real positive cases in the data; condition Negative (N) the number of real negative cases in the data.

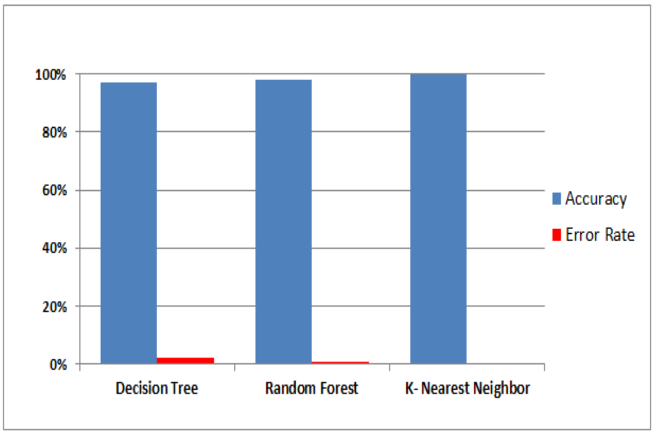

CHART 1: ACCURACY AND ERROR RATE OF CLASSIFIERS

TABLE 11: COMPARING ACCURACY, ERROR RATE, AND TIME TAKEN ON IRIS DATASET

Classifier	Accuracy	Error Rate	Time taken to build model (second)
Decision Tree (DT)	98%	2%	0.00
Random Forest (RF)	99.33%	0.6667 %	0.02
K- Nearest Neighbor (K-NN)	100%	0%	0.00

VI. COMPARATIVE STUDIES

Several classification algorithms based on the classifier used in this study were recorded in related works in this article, illustrating the important tasks that were posed by the researchers with each method tested. In this part, the results achieved in this study were compared with the studies that have been achieved by the research in the related work. Study [25] used j48 and RF on IRIS datasets to increase their efficiency and obtained 95.83% accuracy for j48 and 95.55% accuracy for RF, as seen in related work. Compared to this research, the author used j48 and RF to check for IRIS flower, but got better results, with j48 and RF accuracy of 98% and 99.33%, respectively.

In the study [29], many optimization methods were used to classify flowers on the IRIS datasets. In comparison to other approaches such as K-NN, LR, and NN, the SVM method had the best precision, which was 98%, according to the evaluation results. However, in this study, the employed classifier performed on the same datasets namely IRIS flowers, while the researcher obtained better performance, which was 100% obtained by K-NN. Although the handprint recognition method was performed using a variety of approaches in image datasets in the study [32]. The strongest outcome achieved between them is that it has a 99.7% precision. In comparison, three classifiers are used on IRIS based on this research, with much better results than Study [32], where the strongest classifier is the K-NN.

Finally, when the results of related work research algorithms are compared to the results of this research, the results of this analysis tend to be higher, as shown in "TABLE 11."As contrasted to the other algorithms, K-NN worked higher than DT and RF, with 100% accuracy and no error rate. In addition, "TABLE 12" outlines the relation of this analysis to the studies analyzed by the researchers in the related work.

TABLE 12: COMPARISON BETWEEN THIS STUDY AND RELATED WORK

Study	Dataset(s)	Classifier	Accuracy
[25]	IRIS	- J48 - RF	J48: 95.83% RF: 95.55%
[26]	IRIS, Car Assessment, Bottle, and WINE	- DT (C4.5)	DT (C4.5): 92%
[27]	IRIS	- PCA - LDA - LR - RF	PCA: 86% LDA: 100% LR: 96% RF: 94%
[28]	Iris, Pima, Seeds, Waveform, WDBC, Wine, and Pen-based	- K-NN	K-NN: 95.26%.
[29]	IRIS	- NN - LR - K-NN - SVM	NN: 96.67% LR: 96.67% K-NN: 96.67% SVM: 98%.
[30]	facial images	- 2DPCA - K-NN	- 2DPCA: 94.74% - K-NN: 97.37%
[31]	breast cancer	- RF	RF: 95%
[32]	Images	- PRS - LBP - DT(C5.0) - K-NN	PRS: 99.7% LBP: 92% DT(C5.0): 70.25% K-NN: 95%,
[33]	Eyes images	- NN - DT - Back Propagation	hybridization of the three algorithms make the system model accurate and efficient
[34]	IRIS	- DT	DT: 98%
This Study	IRIS	- DT - RF - K-NN	DT: 98% RF: 99.33% K-NN: 100%

VII. CONCLUSION

Nowadays, classification is the most often utilized in machine learning problems with a number of applications such as face recognition, flower classification, clustering, and so on. In order to construct a model, the classification algorithm creates a connection between the input and output characteristics and attempts to predict the target population with the greatest accuracy. The main objective of this study was to come to a consensus on how well K-nearest neighbors, decision tree (j48), and random forest algorithms performed in IRIS flower classification. According to the findings, both approaches yield strong classification outcomes, and the precision is calculated by the number of principal components used. The analysis also found that when the percentage of training data improves, so does the degree of precision. In comparison to random forest, which achieved 99.33% accuracy, and decision tree (j48), which achieved 98% accuracy, the experimental findings revealed that K-nearest neighbors performed significantly better, achieving 100% accuracy. In the future, analyses on separate data sets will be generated, and different methods will be utilized and mixed to produce improved distinction results.

ACKNOWLEDGEMENT

The researchers would like to indicate that Duhok Polytechnic University supported this study with all of its financial services and motivation.

REFERENCES

[1] M. J. H. Mughal, “Data Mining: Web Data Mining Techniques, Tools and Algorithms: An Overview,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, 2018, doi: 10.14569/IJACSA.2018.090630.

[2] D. Q. Zeebaree, A. M. Abdulazeez, O. M. S. Hassan, D. A. Zebari, and J. N. Saeed, *Hiding Image by Using Contourlet Transform*. press, 2020.

[3] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[4] M. A. Sulaiman, "Evaluating Data Mining Classification Methods Performance in Internet of Things Applications," *J. Soft Comput. Data Min.*, vol. 1, no. 2, pp. 11–25, 2020.

[5] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and D. A. Zebari, "Machine learning and region growing for breast cancer segmentation," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 88–93.

[6] S. H. Haji and A. M. Abdulazeez, "COMPARISON OF OPTIMIZATION TECHNIQUES BASED ON GRADIENT DESCENT ALGORITHM: A REVIEW," *PalArchs J. Archaeol. Egypt Egyptol.*, vol. 18, no. 4, Art. no. 4, Feb. 2021.

[7] I. Ibrahim and A. Abdulazeez, "The Role of Machine Learning Algorithms for Diagnosing Diseases," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 10–19, 2021.

[8] P. Galdi and R. Tagliaferri, "Data mining: accuracy and error measures for classification and prediction," *Encycl. Bioinforma. Comput. Biol.*, pp. 431–6, 2018.

[9] D. Maulud and A. M. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 4, pp. 140–147, 2020.

[10] G. Gupta, "A self explanatory review of decision tree classifiers," in *International conference on recent advances and innovations in engineering (ICRAIE-2014)*, 2014, pp. 1–7.

[11] N. S. Ahmed and M. H. Sadiq, "Clarify of the random forest algorithm in an educational field," in *2018 international conference on advanced science and engineering (ICOASE)*, 2018, pp. 179–184.

[12] T. Bahzad and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *J. Appl. Sci. Technol. Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[13] D. Q. Zeebaree, H. Haron, and A. M. Abdulazeez, "Gene selection and classification of microarray data using convolutional neural network," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 2018, pp. 145–150.

[14] N. M. Abdulkareem and A. M. Abdulazeez, "Machine Learning Classification Based on Radom Forest Algorithm: A Review," *Int. J. Sci. Bus.*, vol. 5, no. 2, pp. 128–142, 2021.

[15] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, 2015.

[16] A. S. Eesa, A. M. Abdulazeez, and Z. Orman, "A DIDS Based on The Combination of Cuttlefish Algorithm and Decision Tree," *Sci. J. Univ. Zakho*, vol. 5, no. 4, pp. 313–318, 2017.

[17] K. Rai, M. S. Devi, and A. Guleria, "Decision tree based algorithm for intrusion detection," *Int. J. Adv. Netw. Appl.*, vol. 7, no. 4, p. 2828, 2016.

[18] M. Czajkowski and M. Kretowski, "Decision tree underfitting in mining of gene expression data. An evolutionary multi-test tree approach," *Expert Syst. Appl.*, vol. 137, pp. 392–404, 2019.

[19] D. M. Abdulqader, A. M. Abdulazeez, and D. Q. Zeebaree, "Machine Learning Supervised Algorithms of Gene Selection: A Review," *Mach. Learn.*, vol. 62, no. 03, 2020.

[20] S. Dahiya, R. Tyagi, and N. Gaba, "Comparison of ML classifiers for Image Data," EasyChair, 2020.

[21] S. F. Khorshid and A. M. Abdulazeez, "BREAST CANCER DIAGNOSIS BASED ON K-NEAREST NEIGHBORS: A REVIEW," *PalArchs J. Archaeol. EgyptEgyptology*, vol. 18, no. 4, pp. 1927–1951, 2021.

[22] D. A. Zebari, D. Q. Zeebaree, A. M. Abdulazeez, H. Haron, and H. N. A. Hamed, "Improved Threshold Based and Trainable Fully Automated Segmentation for Breast Cancer Boundary and Pectoral Muscle in Mammogram Images," *IEEE Access*, vol. 8, pp. 203097–203116, 2020.

[23] A. Torfi, "Nearest Neighbor Classifier–From Theory to Practice," 2020.

[24] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and D. A. Zebari, "Trainable model based on new uniform LBP feature to identify the risk of the breast cancer," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 106–111.

[25] Y. Lakhdoura and R. Elayachi, "Comparative Analysis of Random Forest and J48 Classifiers for 'IRIS' Variety Prediction," *Glob. J. Comput. Sci. Technol.*, 2020.

[26] M. M. Mijwil and R. A. Abttan, "Utilizing the Genetic Algorithm to Pruning the C4. 5 Decision Tree Algorithm," *Asian J. Appl. Sci. ISSN 2321–0893*, vol. 9, no. 1, 2021.

[27] D. Rana, S. P. Jena, and S. K. Pradhan, "Performance Comparison of PCA and LDA with Linear Regression and Random Forest for IRIS Flower Classification," *PalArchs J. Archaeol. EgyptEgyptology*, vol. 17, no. 9, pp. 2353–2360, 2020.

[28] C. Gong, Z. Su, P. Wang, and Q. Wang, "Cumulative belief peaks evidential K-nearest neighbor clustering," *Knowl.-Based Syst.*, vol. 200, p. 105982, 2020.

[29] A. Shukla, A. Agarwal, H. Pant, and P. Mishra, "Flower Classification using Supervised Learning," vol. 9, no. 05, pp. 757–762, May 2020.

[30] E. Sugiharti and A. T. Putra, "Facial recognition using two-dimensional principal component analysis and k-nearest neighbor: a case analysis of facial images," in *Journal of Physics: Conference Series*, 2020, vol. 1567, no. 3, p. 032028.

[31] J. Quist, L. Taylor, J. Staaf, and A. Grigoriadis, "Random Forest Modelling of High-Dimensional

Mixed-Type Data for Breast Cancer Classification," *Cancers*, vol. 13, no. 5, p. 991, 2021.

[32] M. S. KADHM, H. AYAD, and M. J. MOHAMMED, "PALMPRINT RECOGNITION SYSTEM BASED ON PROPOSED FEATURES EXTRACTION AND (C5. 0) DECISION TREE, K-NEAREST NEIGHBOUR (KNN) CLASSIFICATION APPROACHES," *J. Eng. Sci. Technol.*, vol. 16, no. 1, pp. 816–831, 2021.

[33] R. O. Ogundokun, P. O. Sadiku, S. Misra, O. E. Ogundokun, J. B. Awotunde, and V. Jaglan, "Diagnosis of Long Sightedness Using Neural Network and Decision Tree Algorithms," in *Journal of Physics: Conference Series*, 2021, vol. 1767, no. 1, p. 012021.

[34] K. Sarpatwar *et al.*, "Privacy Enhanced Decision Tree Inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 34–35.

[35] Z. Bilgin and M. Gunestas, "Explaining Inaccurate Predictions of Models through k-Nearest Neighbors," 2021.

[36] Y. A. Yakub, "DATA MINING COURSEWORK," 2019.

[37] M. S. Abirami and J. Vasavi, "A Qualitative Performance Comparison Of Supervised Machine Learning Algorithms For Iris Recognition," *Eur. J. Mol. Clin. Med.*, vol. 7, no. 6, pp. 1937–1946, 2020.

[38] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[39] A. Viloria, G. C. Acuña, D. J. A. Franco, H. Hernández-Palma, J. P. Fuentes, and E. P. Rambal, "Integration of data mining techniques to PostgreSQL database manager system," *Procedia Comput. Sci.*, vol. 155, pp. 575–580, 2019.

[40] S. Raschka, "Naive Bayes and Text Classification I - Introduction and Theory," *ArXiv14105329 Cs*, Feb. 2017, Accessed: Apr. 03, 2021. [Online]. Available: http://arxiv.org/abs/1410.5329.

[41] R. Kumar and R. Verma, "Classification algorithms for data mining: A survey," *Int. J. Innov. Eng. Technol. IJIET*, vol. 1, no. 2, pp. 7–14, 2012.

[42] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[43] S. Singh and P. Gupta, "Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey," *Int. J. Adv. Inf. Sci. Technol. IJAIST*, vol. 27, no. 27, pp. 97–103, 2014.

[44] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Adv. Space Res.*, vol. 41, no. 12, pp. 1955–1959, 2008.

[45] Priyanka and D. Kumar, "Decision tree classifier: a detailed survey," *Int. J. Inf. Decis. Sci.*, vol. 12, no. 3, pp. 246–269, 2020.

[46] V. Cheushev, D. A. Simovici, V. Shmerko, and S. Yanushkevich, "Functional entropy and decision trees," in *Proceedings. 1998 28th IEEE International Symposium on Multiple-Valued Logic (Cat. No. 98CB36138)*, 1998, pp. 257–262.

[47] T. Maszczyk and W. Duch, "Comparison of Shannon, Renyi and Tsallis entropy used in decision trees," in *International Conference on Artificial Intelligence and Soft Computing*, 2008, pp. 643–651.

[48] Y. Liu, L. Hu, F. Yan, and B. Zhang, "Information gain with weight based decision tree for the employment forecasting of undergraduates," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 2210–2213.

[49] D. Bui, B. Pradhan, O. Löfman, and I. Revhaug, "Landslide Susceptibility Assessment in Vietnam Using Support Vector Machines, Decision Tree, and Naïve Bayes Models," *Math. Probl. Eng.*, vol. 2012, p. 26, Apr. 2012, doi: 10.1155/2012/97/46/38.

[50] A. M. Abdulazeez, D. M. Hajy, D. Q. Zeebaree, and D. A. Zebari, "Robust watermarking scheme based LWT and SVD using artificial bee colony optimization," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, pp. 1218–1229, 2021.

[51] S. Sathyadevan and R. R. Nair, "Comparative analysis of decision tree algorithms: ID3, C4. 5 and random forest," in *Computational intelligence in data mining-volume 1*, Springer, 2015, pp. 549–562.

[52] I. Reis, D. Baron, and S. Shahaf, "Probabilistic random forest: A machine learning algorithm for noisy data sets," *Astron. J.*, vol. 157, no. 1, p. 16, 2018.

[53] A. Wadoux, D. Brus, and G. Heuvelink, "Sampling design optimization for soil mapping with random forest," *Geoderma*, vol. 355C, Aug. 2019, doi: 10.1016/j.geoderma.2019.113913.

[54] L. Demidova and M. Ivkina, "Defining the Ranges Boundaries of the Optimal Parameters Values for the Random Forest Classifier," in *2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA)*, 2019, pp. 518–522.

[55] C. Iwendi *et al.*, "COVID-19 patient health prediction using boosted random forest algorithm," *Front. Public Health*, vol. 8, p. 357, 2020.

[56] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[57] D. Devetyarov and I. Nouretdinov, "Prediction with confidence based on a random forest classifier," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2010, pp. 37–44.

[58] Y. He, C. Wang, F. Chen, H. Jia, D. Liang, and A. Yang, "Feature comparison and optimization for 30-m winter wheat mapping based on Landsat-8 and Sentinel-2 data using random forest algorithm," *Remote Sens.*, vol. 11, no. 5, p. 535, 2019.

[59] J. Gou, T. Xiong, and Y. Kuang, "A Novel Weighted Voting for K-Nearest Neighbor Rule," *J. Comput.*, vol. 6, no. 5, pp. 833–840, May 2011, doi: 10.4304/jcp.6.5.833-840.

[60] J. Gou, L. Du, Y. Zhang, and T. Xiong, “A New Distance-weighted k -nearest Neighbor Classifier,” *J Inf Comput Sci*, vol. 9, Nov. 2011.

[61] Zhe Zhou, Chenglin Wen, and Chunjie Yang, “Fault Detection Using Random Projections and k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes,” *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 70–79, Feb. 2015, doi: 10.1109/TSM.2014.2374339.

[62] D. Q. Zeebaree, A. M. Abdulazeez, D. A. Zebari, H. Haron, and H. N. A. Hamed, “Multi-Level Fusion in Ultrasound for Cancer Detection Based on Uniform LBP Features,” 2021.

[63] S. Rodríguez González *et al.*, Eds., *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, vol. 1242. Cham: Springer International Publishing, 2021.

[64] M. Khan, Q. Ding, and W. Perrizo, “k-nearest neighbor classification on spatial data streams using P-trees,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2002, pp. 517–528.

[65] A. Kataria and M. D. Singh, *A Review of Data Classification Using K-Nearest Neighbour Algorithm*, vol. 3. 2013.

[66] P. Čech, J. Lokoč, and Y. N. Silva, “Pivot-based approximate k-NN similarity joins for big high-dimensional data,” *Inf. Syst.*, vol. 87, p. 101410, Jan. 2020, doi: 10.1016/j.is.2019.06.006.

[67] B. Bratić, M. E. Houle, V. Kurbalija, V. Oria, and M. Radovanović, “NN-Descent on High-Dimensional Data,” in *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, Novi Sad Serbia, Jun. 2018, pp. 1–8, doi: 10.1145/3227609.3227643.

[68] Y. Yang, H.-G. Yeh, W. Zhang, C. J. Lee, E. N. Meese, and C. G. Lowe, “Feature Extraction, Selection, and K-Nearest Neighbors Algorithm for Shark Behavior Classification Based on Imbalanced Dataset,” *IEEE Sens. J.*, vol. 21, no. 5, pp. 6429–6439, Mar. 2021, doi: 10.1109/JSEN.2020.3038660.

[69] D. H. B. Kekre, T. Management, S. D. Thepade, M. P. S. Of, A. Parkar, and T. S. Engineering, *A Comparison of Haar Wavelets and Kekre’s Wavelets for Storing Colour Information in a Greyscale Image*. .

[70] A. Pandey, “MACHINE LEARNING BASED DDoS ATTACK DEDUCTION USING WEKA,” 2020.

[71] R. R. Bouckaert *et al.*, “WEKA—Experiences with a Java Open-Source Project,” p. 9, 2010.

[72] K. P. S. Attwal and A. S. Dhiman, “Exploring data mining tool-Weka and using Weka to build and evaluate predictive models,” *Adv. Appl. Math. Sci.*, vol. 19, no. 6, pp. 451–469, 2020.

[73] S. B. Aher and L. Lobo, “Data mining in educational system using weka,” in *International Conference on Emerging Technology Trends (ICETT)*, 2011, vol. 3, pp. 20–25.