In-Depth Comparison of Supervised Classification Models - Performance and Adaptability to Practical Requirements

Mouataz IDRISSI KHALDI[1], Allae ERRAISSI[2], Mustapha HAIN[3], Mouad BANANE[4]
Laboratory of Artificial Intelligence & Complex Systems Engineering,
Hassan II University, ENSAM, Casablanca, Morocco[1, 3, 4]
Chouaib Doukkali University, El Jadida, Morocco[2]

***Abstract*—In this paper, we carried out an in-depth comparative analysis of five major supervised classification algorithms: Naïve Bayes, Decision Tree, Random Forest, KNN and SVM. These models were evaluated through a rigorous literature review, based on 20 criteria grouped into five key dimensions: algorithm performance, computational efficiency, practicality and ease of use, data compatibility and practical applicability. The results show that each algorithm has specific strengths and limitations: SVM and Random Forest stand out for their robustness and accuracy in complex environments, while Naïve Bayes and Decision Tree are appreciated for their speed, simplicity and interpretability. KNN, despite its intuitive approach, suffers from high complexity in the prediction phase, limiting its effectiveness on large datasets. This study aims to provide a structured framework for researchers and practitioners in various fields, such as healthcare, finance, industry and education, where supervised classification algorithms play a central role in decision-making. In addition, the results highlight the importance of selecting algorithms according to specific needs, and open up promising prospects, including the development of hybrid models and improved real-time data processing.**

Keywords—Supervised classification; Naïve Bayes; decision tree; Random Forest; k-nearest neighbor; Support Vector Machine; algorithm performance; interpretability

I. Introduction

In today's era of megadata, supervised classification models play a central role in transforming raw data into actionable information in critical fields such as healthcare, finance and industry. The rise of artificial intelligence (AI) and the increasing availability of data have reinforced the importance of supervised classification as a pillar of machine learning systems. These models are used, for example, to improve medical diagnosis by identifying complex patterns in health data, to detect fraud in finance and to predict preventive maintenance in the manufacturing industry. However, in view of the growing diversity of available models and the varied requirements of practical applications, in-depth benchmarking is needed to guide informed choices and maximize the effectiveness of solutions.

The massive increase in data volumes is accompanied by a proliferation of classification models, each with advantages and limitations depending on the application context. This diversity complicates the choice of the optimal model, requiring an assessment of technical performance, resource requirements and adaptability to specific data types, such as text, images and time series. For example, the prediction of academic success or predictive diagnostics in medicine require models capable of combining accuracy, robustness and speed. A thorough understanding of the trade-offs between these aspects is therefore essential to ensure that models meet performance targets.

Our study responds to a pressing need for synthetic and comprehensive comparisons of supervised classification models. Existing research often focuses on specific criteria, such as accuracy or algorithmic complexity, without offering a clear overview that takes into account the impact on real applications. In addition, certain gaps persist, notably concerning the adaptability of models to multimodal or unbalanced data, and the management of trade-offs between accuracy and computational resources. By proposing a literature review focusing on these criteria, this study aims to provide a valuable tool for researchers and practitioners, enabling them to select algorithms suited to their specific needs.

Faced with these challenges, the central problem of this study can be formulated as follows: What are the best-performing supervised classification models in terms of accuracy, robustness, computational complexity and adaptability, and how does their performance vary according to data type and application context? To answer this question, an in-depth literature review appeared to be the most appropriate method. This approach makes it possible to pool and analyze a wide range of academic and empirical studies on supervised classification models, thus drawing out clear trends and identifying existing trade-offs.

The methodology adopted is based on a rigorous bibliographical search covering publications from scientific journals, international conferences and recent empirical studies. The selected models are evaluated according to defined criteria, such as prediction accuracy, generalizability, robustness, computation time and adaptability to heterogeneous and missing data. These criteria were chosen for their relevance to the evaluation of practical performance, and for their ability to cover the actual requirements of the systems deployed. Strict inclusion criteria were applied to guarantee the representativeness and quality of the studies selected.

The results show that some models, such as Random Forest and SVM, stand out for their robustness and accuracy in complex contexts, while models like Naïve Bayes and Decision Tree are appreciated for their interpretability, ease of implementation and speed. Comparative analysis has identified notable trade-offs between accuracy and computational complexity, offering guidelines for optimal model selection according to system constraints.

In conclusion, this study proposes a framework for the selection of supervised classification algorithms according to specific user needs. It also highlights promising avenues of research, such as the exploration of hybrid models combining the strengths of the algorithms compared in order to overcome their limitations. Prospects include the development of models better adapted to the challenges of big data and real-time processing requirements.

The paper is structured into five main sections. Section I is dedicated to the introduction and presents the background, problem and objectives of the study. Section II provides a detailed theoretical framework on supervised classification and discusses the main models, while presenting their current limitations and challenges. This section is subdivided into several parts, including an overview of machine learning, a description of the types of supervised classification algorithms and a discussion of their limitations. Section III describes the methodology adopted, covering algorithm selection, evaluation criteria and data collection. Section IV presents the results of the comparative analysis and offers an in-depth discussion of the observations made, followed by a reflection on the limitations of the results. Finally, Section V concludes the study by summarizing the main contributions and suggesting avenues for future research.

II. Theoretical Framework and Background

A. Introduction to Supervised Classification Models

1) Machine learning-an overview: Machine Learning is a branch of artificial intelligence that aims to equip systems with the ability to discover patterns and relationships within datasets, without being explicitly programmed for each specific task [1][2][3]. By forming models from training data, it enables the automation of complex tasks such as image recognition or the prediction of user behavior [1][3][4]. This field relies on techniques that enable machines to learn from past data or interactions with their environment, without requiring specific programming for each task [5][6]. More broadly, Machine Learning encompasses the analysis, design, development and implementation of methods that enable machines to systematically improve themselves, offering solutions to problems through advanced algorithmic approaches [7].

As Mitchell explains in [1], a learning system is able to improve its performance with experience, i.e. from data. Machine Learning can be divided into three broad categories: supervised, unsupervised and reinforcement learning. This classification is based on the way the model learns from the data.

2) Supervised learning: Supervised Learning is an approach to Machine Learning in which a model is trained on a set of labeled data. This means that each instance of training data is associated with a known label or response. The model uses these examples to learn to predict the label of new instances.

In supervised learning, two main tasks are often distinguished:

- Regression: The model predicts a continuous numerical value as a function of input data (e.g. temperature prediction).
- Classification: the model predicts a category or class from a predefined set (e.g., spam detection in e-mail).

Supervised classification, the focus of our study, is therefore a sub-category of supervised learning. Hastie et al in [2], point out that supervised learning is particularly effective when data is abundant and well-labeled, as it enables the construction of accurate and robust models.

3) Supervised classification: Supervised Classification involves using a set of labeled data to train a model that can predict the class to which new observations belong. This is a key task in many fields, from medicine (e.g. medical diagnosis) to security (intrusion detection) to financial services (fraud detection).

The Classification model learns to distinguish between different classes by identifying the characteristics that separate them. For example, in a dataset containing images of dogs and cats, a classification model learns to spot specific characteristics to correctly identify each category.

The advantages of Supervised Classification include:

High accuracy: when data is reliably labeled.

Adaptability: Models can be applied in a variety of sectors requiring precise categorization.

Transparency: Some algorithms, such as decision trees, allow decisions to be interpreted, making it easier to understand the criteria used.

However, these models generally require large, well-balanced data sets to avoid bias and ensure good generalizability. According to Domingos in [3], it is crucial that the training data represent the future data well for the model to be robust and reliable.

B. Main Types of Classification Models

1) Naïve Bayes: Naïve Bayes is a supervised classification algorithm based on Bayes' theorem and the assumption of conditional independence between data characteristics. This algorithm is particularly appreciated for its simplicity and speed, making it highly effective in automatic natural language processing (NLP) and text classification tasks [8].

Despite its simplifying assumption, Naïve Bayes often performs competitively against more complex algorithms.

Mathematically, the algorithm is based on Bayes' theorem, formulated as follows:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{1}$$

Where $P(C|X)$ represents the probability that an observation X belongs to a class C.To simplify the calculation, Naïve Bayes assumes that features x_i are conditionally independent, allowing the following approximation:

$$P(X|C) = \prod_{i=1}^{n} P(x_i|C) \tag{2}$$

This simplification makes Naïve Bayes particularly fast and undemanding in terms of computational resources.

Historically, this algorithm was introduced in the 1960s as a statistical tool for probabilistic classification. Its adoption in machine learning was reinforced by the work of [8], who demonstrated its effectiveness in contexts where conditional independence is respected.

The algorithm works by estimating the probability of each class for a given observation, then assigning that observation to the class with the highest probability. Naïve Bayes calculates these probabilities using pre-processed training data.

There are several variants of Naïve Bayes adapted to specific types of data. Gaussian Naïve Bayes, for example, is used for continuous data, assuming a normal distribution of features [9]. Multinomial Naïve Bayes is particularly suited to discrete data, such as word occurrences in text documents [2], while Bernoulli Naïve Bayes applies to binary data.

Naïve Bayes has many advantages, not least its simplicity of implementation and speed of execution, even on high-dimensional data. However, it suffers from limitations when features are highly correlated or when the conditional independence assumption is violated, which can reduce its accuracy [8].

The algorithm has applications in various fields. For example, it is used for spam filtering, where it classifies e-mails as junk or not [10]. In sentiment analysis, it is used to determine the polarity (positive or negative) of textual opinions [2]. Finally, it is applied to facial recognition, rapidly identifying patterns based on discrete features [9].

In short, Naïve Bayes is a fundamental tool in the supervised learning toolbox, combining mathematical simplicity with practical efficiency. Its remarkable performance in specific contexts, combined with its speed, explains its widespread adoption in many applications.

2) Decision tree: A decision tree is a supervised classification model structured in the form of a tree, where each node represents a question or test on a feature of the data, each branch corresponds to a possible outcome of that test, and each leaf represents a class or final decision. Decision trees are widely used because of their simplicity and interpretability, enabling users to visualize and understand the decision-making process clearly and explicitly [10].

Decision trees are based on sound mathematical foundations, notably the division criteria used to determine the most discriminating features at each stage. Among these criteria, information gain, used by the ID3 and C4.5 algorithms, measures entropy reduction after a specific division [10]. It is defined by the following formula:

$$Gain(T,X) = Entropy(T) - \sum_{v \in V} \frac{|T_v|}{|T|}.Entropy(T_v) \tag{3}$$

where T is the data set at the current node, X is the feature being tested, and T_v represents the subset of data corresponding to a value v of X. The Gini index, used by the CART algorithm, is another popular criterion, quantifying the homogeneity of classes in the subsets generated. It is defined by :

$$Gini(T) = 1 - \sum_{i=1}^{C} P_i^2 \tag{4}$$

where P_i is the proportion of data belonging to class i [11].

The history of decision trees is marked by significant advances. Ross Quinlan introduced the ID3 algorithm in 1986, one of the first models based on information gain [12]. He later developed C4.5, an extension of ID3 capable of handling continuous data and missing values [10]. At the same time, Leo Breiman proposed the CART (Classification and Regression Trees) algorithm in 1984, which introduced the use of the Gini index and opened up possible applications for regression [11]. These contributions laid the methodological foundations for decision trees, positioning them as an essential tool in supervised learning.

Decision trees work by recursively dividing the data according to the most discriminating characteristics, based on the criteria mentioned. For example, to predict whether a person will buy a product based on their age and income, a decision tree would ask questions such as: “Is the person under 30 ?” or “Is the income over 50,000 ?”. Each answer leads to a specific decision, represented by a leaf in the tree.

Decision trees come in several variants, adapted to specific needs. ID3 (Iterative Dichotomiser 3), proposed by [12], is based on information gain as the dividing criterion. Its extensions, C4.5 and C5.0, introduce the management of continuous data and missing values [10]. Finally, CART, developed by study [11], uses the Gini index and can be applied to classification and regression tasks.

The advantages of decision trees are numerous. They offer a high degree of interpretability, thanks to their explicit and easily visualized structure [10]. They can be adapted to both categorical and continuous data, making them versatile in a variety of contexts [11]. What's more, their speed makes them effective for medium-sized data sets. However, they also have their limitations. Decision trees are prone to overlearning, particularly in the absence of regularization, which can impair their generalization ability [13]. They are also sensitive to variations in the training data, which can affect their stability [2].

In terms of applications, decision trees play a key role in various fields. In bioinformatics, they are used to identify genes associated with disease [10]. In the financial sector, they are used to analyze credit decisions and detect fraud [11]. Finally, in industry, they are applied for quality control and predictive maintenance [9]. Their flexibility and simplicity explain their widespread adoption in sectors requiring clear, explainable decisions.

In conclusion, decision trees represent a robust and intuitive method for solving classification and regression problems. Although they present challenges such as overlearning and data sensitivity, their advantages and wide range of applications make them an indispensable tool in the arsenal of supervised classification algorithms.

3) Random forest: Random Forest is a powerful ensemble algorithm that combines several independent decision trees to improve the robustness and accuracy of predictions. Developed by [14], this algorithm is based on two fundamental principles: bootstrap aggregation and random feature selection [14]. By aggregating the predictions of several trees, Random Forest reduces errors due to variance and increases the stability of results.

Mathematically, each tree is constructed from a bootstrap sample of the training data. At each node, only a random sub-selection of features is evaluated to determine the best split. This approach introduces diversity between trees, improving the generalizability of the model.

Historically, Random Forest was conceived as an improvement on conventional decision trees, in response to their tendency to overlearn. [14] Demonstrated that the introduction of randomness in both data and features makes the model more resistant to noisy data and outliers.

Random Forest works by independently generating a large number of decision trees, then combining their predictions. For a classification task, the predicted class is determined by a majority vote among the trees. In regression, the final prediction is obtained by averaging the tree outputs.

The algorithm has several advantages. It is robust to noisy data, efficiently handles unbalanced classes and requires no prior data normalization. However, its main drawback is its increased complexity, which makes interpretation more difficult than with a single decision tree [13].

Random Forest is used in a variety of fields. In finance, it is applied to credit scoring and fraud detection [14]. In bioinformatics, it is used for feature selection and classification of genomic data [9]. In the energy sector, it is used to predict energy consumption as a function of various factors [13].

In conclusion, Random Forest is a flexible, high-performance method for a wide range of classification and regression tasks. Its robustness and precision make it an indispensable tool in complex, noisy environments.

4) K-Nearest Neighbor (KNN): K-Nearest Neighbor (KNN) is a supervised classification algorithm based on the concept of proximity in a multidimensional space. It classifies an observation according to the majority class among its k nearest neighbors. Introduced by study [15], KNN is one of the simplest and most intuitive algorithms in supervised learning [15].

In mathematical terms, KNN relies on a distance measure to assess the proximity between observations. The Euclidean distance is the most commonly used and is defined by:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (5)$$

where x_i and y_i represent the characteristics of points x and y. For a new point, KNN identifies the k nearest neighbors, then determines the majority class among them.

Historically, KNN has gained in popularity thanks to its simplicity and adaptability to different types of data. However, it is particularly sensitive to the choice of k, which directly influences its performance. Too small a k makes the model sensitive to noise, while too large a k can dilute important features [2].

KNN works as follows: for each new observation, the algorithm identifies the k closest observations in the training data, using a predefined distance measure. It then assigns the majority class among these neighbors to the observation in question.

KNN offers several advantages, not least its simplicity and effectiveness on small or well-separated data sets. However, it is computationally expensive for large datasets, as it requires each new observation to be compared with the training dataset. In addition, it is sensitive to irrelevant or redundant features, which can affect its accuracy [2].

KNN has many applications. In product recommendation, it identifies similar products based on user preferences. In image recognition, it classifies images based on distances in pixel space. In healthcare, KNN is used to diagnose diseases based on biological characteristics [2].

In summary, KNN is a versatile and efficient algorithm for classification and regression tasks. Despite its limitations in terms of computational complexity, its simplicity and adaptability make it a preferred choice for applications requiring proximity-based classification.

5) Support Vector Machine (SVM): The Support Vector Machine (SVM) is a supervised classification algorithm designed to maximize the margin between classes in a multidimensional space. Developed by study [16], SVM revolutionized classification methods by introducing the idea of kernels to project data into higher-dimensional spaces, thus facilitating their linear separation [16].

$$\min_{\omega,b} \frac{1}{2} \|\omega\|^2 \quad (6)$$

under the constraints :

$$y_i(\omega^T x_i + b) \geq 1, \forall i \quad (7)$$

where ω is the weight vector, b the bias, x_i the observations and y_i their respective labels. By maximizing the margin between the points closest to the classes, called support vectors, SVM guarantees better generalization [17].

The use of kernels enables SVM to handle non-linear problems. Common kernels include linear, polynomial and radial basis function (RBF). These functions transform the data into a higher-dimensional space, where it becomes linearly separable [16].

SVM works by constructing an optimal hyperplane that maximizes the margin between classes. For non-linear data, SVM uses kernels to perform an implicit transformation into a higher-dimensional space. In this way, data are optimally separated, minimizing classification errors.

SVM is particularly efficient for small to medium-sized datasets, but its computational cost can become prohibitive for large databases. In addition, it requires careful selection of hyperparameters, such as the C parameter (controlling the trade-off between maximum margin and error) and the choice of kernel [17].

SVM applications cover a wide range of fields. In bioinformatics, it is used to classify genes and predict diseases. In the financial sector, it analyzes trends and predicts market behavior. In computer vision, SVM is applied to facial and object recognition [2].

In conclusion, SVM is a powerful and flexible algorithm, capable of processing both linear and non-linear data. Although it requires expertise to optimize its parameters, its accuracy and ability to handle complex problems make it a preferred choice for many applications.

C. Current Limitations and Challenges

Despite their many applications and successes, supervised classification algorithms have inherent limitations and challenges that impact their adoption in certain contexts. These limitations vary from algorithm to algorithm and are influenced by factors such as the nature of the data, available resources, and the specific requirements of application domains.

1) Dependence on data quality: The performance of supervised classification algorithms is highly dependent on the quality of the training data. Noisy, unbalanced or incomplete data can lead to bias and reduced accuracy [2][18]. For example, Decision Trees are prone to overfitting when training data contains anomalies or extreme values [10].

2) Scalability problem: Algorithms such as KNN and SVM can be inefficient on large datasets due to their algorithmic complexity [9][14]. KNN, with its O(n) complexity, becomes impractical for databases containing millions of points [15].

3) Lack of interpretability: Some algorithms, such as Random Forest, produce complex models that are difficult for non-expert users to interpret [14][19]. Although Random Forest offers high accuracy, its inner workings are often described as “black box”, limiting its adoption in sensitive fields such as medicine [19].

4) Sensitivity to overfitting: Models such as Decision Trees tend to overfit on training data, losing their ability to generalize on new data [10][2]. An unregularized Decision Tree may provide perfect results on training data but fail on test sets [13].

5) Resources and computing time: Algorithms like SVM and Random Forest require significant resources for training, which can be prohibitive in resource-constrained environments [9][16]. SVM with nonlinear kernels can require several hours to train on large datasets [16].

6) Adaptability problem: Some models, such as Naïve Bayes, assume feature independence, which limits their performance on datasets where features are highly correlated [8][9]. In textual datasets, Naïve Bayes' conditional independence assumption can lead to a significant loss of accuracy [8].

7) Hyperparameter dependence: Algorithm performance often depends on the fine-tuning of hyperparameters, which requires in-depth expertise and resources for cross-validation [2][16]. SVM efficiency is strongly influenced by the choice of kernel, C, and γ [2].

8) Generalization and robustness: Models sometimes lack robustness in the face of unbalanced or noisy data, which can lead to inconsistent performance [20][21]. Although Random Forests are robust in many cases, their performance can plummet when minority classes are severely under-represented [20].

9) Ethical and regulatory challenges: The use of models in sensitive areas (health, finance) raises ethical issues related to fairness and transparency [19]. Biases introduced in training data can lead to discriminatory decisions in classification systems [19].

These limitations and challenges underline the need to tailor supervised classification algorithms to the specifics of applications. Future research should focus on developing models that are more interpretable, scalable, and robust to data variations. At the same time, solutions such as model hybridization or the integration of unsupervised approaches can help overcome some of these challenges.

III. Methodology

The aim of this work is to analyze and compare supervised classification algorithms and ensure a comprehensive and balanced evaluation based on well-defined criteria and a rigorous literature review. The methodology includes algorithm selection, definition of evaluation criteria, data collection and organization of the comparative synthesis.

A. Algorithm Selection

Five algorithms widely recognized in the scientific and industrial literature were selected: Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM).

This choice is justified by:

- Representativeness of approaches: These algorithms cover a wide range of methods (probabilistic, tree-based, instance-based, etc.).

- Popularity and relevance: Their widespread adoption in industry and research testifies to their effectiveness in a variety of contexts [14], [16].

- Richness of available studies: These algorithms have been extensively documented, enabling in-depth comparison based on reliable experimental data. Naïve Bayes for text classification and spam detection [8]. Random Forest for fraud detection and medical diagnostics [14].

B. *Criteria for Comparing Classification Algorithms*

The comparison of classification algorithms requires rigorously defined criteria to assess their performance and suitability for specific tasks. This subsection presents the criteria and sub-criteria used, explains their importance, and justifies their selection.

1) Presentation of the criteria

- Criteria for comparing supervised classification algorithms play a key role in assessing their effectiveness, performance and adaptability to different contexts. These criteria, carefully defined and justified, cover the essential dimensions for ensuring a rigorous and scientifically valid analysis.

- Prediction accuracy is one of the most fundamental criteria, measuring the percentage of observations correctly classified by a model. It is calculated as the ratio between the number of correct predictions and the total number of predictions [2]. This criterion is crucial for applications where errors could have serious consequences, such as in medicine or finance. It directly reflects the reliability of a model and its suitability for accuracy requirements in critical environments.

- Robustness represents the ability of an algorithm to maintain its performance despite noisy or disturbed data [13]. In real-world environments where data is often imperfect, this criterion ensures that the algorithm remains reliable, even in the presence of errors or anomalies in the data. This is particularly important in fields such as bioinformatics or facial recognition.

- The tendency to overfitting assesses whether a model is able to avoid too much adaptation to training data, which would compromise its ability to generalize on new data [2]. Algorithms with a high risk of overfitting often require regularization techniques to guarantee stable performance. This criterion is essential in applications requiring the ability to generalize over varied data sets.

- Generalizability is closely linked to the tendency to overfitting, and measures the extent to which an algorithm can successfully predict on unseen data [22]. This criterion is essential for assessing the durability of model performance in dynamic and unpredictable environments.

- Training time refers to the time required to build a model from training data [2]. In contexts where models need to be updated frequently, such as real-time recommendation systems, this criterion helps to identify algorithms suited to strict time constraints.

- Prediction time corresponds to the time required to provide a prediction on new data [2]. This criterion is essential in applications where responsiveness is crucial, such as autonomous vehicles or industrial control systems. It ensures that the algorithm can respond quickly to real-time requests.

- Memory requirement measures the amount of memory needed to store the model and make predictions [17]. In resource-constrained environments, such as embedded devices, this criterion enables the selection of memory-efficient algorithms.

- Algorithmic complexity evaluates the computational cost of an algorithm in terms of time and space, often expressed in O notation (Hastie et al., 2009). This criterion is crucial in determining whether an algorithm can be used efficiently with large datasets or in resource-constrained environments.

- Ease of implementation examines the simplicity with which an algorithm can be integrated into an existing system [10]. An easy-to-deploy model reduces the costs and time associated with development, which is particularly advantageous in projects requiring rapid integration.

- Model interpretability refers to a model's ability to provide understandable explanations for its predictions [19]. This criterion is essential in regulated sectors, such as healthcare or finance, where transparency of decisions is paramount in building user trust.

- The level of expertise required reflects the skills needed to configure and use an algorithm effectively [3]. This criterion is important for organizations with limited human resources in technical expertise, as it can influence the ease of model adoption.

- Available documentation and libraries also play a key role in algorithm implementation [23]. Comprehensive documentation and well-supported libraries simplify the learning process and enable faster adoption of algorithms.

- Adaptability to multimodal data assesses a model's ability to simultaneously process different types of data, such as text, images or audio signals [21]. This criterion is crucial for modern applications such as voice assistants or multimodal recognition.

- Adaptability to unbalanced data measures an algorithm's ability to efficiently handle datasets where certain classes are under-represented [20]. This criterion is essential in fields such as fraud detection or the prediction of rare diseases, where the scarcity of cases of interest complicates model training.

- Handling missing data is an important criterion for assessing an algorithm's ability to operate in environments with incomplete data [18]. Models capable of tolerating missing values are particularly valuable in applications where complete data is rarely available.

- Dimensionality reduction is a criterion that measures a model's ability to identify and use the most important features while reducing noise [24]. This improves the efficiency of algorithms and their ability to handle complex data sets.

- Application domains reflect the usefulness of algorithms in a variety of sectors, such as finance or healthcare, where solutions like Random Forest have demonstrated their effectiveness in fraud detection and genomic analysis [14]. This criterion is crucial for assessing the relevance of algorithms in specific contexts.

- Impact on practical decisions assesses the direct contribution of algorithms to decision-making. For example, Decision Trees are valued for their interpretability, facilitating reliable medical diagnoses [10]. This criterion is essential in determining how an algorithm can transform critical decision-making processes.

- Integration into existing systems measures the ease with which algorithms can be implemented. Naïve Bayes are simple to deploy and easily adaptable to tools such as spam filtering systems [3]. Conversely, SVMs, with their complexity, pose integration challenges [16], underlining the importance of this criterion in assessing their practical applicability.

- Implementation cost, finally, evaluates the resources required to operate an algorithm. Naïve Bayes, for example, are cost-effective, unlike SVMs, which require greater resources due to their algorithmic complexity [2]. This criterion is essential for judging the suitability of algorithms for real system constraints.

2) Distribution of evaluation criteria

- Technical performance: Prediction accuracy, robustness, tendency to overfitting and generalizability.

- Computational efficiency: Computation time for training and prediction, memory requirements and algorithmic complexity.

- Practicality and Ease of Use: Ease of implementation, Interpretability of models, Level of expertise required and Documentation and libraries available.

- Data compatibility: Adaptability to data (multimodal, unbalanced), Handling of missing data and Reduction of dimensionality.

- Applicability and Practical Impact: Application domains, Impact on practical decisions, Integration into existing systems and Implementation cost.

These dimensions cover the essential aspects of algorithms, ensuring a balanced assessment between their raw performance, practicality, and adaptability to real-life use cases [2] [21].

3) Data collection

a) Publication selection criteria

- Period: Publications from 2000 to 2025 to include recent advances while maintaining a historical perspective.

- Types of publication: Papers in indexed journals (Elsevier, Springer, IEEE), Proceedings of major conferences (NeurIPS, ICML, AISTATS) and relevant industrial technical reports.

- Context of studies: Include work covering various fields of application (health, finance, education, etc.).

b) Research process

- Databases searched: Google Scholar, PubMed, IEEE Xplore, Scopus.

- Screening: The selected studies had to provide detailed experimental results and quantitative comparisons between several algorithms.

c) Validation of results

- Each publication is reviewed in two stages: Analysis of the abstract and introduction to confirm relevance, and reading of the methodology and results sections to ensure data quality.

d) Data synthesis

- Thematic organization: Publications are grouped by application (healthcare, finance, industry, etc.) and evaluation criteria (accuracy, robustness, etc.).

- Metrics extraction: Experimental results (accuracy, computation time) are extracted to feed comparative analyses.

4) Organization of the comparative summary

a) Results format: Synthetic tables: Presentation of criteria in tabular form for quick comparison of algorithms.

b) Qualitative analysis: Textual explanation of algorithm strengths and weaknesses, contextualized by extracted results.

c) Cross-validation: Comparison of experimental results between different publications to reinforce reliability. Narrative analysis to incorporate variations due to specific study contexts. Fig. 1 shows the comparative study workflow.

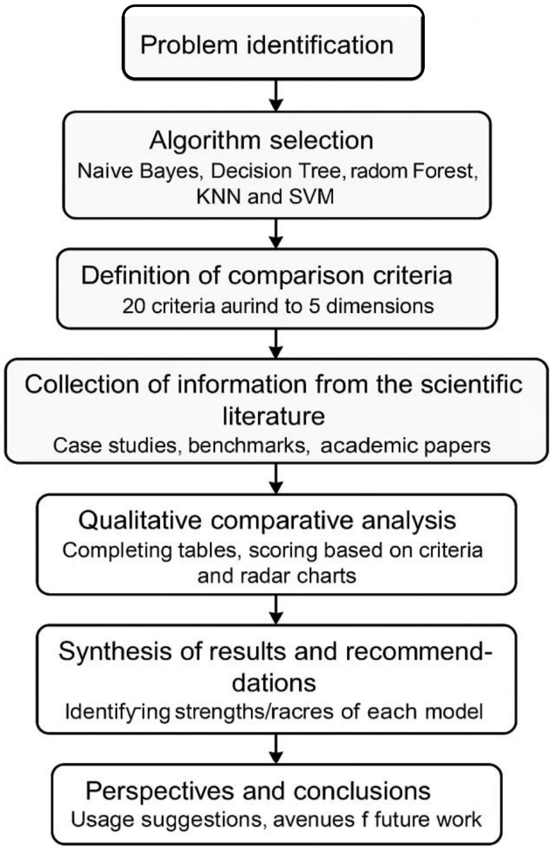
Problem identification
Algorithm selection
Naive Bayes, Decision Tree, radom Forest, KNN and SVM
Definition of comparison criteria
20 criteria aurind to 5 dimensions
Collection of information from the scientific literature
Case studies, benchmarks, academic papers
Qualitative comparative analysis
Completing tables, scoring based on criteria and radar charts
Synthesis of results and recommend-dations
Identifying strengths/racres of each model
Perspectives and conclusions
Usage suggestions, avenues f future work

Fig. 1. The comparative study workflow shows the overall study process.

IV. Results and Discussion

A. Presentation of Results

The results are organized according to the dimensions defined in the methodology: technical performance, computational efficiency, practicality, compatibility with data, and practical applicability.

1) Technical performance: This table evaluates the accuracy, robustness to perturbations, tendency to overfitting, and generalizability of the models. The qualitative scores are normalized on a scale of 5 and accompanied by accuracy ranges reported in the literature. Scientific references are provided for each criterion to support the evaluations. Random Forest (RF) and Support Vector Machine (SVM) models stand out overall for their high performance, while Naive Bayes (NB) shows more variable performance depending on the criteria.

This figure illustrates the comparative performance of the five supervised classification algorithms (Naive Bayes, Decision Tree, Random Forest, SVM, and KNN) based on the average scores assigned to each according to the five dimensions defined in the study: technical performance, computational efficiency, ease of use, data compatibility, and practical applicability. The colors of the bars represent the different dimensions of analysis (e.g., yellow = performance, orange = efficiency, red = ease of use, blue = data compatibility, etc.). Each score ranges from 1 (low) to 5 (excellent). We can see that Random Forest and SVM achieve the highest scores overall, while Naive Bayes and KNN show more mixed results depending on the criteria.

TABLE I. Comparison of Algorithms Based on Technical Performance Criteria

Algorithm	Accuracy (%) Yellow (Y)	Robustness Orange(O)	Tendency to overfitting Red(R)	Generalizability Blue (B)
NB	3/5 45 – 98 [25][26][27][28][29]	2/5 [40]	4/5 [45]	3/5 [9]
DT	3/5 64 – 95 [25][30][31][32][33]	3/5 [41]	2/5 [46]	3/5 [14]
RF	5/5 71 – 98 [30][31][33][34][35]	5/5 [42]	4/5 [41]	5/5 [49]
KNN	3/5 64 – 98 [26][36][36][38][39]	2/5 [43]	2/5 [47]	4/5 [50]
SVM	5/5 70 – 97 [27][28][29][35][38]	4/5 [44]	4/5 [48]	5/5 [9]

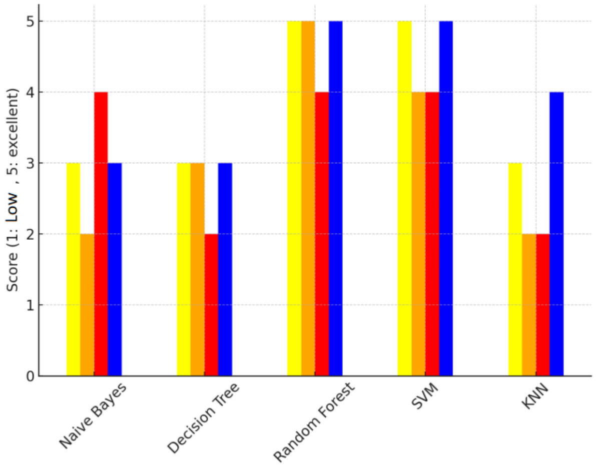

Fig. 2. Comparison of algorithms based on technical performance criteria.

2) Computational efficiency: This table compares the five algorithms studied according to four key aspects of computational efficiency. Each criterion is evaluated on a scale of 1 to 5. The results highlight the superiority of Naive Bayes across all criteria, with perfect scores (5/5), while Random Forest and SVM show compromises between accuracy and computational cost. The bibliographic references associated with each score are also indicated to ensure the traceability of the evaluations.

TABLE II. COMPARISON OF ALGORITHMS BASED ON COMPUTATIONAL EFFICIENCY

Algorithm	Time for training (Y)	Time for prediction (B)	Memory requirements (O)	Algorithmic complexity (R)
NB	5/5 [45]	5/5 [3]	5/5 [2]	5/5 [45]
DT	4/5 [47]	4/5 [47]	4/5 [2]	4/5 [41]
RF	2/5 [49]	3/5 [41]	3/5 [42]	2/5 [49]
KNN	3/5 [51]	1/5 [43]	2/5 [43]	3/5 [51]
SVM	2/5 [52]	3/5 [48]	3/5 [48]	2/5 [52]

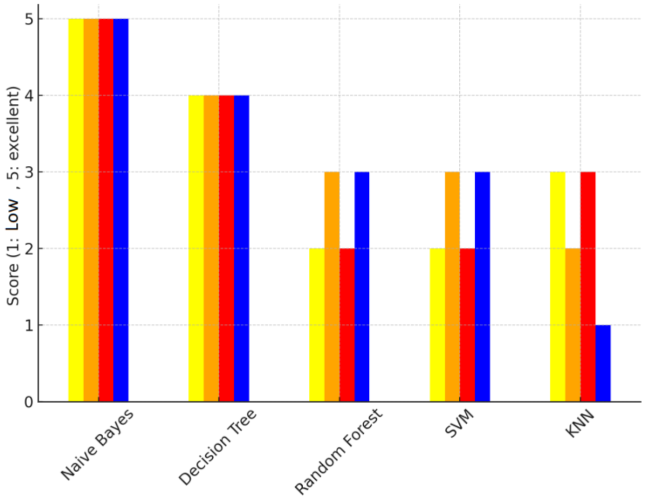

Fig. 3. Comparison of algorithms based on computational efficiency.

This figure provides a comparative visual representation of the five algorithms studied in Table II, according to the following criteria: Time for training (Yellow), Memory requirements (Orange), Algorithmic complexity (Red), and Time for prediction (Blue). Each criterion is rated on a scale of 1 to 5.

3) Practicality and ease of use: This table evaluates the five supervised classification algorithms (Naive Bayes, Decision Tree, Random Forest, k-NN, SVM) according to four practical criteria: ease of implementation, interpretability, level of expertise required, and availability of documentation and associated libraries. Scores are rated on a scale of 1 to 5.

A simplified visualization of the results in Table III, this figure compares the performance of the five algorithms according to four criteria: Ease of implementation (yellow), Interpretability (orange), Level of expertise required (red), and Documentation and libraries available (blue). Scores are given on a scale of 5 to allow for quick comparative reading.

4) Data compatibility: This table evaluates the five algorithms according to four compatibility sub-criteria: Adaptability to multimodal data, Management of missing data, Dimensionality reduction, and Adaptability to unbalanced data. Each criterion is scored out of 5, and the associated bibliographic references are indicated in brackets. These scores illustrate the extent to which each model is able to adapt to the diversity and imperfections of the data used in supervised classification.

TABLE III. COMPARISON OF ALGORITHMS BASED ON PRACTICALITY AND EASE OF USE

Algorithm	Ease of implementation (Y)	Interpretability (O)	Level of expertise require (R)	Documentation and libraries available (B)
NB	5/5 [45]	3/5 [3]	5/5 [45]	5/5 [9]
DT	4/5 [51]	5/5 [53]	4/5 [48]	5/5 [23]
RF	3/5 [41]	2/5 [49]	3/5 [52]	5/5 [9]
KNN	5/5 [51]	2/5 [43]	5/5 [51]	5/5 [23]
SVM	2/5 [48]	2/5 [54]	2/5 [48]	5/5 [9]

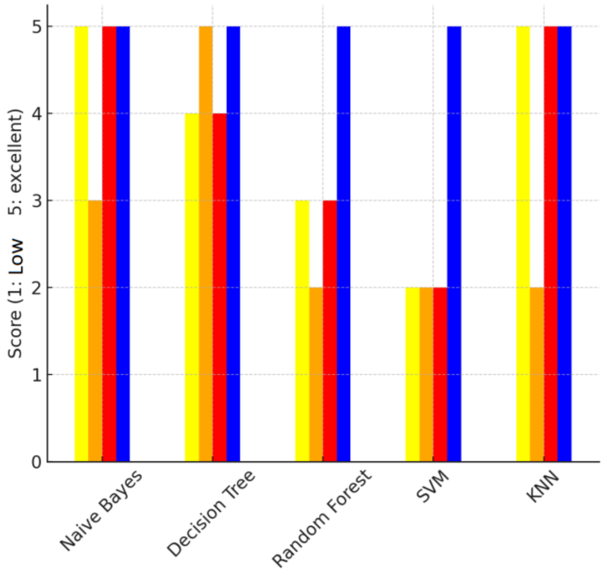

Fig. 4. Comparison of algorithms based on practicality and ease of use.

TABLE IV. COMPARISON OF ALGORITHMS BASED ON COMPATIBILITY WITH DATA

Algorithm	Adaptability to multimodal data (Y)	Handling of missing data (O)	Reduction of dimensionality (R)	Adaptability to unbalanced data (B)
NB	3/5 [55]	3/5 [18]	4/5 [24]	2/5 [55]
DT	4/5 [47]	4/5 [41]	3/5 [47]	3/5 [54]
RF	5/5 [49]	5/5 [42]	4/5 [49]	4/5 [49]
KNN	3/5 [51]	2/5 [43]	3/5 [51]	2/5 [43]
SVM	4/5 [54]	2/5 [48]	5/5 [48]	4/5 [52]

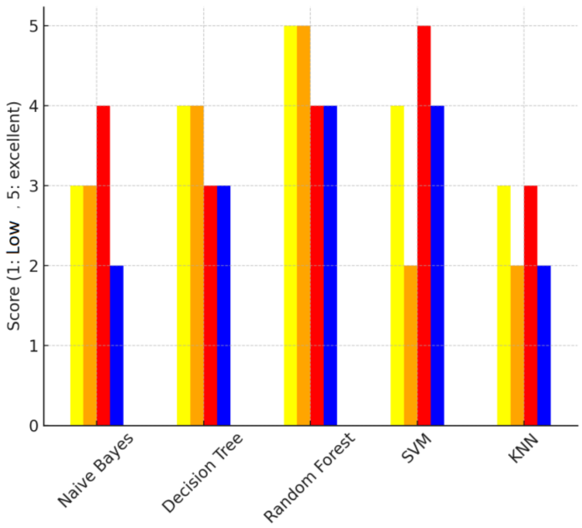

Fig. 5. Comparison of algorithms based on compatibility with data.

Summary graph based on data from Table IV, showing the scores assigned to algorithms on four criteria: Adaptability to multimodal data (yellow), Handling of missing data (orange), Reduction of dimensionality (red), and Adaptability to unbalanced data (blue). This visualization facilitates comparative interpretation between models.

5) Applicability and practical impact: This table presents a comparative evaluation of five supervised classification algorithms—Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (k-NN) [37], and Support Vector Machine (SVM)—based on four criteria related to their applicability in real-world contexts. The criteria studied include: (1) diversity of application domains, (2) impact on practical decision-making, (3) degree of integration into existing systems, and (4) implementation cost. Each criterion is rated on a scale of 1 to 5, where 5 indicates the best performance. The scores are accompanied by bibliographic references [45], [2], [14], etc., attesting to the origin of the evaluations. This summary aims to guide practitioners in choosing an algorithm based on concrete constraints, such as ease of implementation, compatibility with existing systems, or decision-making scope. It thus complements purely technical or statistical comparisons by integrating the dimension of operational applicability.

This visualization illustrates the comparative scores from Table V for the four criteria selected: Application domains (yellow), Impact on practical decisions (orange), Integration into existing systems (red), and Implementation cost (blue). It highlights the relative strengths of each algorithm through a clear representation.

B. Discussion

Quantitative analysis of supervised classification algorithms reveals clear but complementary differences in their technical, operational, and practical properties. These results can be interpreted in several ways depending on the deployment objectives and context of use.

TABLE V. COMPARISON OF ALGORITHMS ACCORDING TO APPLICABILITY AND PRACTICAL IMPACT

Algorithm	Application domains (Y)	Impact on practical decisions (O)	Integration into existing systems (R)	Implementation cost (B)
NB	4/5 [45]	3/5 [2]	4/5 [45]	5/5 [9]
DT	4/5 [41]	4/5 [14]	4/5 [48]	4/5 [41]
RF	5/5 [49]	5/5 [49]	4/5 [42]	3/5 [49]
KNN	3/5 [43]	3/5 [51]	3/5 [51]	5/5 [51]
SVM	5/5 [56]	4/5 [54]	3/5 [52]	3/5 [48]

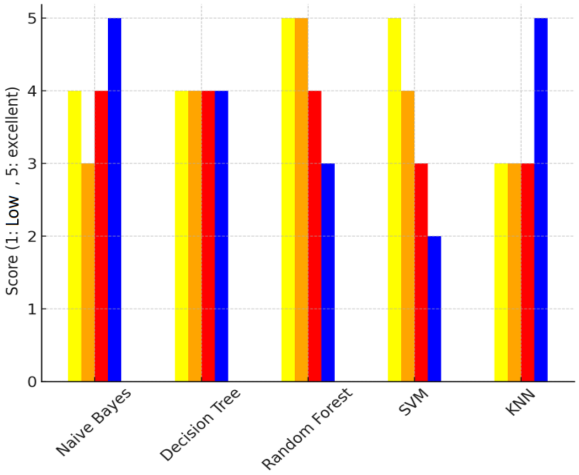

Fig. 6. Comparison of algorithms according to applicability and practical impact.

The Random Forest and SVM algorithms clearly stand out in terms of accuracy, generalization ability, and robustness (Table I, Fig. 2). Their high scores confirm their ability to handle complex problems requiring reliable prediction, such as in medical imaging or cybersecurity. Conversely, although Naive Bayes has an elegant probabilistic structure, its simplification of inter-variable dependencies limits its overall accuracy, particularly in highly correlated contexts. Decision Tree shows moderate performance, while KNN suffers from high sensitivity to noise and overfitting.

Naive Bayes and, to a lesser extent, Decision Tree, are notable for their computational lightness (Table II, Fig. 3). These methods are appropriate in real-time processing contexts or on machines with limited resources. In contrast, Random Forest and SVM, although powerful, have a significant computational cost, both in training and prediction. KNN, although without a learning phase, is slowed down at the time of classification due to the need to scan the entire dataset.

Naive Bayes and KNN stand out as the most accessible, requiring little expertise and being very easy to implement (Table III, Fig. 4). Decision Tree combines this accessibility with excellent interpretability, making it a popular model for decision support systems. On the other hand, SVM and Random Forest are often perceived as “black boxes” requiring more sophisticated configuration and increased expertise, which can limit their adoption in constrained industrial contexts.

The results show that Random Forest is the most versatile, effectively handling missing data, class imbalances, and dimensionality reduction. SVM follows closely behind, provided rigorous preprocessing is performed (Table IV, Fig. 5). Naive Bayes and KNN, on the other hand, are sensitive to data quality and structure. The former assumes conditional independence of variables, while the latter suffers from the effects of dimensionality and noise.

Random Forest and Decision Tree are very attractive from an application standpoint, with broad sectoral adaptability and a concrete impact on decision-making (Table V, Fig. 6). Naive Bayes is also valued for its low cost and ease of integration. On the other hand, SVM remains more expensive to implement, although it offers high performance. Finally, KNN remains unpopular in production environments due to its limited scalability.

C. Limitations of the Results

Although this comparative review provides an in-depth summary of the literature, several limitations should be noted:

1) Lack of empirical validation: The scores are based on bibliographic analyses and do not reflect direct experiments on specific datasets. An empirical study would have allowed for more contextual calibration of the assessments.

2) Variability of application contexts: Algorithm performance can vary significantly depending on the domain (healthcare, finance, language processing, etc.). As this study is general in nature, it does not capture all the nuances specific to each sector.

3) Implementation changes: The performance and practicality of algorithms can be influenced by the software libraries used, which evolve rapidly. Results may therefore change in the short term.

4) Equal weighting of criteria: Aggregating scores assumes uniform weighting of the 20 criteria, which does not necessarily reflect the actual importance of each criterion depending on the context of use.

V. Conclusion

This comparative study, based on an in-depth literature review, evaluated five major supervised classification algorithms—Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN)—across twenty criteria divided into five dimensions: technical performance, computational efficiency, practicality, data compatibility, and applicability.

The results show that Random Forest offers an excellent compromise between accuracy, robustness, and applicability, making it a preferred choice for high-stakes use cases. Naive Bayes, on the other hand, stands out for its simplicity, speed, and low implementation cost, making it ideal for resource-constrained environments. Decision Tree occupies a middle ground, combining interpretability and good performance. On the other hand, SVM, despite its remarkable technical performance, remains difficult to implement and integrate into real systems. Finally, KNN has significant limitations in terms of scalability and robustness, although it remains useful for simple and exploratory cases.

This study highlights the importance of a contextualized multi-criteria assessment in the choice of classification algorithms. No model is universally applicable; each algorithm has strengths and weaknesses that must be weighed according to the specific constraints of the problem, the domain, and the technical environment.

Future work could expand on this bibliographic analysis through rigorous experimental validation, applying the five classification algorithms to real-world datasets from various fields such as education, healthcare, and finance. Such an extension would allow theoretical observations to be compared with empirical performance and recommendations for use to be refined according to the application context. In addition, a promising avenue lies in integrating recent advances in AutoML to automate the selection, configuration, and optimization of models based on the nature of the data. Exploring hybrid learning techniques (e.g., combining multiple algorithms via ensembles or mixed architectures) could also improve robustness and predictive performance. Finally, the use of supervised deep learning approaches, in particular deep neural models or pre-trained neural networks (e.g., BERT, LSTM, CNN), would represent a step forward in the processing of more complex data (text, images, etc.). A complementary benchmark could thus be considered to compare classical models and neural architectures on broader common criteria (training time, ability to generalize, sensitivity to imbalances, etc.).

References

[1] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[2] Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[3] Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. Communications of the ACM, 55(10), 78-87.

[4] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

[5] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. The MIT Press.

[6] Mondal, B. (2020). Artificial Intelligence: State of the Art. In: Balas, V., Kumar, R., Srivastava, R. (eds), Recent Trends and Advances in Artificial Intelligence and Internet of Things. Intelligent Systems Reference Library, vol 172. Springer, Cham.

[7] Erraissi, A., Belangour, A., Tragha, A. (2017). A Comparative Study of Hadoop-based Big Data Architectures. International Journal of Web Applications, 9(4), 129-137.

[8] Domingos, P., Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Machine Learning, 29(2-3), 103-130.

[9] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

[10] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

[11] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). Classification and Regression Trees. Chapman & Hall/CRC.

[12] Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106.

[13] Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. Proceedings of the International Workshop on Multiple Classifier Systems.

[14] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[15] Cover, T. M., Hart, P. E. (1967). Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[16] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer.

[17] Schölkopf, B., Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

[18] Little, R. J. A., Rubin, D. B. (2020). Statistical Analysis with Missing Data. Wiley.

[19] Lipton, Z. C. (2016). The Mythos of Model Interpretability. arXiv preprint arXiv:1606.03490.

[20] He, H., Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.

[21] Zhou, Z. H., Feng, J. (2020). Deep Forest: Towards an Alternative to Deep Neural Networks. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI).

[22] Caruana, R., Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. Proceedings of the 23rd International Conference on Machine Learning (ICML).

[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[24] Guyon, I., Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157-1182.

[25] Hafidz, H., & Fakhriza, M. (2024). Comparison of Naive Bayes Algorithms and Decision Tree for Classifying Hero Fighter Items in the Mobile Legends. Journal of Applied Science, Engineering, Technology, and Education, 6(2), 127-142.

[26] Buhori, K. (2024). Comparative Analysis of KNN and Neavy Bayes Algorithms in Socio-Economic Data Classification in Indonesia. Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi, 15(2), 222–232.

[27] Sari, P. R., Indah, D. R., Rasywir, E., Firdaus, Mgs. A., & Athalina, G. (2024). Comparison of Naive Bayes and SVM Algorithms for Sentiment Analysis of PUBG Mobile on Google Play Store. Jurnal Sistem Informasi, 13(6), 2767.

[28] Azhar, A., Masruroh, S. U., Wardhani, L. K., & Okfalisa, O. (2023). Performance comparison of the Naive Bayes algorithm and the k-NN lexicon approach on Twitter media sentiment analysis. Science, Technology, and Communication Journal, 3(2), 35-40.

[29] Wibawa, A. P., Kurniawan, A. C., Murti, D. M. P., Adiperkasa, R. P., Putra, S. M., Kurniawan, S. A., & Nugraha, Y. R. (2019). Naïve Bayes Classifier for Journal Quartile Classification. International Journal of Recent Contributions from Engineering, Science & IT (iJES), 7(2), 91–99.

[30] Kinasih, A., Handayani, A., Ardiansah, J., Damanhuri, N. (2024). Comparative analysis of decision tree and random forest classifiers for structured data classification in machine learning. Science in Information Technology Letters, 5, 13-24.

[31] Esmaily, H., Tayefi, M., Doosti, H., Ghayour-Mobarhan, M., Nezami, H., Amirabadizadeh, A. (2018). A Comparison between Decision Tree and Random Forest in Determining the Risk Factors Associated with Type 2 Diabetes. J Res Health Sci, 18(2), 412.

[32] Mohammadi-Pirouz, Z., Hajian-Tilaki, K., Sadeghi Haddat-Zavareh, M., et al. (2024). Development of decision tree classification algorithms in predicting mortality of COVID-19 patients. Int J Emerg Med, 17, 126.

[33] Mitschek, M., Esquivel, R. (2023). A Comparative Analysis of Decision Tree Classification Algorithms for Blended Learning Analytics in WEKA. Celt, 23, 321-334.

[34] Gangadhar, C., Roy, P., Kumar, R., Ramesh, J., Ravikanth, S., Akhila, N. (2025). Wearable Sensor-based Fall Detection for Elderly Care Using Ensemble Machine Learning Techniques. Measurement Sensors, 101870.

[35] Afuan, L., & Isnanto, R. (2024). Enhanced Fall Detection using Optimized Random Forest Classifier on Wearable Sensor Data. Journal of Applied Data Sciences, 6(1), 213-224.

[36] Madi, S., Baba-Ali, A.R. (2024). A new hybrid incremental learning system for an enhanced KNN algorithm (hoKNN). Evolving Systems, 15, 1001–1019.

[37] Uddin, S., Haque, I., Lu, H., et al. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. Sci Rep, 12, 6256.

[38] Shdefat, A.Y., Mostafa, N., Al-Arnaout, Z., et al. (2024). Optimizing HAR Systems: Comparative Analysis of Enhanced SVM and k-NN Classifiers. Int J Comput Intell Syst, 17, 150.

[39] Çetin, A., Büyüklü, A. (2024). A new approach to K-nearest neighbors distance metrics on sovereign country credit rating. Kuwait Journal of Science, 52(1).

[40] Sanjay, K. D., Arya, C. (2016). Automatic Text Classification in Information Retrieval: A Survey. ICTCS '16, ACM, Article 131.

[41] Goyal, G., Karwal, S., Garg, R. (2023). Performance Comparison of Dysgraphia Detection Algorithms. 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), IEEE.

[42] Kumar, S., Laxkar, D., Adhikari, S., Vijayarajan, V. (2017). Assessment of various supervised learning algorithms using different performance metrics. IOP Conf. Series: Materials Science and Engineering, 263, 042087.

[43] Siraj-Ud-Doulah, M., Alam, M. (2020). Performance Evaluation of Machine Learning Algorithms in Ecological Dataset. Int. J. of Applied Mathematics and Machine Learning.

[44] Chowdhury, M. S. (2023). Comparison of Accuracy and Reliability of Random Forest, SVM, ANN and Maximum Likelihood method in Land use/cover Classification. Environmental Challenges.

[45] Angula, T., Hashiyana, V. (2023). Detection of Structured Query Language Injection Attacks Using Machine Learning Techniques. Int. J. of Computer Science and Information Technology.

[46] Darde, S., Wagh, S. (2024). To Study the Comparative Analysis of Classification Algorithms for Heart Stroke Prediction. Int. J. of Advanced Research in Science, Communication and Technology.

[47] El-Habil, A. (2014). Evaluation of Data Mining Classification Models. IUG Journal of Natural and Engineering Studies, 151–165.

[48] Rodrigues, I., Parayil, A., Shetty, T., Mirza, I. (2020). Use of LDA, KNN, CART, RF, Gaussian NB, SVM to Predict Admission for PG Courses. SSRN.

[49] Das, A., Choudhury, D., Sen, A. (2024). A collaborative empirical analysis on ML-based disease prediction in healthcare. Int. J. of Inf. Technol., 16, 261–270.

[50] Skowronski, L., de Moraes, P.M., de Moraes, M.L.T., et al. (2021). Supervised learning algorithms in the classification of plant populations. Braz. J. Bot, 44, 371–379.

[51] Lakshmi, V. (2025). A Review on K nearest Neighbour Classification Technique in ML. Int. J. of Scientific Research in Science, Engineering and Technology.

[52] Bhatnagar, V., Poonia, R.C. (2021). Comparative Study of Supervised ML Algorithms for Healthcare Dataset Using Orange. In: Intelligent Learning for Computer Vision, Springer.

[53] Walji, K., Erraissi, A., Zakrani, A., Banane, M. (2025). Comparative Performance Evaluation of ML Algorithms in Sentiment Analysis. In: ICAISE 2024, Springer.

[54] Kalcheva, N. A., Karova, M. N. (2022). A Comparison of ML Classification Algorithms for English Works and their Translations. 57th Int. Conf. on Information, Communication and Energy Systems and Technologies (ICEST), IEEE.

[55] Dwivedi, S., Arya, C. (2016). Automatic Text Classification in Information Retrieval: A Survey. ICTCS '16, ACM.

[56] Sen, P.C., Hajra, M., Ghosh, M. (2020). Supervised Classification Algorithms in ML: A Survey and Review. In: Emerging Technology in Modelling and Graphics, Springer.