Evaluating the Impact of Machine Learning Platforms on Cancer Classification Model Performance: A Cross-Platform Comparative Study

Adedayo Seun Olowolayemo
School of Engineering, Technology, and Design
Canterbury Christ Church University (CCCU)
Canterbury, UK
a.olowolayemo502@canterbury.ac.uk

Amina Souag
School of Engineering, Technology, and Design
Canterbury Christ Church University (CCCU)
Canterbury, UK
amina.souag@canterbury.ac.uk

Konstantinos Sirlantzis
School of Engineering, Technology, and Design
Canterbury Christ Church University (CCCU)
Canterbury, UK
Konstantinos.sirlantzis@canterbury.ac.uk

***Abstract* — Machine Learning techniques have become pivotal in advancing predictive models for early cancer detection, addressing the growing need for improved diagnostic efficiency. However, the role of implementation platforms in influencing model performance remains underexplored, even as variations in performance with the same dataset raise questions about platform choice. This study evaluates the impact of three ML implementation tools, the Scikit-learn, KNIME, and MATLAB on the performance of four classification algorithms: Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. Using the publicly available Wisconsin Diagnostic Breast Cancer dataset, these algorithms were implemented under default configurations and compared across key metrics: accuracy, recall, precision, and F1-score. Results revealed significant platform-dependent variations: Scikit-learn achieved consistently higher recall, particularly for Random Forest and Gradient Boosting, making it more effective at minimising false negatives critical in cancer diagnosis. MATLAB demonstrated superior precision, especially for Random Forest and Gradient Boosting, indicating potential in reducing false positives. KNIME, while effective in specific contexts, underperformed in recall and precision, raising concerns in scenarios requiring high sensitivity and specificity. These findings underscore the importance of platform selection based on predictive task requirements, especially in healthcare, where balancing false positives and false negatives is crucial. The study provides actionable insights for selecting ML platforms to enhance diagnostic accuracy in cancer classification tasks, with source code and data fully accessible through a public GitHub repository.**

Keywords - Cancer; Machine Learning; Python Scikit-learn; KNIME; MATLAB; Wisconsin Diagnostic Breast Cancer.

I. INTRODUCTION

Cancer remains a significant global health threat, causing nearly 10 million deaths in 2020 approximately one in six deaths globally underscoring its devastating impact and the urgent need for more effective prevention, early detection, and treatment strategies [1][2][3][4]. According to the World Health Organization (WHO), the disease affects individuals of all ages, including about 400,000 children each year. Notably, breast, lung, and colorectal cancers had the highest incidence rates, with lung cancer leading in mortality, followed by colorectal, liver, stomach, and breast cancers, as shown in Fig. 1. This figure depicts the distribution of new cancer cases and cancer-related deaths by type for 2020, highlighting the global burden of specific cancers and emphasizing the importance of early diagnosis and screening to reduce mortality and mitigate the far-reaching impacts of the disease [5].

Cancer arises from the uncontrolled division of cells, resulting in the formation of *tumours* that are classified as either *malignant* or *benign*. Malignant tumours are of particular concern due to their ability to invade surrounding tissues and spread to other parts of the body through metastasis [6]. This invasive behaviour complicates treatment, often requiring a combination of surgery, chemotherapy, and radiation [7]. Once metastasis occurs, malignant cells can establish secondary tumours in distant organs, such as the lungs, brain, or liver, further increasing the complexity of treatment and affecting patient prognosis [8].

In contrast, benign tumours, though also characterised by abnormal cell growth, remain localised and do not spread to other areas of the body. While generally less harmful, they can still pose risks depending on their size and location, particularly if they press on critical organs or tissues [9]. Treatment for benign tumours is typically less aggressive, though surgical removal may be necessary in cases where they cause discomfort or complications.

The distinction between malignant and benign tumours is crucial in understanding cancer progression, as well as the urgency and approach to treatment. This disease's complexity spans multiple organs including the breast,

kidneys, brain, lungs, prostate, ovaries, and skin, hence posing significant challenges for healthcare professionals. Despite advances in treatment, timely diagnosis remains critical; delays can lead to advanced stages of cancer that are more difficult to treat and are often associated with higher mortality rates.

Scientists are increasingly directing significant resources toward revolutionising the cancer diagnostic process, recognizing that early and accurate diagnosis can drastically improve patient outcomes. In this endeavour, Artificial Intelligence (AI) has emerged as a key player, demonstrating its potential across various domains, and now offering promising solutions in healthcare [10]. What sets AI apart in the medical field, particularly in cancer diagnosis, is its capacity to process and analyse vast amounts of complex data at speeds and scales that far exceed human capabilities.

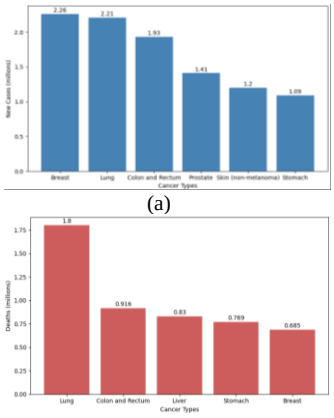

Figure 1. Chart of (a) New cancer cases by cancer type and (b) Cancer deaths by cancer type in 2020 [5]

Machine Learning (ML), a subset of AI, is not only enhancing efficiency but also transforming the nature of medical research. This transformation is evident in numerous studies [11][12][13], where AI techniques have been employed for the classification and prediction of cancer, as well as patient survival outcomes. Due to their ability to learn from data, ML algorithms trained on datasets can identify patterns and markers often imperceptible to human observers. [14]. This has led to breakthroughs in diagnostic precision, allowing for more accurate differentiation between diseases, including cancerous and non-cancerous conditions. Beyond diagnosis, ML is being leveraged to improve prognostic accuracy by predicting disease progression and response to treatment [15], helping health professionals make more informed decisions tailored to individual patients.

As ML continues to evolve, its impact extends beyond speed and precision; it has the potential to reshape the entire framework of cancer care. By integrating AI tools into clinical workflows, the hope is not only to expedite the diagnostic process but to also develop a more personalised, data-driven approach to treatment, where ML models help guide therapeutic choices with unprecedented accuracy. This shift represents a fundamental transformation in the healthcare industry, where the convergence of data science and medical practice could lead to faster, more reliable diagnoses and ultimately, improved survival rates for cancer patients.

By harnessing advanced computational techniques, ML algorithms ranging from Logistic Regression (LR), Decision Trees (DT), Random Forest (RF), and Gradient Boosting (GBoost), among several others used for cancer diagnosis, extract insights from intricate medical data used in revolutionising clinical decision-making and improving patient outcomes from pinpointing diseases through image analysis [16] to forecasting patient responses to therapies.
However, a critical aspect that we found to be underexplored is the impact of implementation platforms on which the algorithms are trained, and models are developed, such as Python Scikit-learn, KNIME, and MATLAB on the performance of these algorithms. Therefore, understanding the nuanced influence of implementation platforms on ML algorithms is pivotal.

Against this backdrop, this study employed supervised learning to train models on Wisconsin Diagnostics Breast Cancer (WDBC) dataset [17] to evaluate the performance metrics of ML algorithms including accuracy, precision, recall, and F1-Score. The focus was on understanding the nuanced relationship between implementation platforms and the efficacy of these algorithms, emphasizing the potential impact of platform choice on algorithm behavior and highlighting the need to discern these disparities.
To achieve this, the study addresses two pivotal inquiries:

(1) It seeks to answer whether the choice of the implementation platform impacts the performance of ML algorithms in cancer data classification, and

(2) identifies which of the selected algorithms performed best in cancer dataset binary classification task.

This study delves into the complex interaction between ML algorithms, the platforms on which they are implemented, and the significance of the features within the dataset. Rather than focusing on optimising hyperparameters, the research aims to unearth deeper, more fundamental insights into how platform-specific factors influence model outcomes, including accuracy, efficiency, and predictive robustness. By utilizing the WDBC dataset, the study trains ML models to classify tumours as malignant or benign, a task critical for early cancer diagnosis. The focus on platform comparison allows for an exploration of how the underlying architecture and computational efficiency of different ML platforms can affect model performance, independent of tuning techniques.

This approach highlights a broader issue in ML research, how the choice of implementation tools can shape results beyond mere algorithm selection or dataset quality.
In analysing platform impacts on diagnostic accuracy, this study offers valuable contributions to developing more reliable and consistent ML-driven diagnostic systems, ensuring that performance improvements in cancer detection are not limited by the tools used to implement them. Ultimately, these findings provide a roadmap for more informed choices when developing ML models for medical applications, paving the way for advancements that can directly enhance patient outcomes.

The rest of this paper is organised as follows: Section II reviews related works, exploring the use of machine learning in cancer research. It examines studies that have applied ML algorithms, focusing on their implementation methods, train-test split strategies, performance evaluation metrics, dataset sources, and platforms utilised. Section III outlines the methodology adopted in this study, providing a detailed account of data collection, preprocessing, feature selection, and the implementation of selected ML models. Section IV presents the results, supported by an in-depth discussion of their implications. Finally, Section V concludes the paper by summarizing the findings and proposing directions for future research.

II. Related work

Researchers have explored and reported the use of various supervised ML algorithms in different areas of human health and medical fields. Some previous studies reviewed are briefly discussed below.

A. ML in Cancer Research

ML is reshaping the landscape of cancer research by offering powerful tools to improve key areas such as cancer classification and treatment outcome prediction. With its ability to process and analyse large amounts of data more efficiently, ML has allowed researchers to uncover patterns and insights that were previously out of reach, leading to more precise diagnoses and predictions. This section delves into a range of studies that demonstrate the practical benefits of ML in cancer research, shedding light on how it has enhanced diagnostic accuracy and predictive modeling. Despite these strides, there is still room to explore and fine-tune its applications. Through this review, we aim to highlight both the significant progress made and the opportunities for further development, emphasizing the potential for ML to drive even more impactful breakthroughs in cancer treatment and care.

Michael et al. in [18] tested five ML classification algorithms on 912 breast ultrasound images and found that Light Gradient Boosting Machine (LightGBM), the algorithm proposed in their work, which has an accuracy of 99.86%, outperformed other algorithms including K-Nearest Neighbour (KNN), and RF in binary classification of cancerous cells as either malignant or benign. Similarly, Ara et al. in [19] used a ML techniques to develop model for classifying cancer cells into two main categories. Kumar et al. in [20] on the other hand focused on using ML ensemble techniques for breast cancer detection and classification.
Their Optimized Stacking Ensemble Learning (OSEL) model showed a higher accuracy in performing the task than other ensemble ML techniques, such as Stochastic GBoost and XGBoost tested in their research. Ebrahim et al. [21] tested eight predictive algorithms on the National Cancer Institute dataset to identify which algorithm would predict cancer cell more accurately.

B. Selection of Algorithm

In cancer research involving ML, the selection of algorithms is a critical factor that can influence model performance, especially when applied to widely used datasets like the WDBC dataset. Numerous studies have utilised various ML algorithms for tasks such as classification, prediction, and diagnosis. This section reviews the algorithms commonly selected in existing literature, with a particular focus on those used in cancer research. While the current study aims to investigate how implementation tools may impact model performance under default settings, the literature at this stage primarily explores algorithm selection based on factors such as accuracy, ease of use, and compatibility with specific datasets. By examining these studies, we aim to uncover potential reasons behind the popularity of certain algorithms in the context of cancer classification, which can serve as a foundation for understanding the broader landscape of ML applications in healthcare.

LR, a linear model is a powerful predictive analysis tool that is especially useful for binary classification [22]. Zhu et al. in [22] experimented with improved LR in the classification of binary variable and independent variables to predict diabetes. Rahman et al. [23] examined six ML algorithms for predicting Chronic Liver Disease (CLD) and found the LR algorithm to be the most effective in predicting CLD based on the selected features.

Likewise, Tree based algorithms including DT, RF and GBoost are widely researched with the intent of harnessing their strengths particularly in performing classification tasks. Decision Trees (DT) provide a simple and interpretable framework upon which more advanced tree-based models, like Random Forests (RF) and Gradient Boosting (GBoost), are built, partitioning feature spaces into hierarchical branches to effectively capture non-linear relationships and feature interactions, enabling straightforward visualisation of decision-making processes. Moving beyond individual trees, RF combines multiple DTs through ensemble techniques, mitigating overfitting and increasing predictive accuracy [24]. By combining varied perspectives from individual trees, RF provides robust generalization and robustness to noisy data.

By extension, the GBoost algorithm, a more advanced method, embraces an iterative refinement to enhance predictive performance and in particular, Gradient Boosting Trees, such as XGBoost employ sequential tree fitting to target the residuals of prior iterations, systematically improving model predictions. These algorithms perform well in modeling complex relationships, accommodating non-linearities, and excelling in predictive accuracy across domains [25][26]. These characteristics formed the basis on which we selected the algorithms in our study.

C. Train-Test Split

The train-test split is a widely used method in ML, essential for assessing and comparing different algorithms or model configurations. By partitioning a dataset into two segments with one for training and one for testing, it ensures that models are evaluated consistently across the same testing subset. This process provides an unbiased framework for determining how well each model performs, free from the influence of the training data. Metrics such as accuracy, precision, recall, and F1-score, calculated from the test data, offer valuable insights into a model's potential performance in practical, real-world applications. More than just facilitating

model training, the train-test split underscores the necessity of rigorous validation to guarantee that the model's predictions are not only accurate but also reliable when deployed.

For evaluation, datasets used in various studies are split into different proportions using the larger proportion to train algorithms while the smaller proportion is used to test at the inference stage of model development. In [22], the authors assessed the performance of some classical and deep learning algorithms used to predict breast cancer, including DT, LR, KNN, Support Vector Machine (SVM), Recurrent Neural Networks (RNN) and Ensemble Learning. They used Train/Test split of 70:30 and 90:10. DT and Ensemble methods showed higher accuracy both before and after feature selection. Whereas DT did not perform optimally in predicting Kidney Cancer Lung Metastasis, as reported by [27], when trained with 52,222 records from the Surveillance, Epidemiology, and End Results (SEER) database and 492 hospital patient records with Train/Test split of 70:30 returning accuracy of 82% which is significantly lower than in other studies reviewed.

D. Performance Metrics

Efficient model development and deployment require a thorough evaluation to ensure reliable performance in real-world applications. One of the most essential tools in this process is the confusion matrix, which is a tabular representation that summarises the model's predictions against the actual outcomes, giving a detailed breakdown of a model's predictions [28]. It classifies outcomes into four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) as we have illustrated in Fig. 2. By analysing these classifications, the confusion matrix helps reveal where a model excels and where it has not performed as expected or optimally. This level of detailed insight is particularly important in domains like healthcare, where incorrect predictions can have serious consequences, such as misdiagnoses or missed critical conditions.

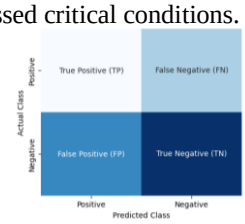

Figure 2. Illustrative Confusion Matrix Table.

Various important performance metrics are obtained from the confusion matrix, each providing a unique perspective for evaluating a model's effectiveness. The most commonly used metric is accuracy, which measures the proportion of correct predictions relative to the total number of predictions. While accuracy provides a broad overview of a model's success, it can be misleading, especially in scenarios with imbalanced datasets.

To address the limitations of accuracy, additional metrics such as precision, recall, and F1-score become crucial. Precision, which measures the proportion of correct positive predictions out of all positive predictions, is particularly relevant when the cost of false positives is high. In medical settings, a false positive incorrectly identifying a healthy individual as sick can lead to unnecessary treatments, anxiety, and strain on healthcare resources. Thus, high precision is essential to minimise the risk of falsely diagnosing healthy patients.

The performance metrics derived from the confusion matrix are computed based on equations (1-4) below.

$$Precision = TP / (TP + FP) \quad (1)$$

$$Recall = TP / (TP + FN) \quad (2)$$

$$F1\ Score = 2 * (Precision * Recall) / (Precision + Recall) \quad (3)$$

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

Conversely, recall focuses on the model's ability to capture all actual positive cases, which is especially important in ensuring that no dangerous conditions are missed. In healthcare, missing a diagnosis, such as failing to detect cancer, can have devastating consequences. Therefore, high recall ensures that all true positive cases are identified, reducing the risk of underdiagnosis in critical conditions.

Balancing precision and recall is where the F1-score proves invaluable. The F1-score is the harmonic mean of precision and recall, offering a balanced evaluation of a model's ability to minimise both false positives and false negatives. This is particularly useful in datasets with class imbalances, where optimising for either precision or recall alone may not provide an accurate reflection of the model's true performance. In medical diagnostics, where both overdiagnosis (false positives) and underdiagnosis (false negatives) can have significant consequences, the F1-score serves as a comprehensive measure that helps to ensure models perform well across the spectrum of possible outcomes.

Ultimately, the confusion matrix and its associated metrics, (accuracy, precision, recall, and F1-score) offer a robust framework for assessing ML models, particularly in sensitive fields like healthcare. These metrics provide a deeper understanding of how models perform in various scenarios, ensuring they are not only accurate but also effective in minimising the risks associated with false predictions. By going beyond basic accuracy, this approach helps build trust in model deployment, ensuring that ML systems can reliably make critical decisions in complex, real-world environments.

Accuracy measures the proportion of correctly predicted instances in the dataset, providing a general overview of predictive success. Precision focuses on correctly predicted positive cases, which is crucial in scenarios like medical diagnoses where false positives can lead to adverse consequences. Recall assesses the model's ability to identify all true positive cases, essential for avoiding missed diagnoses in critical medical conditions. The F1-score balances precision and recall, offering a nuanced evaluation that is particularly useful for datasets with class imbalances. These four metrics collectively provide a comprehensive assessment of a model's performance.

E. Datasets

Data quality is fundamental in ML, shaping model development and real-world utility. The WDBC [17] has been pivotal in healthcare, especially for binary tumour classification, crucial in timely cancer detection and treatment planning. While a number of studies like [17][18][19] employed smaller, open-source WDBC datasets (typically fewer than 600 records and 30 features), other studies in [22] and [15] diverged. For example, [22] used a substantial dataset from the National Cancer Institute (NIH) containing 1.7 million records and 210 features. Despite its size, dataset quality, marked by precision and representativeness, significantly influences outcomes. Smaller datasets with these qualities outperform larger, noisier ones. This distinction is evident in accuracy rates, with open-source datasets achieving 99.12%, 99.67%, and 100%, compared to the model in [22] with a lower accuracy of 97.4%.

F. Implementation Platform

KNIME Analytics, a no-code tool known for its user-friendly interface and extensive integration with external tools, has been widely used in ML research, including studies such as [29], which explored cancer incidence among individuals with HIV in Zimbabwe. KNIME's appeal lies in its accessibility, allowing researchers without advanced programming skills to build and implement complex ML models. Meanwhile, Python, particularly with its rich ecosystem and powerful libraries like Scikit-learn, has established itself as a go-to platform for ML. Multiple studies, such as those in [30][31][32] have employed Python for cancer research, leveraging its versatility and the ability to fine-tune models through code.

In addition to KNIME and Python, MATLAB has also been widely used in ML research. Known for its robust computational capabilities, MATLAB offers a range of toolboxes and functions for developing ML algorithms. Its application in cancer diagnosis and classification tasks has been demonstrated across various studies, where it has been employed to build predictive models and evaluate performance across different classification algorithms. All three platforms including KNIME, Python (Scikit-learn), and MATLAB have significant backing from the scientific community. Each platform offers unique strengths, making it important to understand how platform-specific architectures and tools impact ML algorithm performance, especially in sensitive fields like cancer research.

The findings from the literature are summarised in Table I, which provides a comprehensive overview of recent studies utilizing ML techniques in cancer research. The table outlines critical aspects of each study, including the data sources, train-test split ratios, implementation platforms, algorithms employed, and resulting model accuracy. This summary enables a clear comparison of the approaches and outcomes in applying ML to cancer diagnosis and prognosis, offering insights into the varied impacts of different platforms and algorithms on model performance. (*a '–' has been used in the table to indicate instances where the relevant information was not available in the literature*).

TABLE I. COMPARATIVE REVIEW OF SOME STUDIES THAT USED ML TECHNIQUES IN CANCER RESEARCH.

Author, Year	Data Source	No of Records /Features	Train/Test Split	Implementation Platform	Algorithm Type	Model Accuracy
Ara et al. [19] , 2021	UCI	569/30	75:25	-	SVM, LR, KNN, DT, NB, RF	96.5%
Ebrahim et al. [21],2023	National Cancer Institute (NIH),	70,079/107	70:30 &90:10	- Python	DT, LR, VM, LD, ET, KNN	98.7%
Minnoor et al.[24] 2023	UCI	569/30	80:20	-	RF, SVM, DT, MLP, KNN	100%
Yi et al., [27],2023	SEER& Southwest Hospital, China.	52,714 / -	70:30	Python	LR, XGBoost, RF, SVM, ANN, DT	-
Shafique et al.[29],2023	Kaggle	569/30	75:25		RF, VM, GBoost, LR, MLP, KNN	100%
Uddin et al. [30], 2023	UCI	569/30	70:30	Python	SVM, RF, KNN, NB, DT, LR, AB, GBoost, MLP, NCC, VC	98.7 %
Mahesh et al., [33],2022	Kaggle	143/10	70:30	Python	NB, AltDT, RedEPT, RF	98.20%
Zhang et al [34]., 2022	TCGA	604/ -	-	R & Python	RF, SVM, libD3C	99.67%
Aamir et.al.[35], 2022	UCI	569/26	80:20 &70:30	Python & Tensor Flow	RF, GBoost, SVM, ANN, MLP	99.12%
ATEŞ et al. [36] 2021	Kaggle	569/30	70:30	KNIME	NB, DT, MLP	96.5%
Liu, et al. [37]2018	UCI	569/30	75:25	Python	LR	96.5%
Dora et al., 2017 [45]	UCI	569/30	70:30	MATLAB	ELM	94.52%

While numerous studies have demonstrated the effectiveness of machine learning (ML) in cancer diagnosis and classification, a critical gap persists in understanding how the choice of implementation platform influences model performance. Most research has focused on algorithm selection and dataset quality, operating under the assumption of platform independence. This neglects potential disparities introduced by differences in platform architectures, default configurations, and computational efficiencies, which could significantly affect model outcomes and their broader applicability. Addressing this unexplored area forms the crux of our research.

By systematically investigating how different ML implementation platforms including KNIME, Scikit-learn, and MATLAB impact the performance of widely used classification algorithms in cancer diagnostics, we aim to shed light on a critical yet overlooked factor. Our study explores platform-dependent variations across key metrics such as accuracy, recall, precision, and F1-score, offering a novel perspective that underscores the importance of platform choice in high-stakes applications like healthcare.

The practical implications of this research are substantial as understanding how platform-specific characteristics influence model accuracy, efficiency, and scalability enables

more informed decisions in real-world applications where performance optimisation is essential.

By providing actionable insights and a rigorous methodological framework, this study contributes to the broader discourse in ML research, encouraging further consideration of technical environments and fostering advancements that improve diagnostic workflows and patient outcomes. Ultimately, this work fills a vital gap in existing literature and establishes a foundation for optimising ML workflows across diverse computational platforms

In the sections that follow, we delve into the methodology designed to rigorously test this hypothesis, offering a framework that enables a deeper understanding of how platform-specific characteristics may impact model performance across various algorithms. This novel perspective enhances the current discourse in ML, encouraging further consideration of the technical environments in which models are deployed.

III. METHODOLOGY

This study's methodology comprises systematic steps for a comparative analysis of ML algorithms using the WDBC dataset and three implementation platforms. The process as illustrated in Fig. 7 includes data collection, exploration, feature engineering and selection using filtering and RF techniques. The dataset was split into an 80% training set and a 20% test set before model development, ensuring a robust evaluation process.

A. Data Collection and Preprocessing

We selected the publicly available WDBC dataset from the University of California, Irvine (UCI) ML repository [18] because of its origin in medical research, extensive use in breast cancer-related ML studies, and established reputation in the research community. Its real-world applicability makes it a reliable choice for binary classification tasks. The dataset contains 569 instances and 30 attributes, extracted from digitised Breast Mass Fine Needle Aspiration (FNA) specimens. These attributes include measurements such as "radius_mean," "texture_mean," and "perimeter_mean," which represent features of cell nuclei in biopsy images.

The dataset is divided into two classes: benign tumours, comprising 62.7% of the total instances, and malignant tumours, making up the remaining 37.3%. We show in Fig. 3 the proportion of these two classes, highlighting the distribution of benign and malignant cases for further analysis [18].

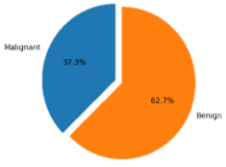

Figure 3. Pie chart showing percentage composition of the class labels as M-malignant and B-benign.

Following the dataset analysis, we conducted a correlation analysis to explore relationships between features, as illustrated in the heatmap in Fig. 4. This step is crucial for feature selection, offering insights into how each feature correlates with the target variable and other features. Identifying multicollinearity when features are highly correlated is essential, as redundant features can complicate the model without enhancing predictive performance. This process helps ensure the model remains efficient and effective.

The correlation analysis serves two purposes: identifying features strongly correlated with the target variable for their predictive potential and detecting pairs of highly correlated features. When features exhibit high correlation (close to ±1), removing one of them helps reduce redundancy and streamline the model without affecting performance.

In this study, where the aim is to investigate the impact of ML implementation platforms on model performance, optimising the feature set before comparing models is crucial. Since models are tested using default platform settings, including only the most relevant features becomes even more important. Retaining irrelevant or redundant features could obscure performance differences between platforms by introducing noise or inflating the models unnecessarily.

The correlation analysis assessed the relationships between features, providing insights into their relevance to the target variable and identifying interdependencies between them. The heatmap in Fig. 4 highlights these correlations, with darker shades indicating stronger relationships and lighter shades indicating weaker ones. This visual helps identify redundant features due to high correlation, guiding better feature selection decisions. Addressing such correlations improves predictive accuracy and reduces the risk of overfitting by ensuring a streamlined feature set. This process enhances the model's overall efficiency and reliability by retaining only the most relevant and independent features.

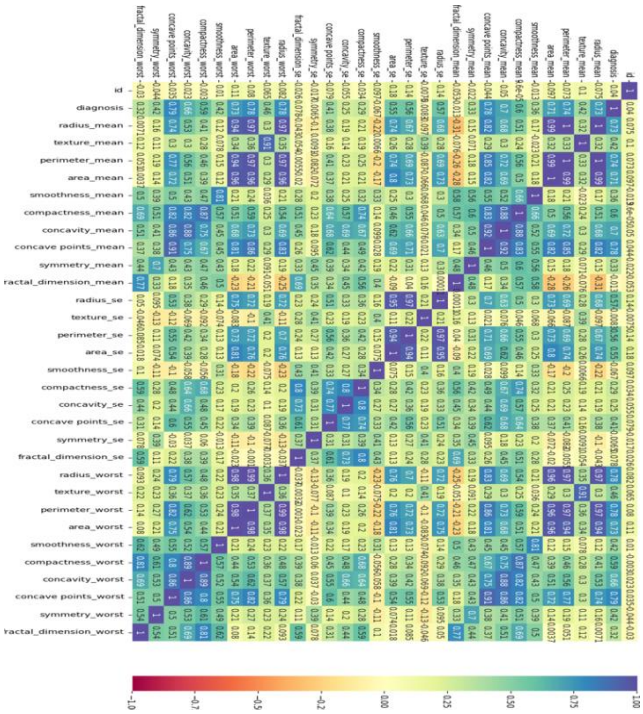

Figure 4. Heatmap of features correlation

In the data preprocessing phase, the dataset was structured into a Python dataframe which we subsequently queried to ascertain the data types and to check for presence of any null or missing values [38]. Table II extracted from our code implementation for Exploratory Data Analysis (EDA), confirms that the WDBC data contains a mix of integer and floating-point values, with no null values identified. Further analysis included detecting outliers using box plots, and the capping method was applied to mitigate their impact, ensuring the dataset's integrity for subsequent analyses. This technique, as presented by [39] involved setting values below the lower whisker to the lower whisker's value and values above the upper whisker to the upper whisker's value, ensuring an unbiased dataset.

TABLE II. WDBC DATASET VARIABLES DATATYPE.

```
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    int32
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int32(1), int64(1)
```

Normalization was achieved through Z-Score Normalization (Standardization). This rescales each feature to a normal distribution with a mean of 0 and a standard deviation of 1 [40][41]. Standardizing features to a common scale is a crucial step in ML to ensure that algorithms do not disproportionately favor features with larger magnitudes. This is especially important for gradient-based models like LR where unscaled features can skew the learning process. By applying z-score standardization (as shown in Equation 5), we normalised each feature to have a mean of zero and a standard deviation of one. This not only enhances the model's ability to learn balanced patterns but also improves the convergence speed during training. This step ultimately helps in improving fairness, accuracy, and overall model performance across a variety of ML algorithms [41].

$$Z=(x-\mu)/\sigma. \quad (5)$$

where z is the scaled value of the feature,
x is the original value of the feature,
μ is the mean value of the feature, and
σ is the standard deviation of the feature.

B. *Feature Selection*

In ML studies, selecting the most informative features is a critical step in optimising model performance. Among the many techniques available for feature selection, Spearman's rank correlation is a popular choice for identifying relevant features based on their effectiveness in handling datasets where relationships between variables and the target are not strictly linear, making it valuable in a wide range of ML tasks. By ranking features according to their correlation with the target, Spearman correlation helps filter out less important features, ultimately improving the model's accuracy and efficiency.

In this study, we implemented a two-step feature selection process utilizing both the Filter Method and the Tree-Based Method. Initially, the Filter Method applied Spearman rank correlation to evaluate the features based on their correlation coefficients with the target variable. Features with correlation coefficients ≤ 0.5 were deemed insignificant and removed, following the guidelines established in previous works by [44]. This threshold-based approach resulted in the selection of 15 out of the 30 original features, which were considered sufficiently relevant for further analysis.

Spearman's rank correlation, being a non-parametric measure, was used here because it can handle monotonic relationships without assuming a linear connection between features and the target variable.

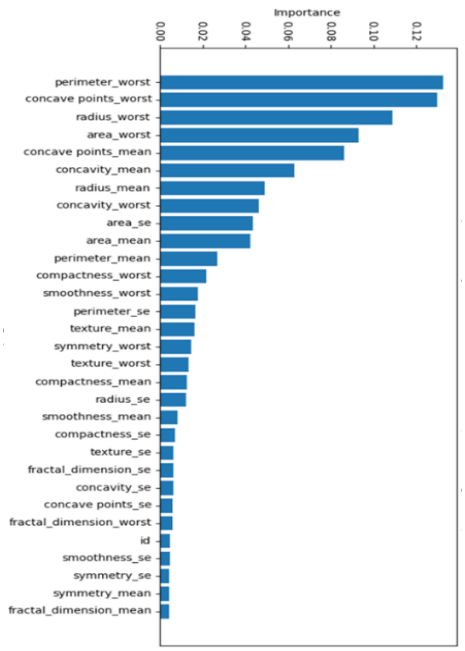

Figure 5. Random Forest features importance ranking, showing their importance.

Following the Spearman rank correlation and initial feature selection, we further validated the importance of the features using a RF classifier. The RF algorithm provided feature importance scores, highlighting the most influential variables for model development, as illustrated in Fig. 5 below. The top features were primarily geometric properties

of the tumour, such as perimeter_worst, concave points_worst, and radius_worst. These features, representing worst-case tumour measurements, played a critical role in distinguishing between classes, suggesting that extreme tumour characteristics are essential for accurate predictions.

In contrast, features such as fractal_dimension_mean, symmetry_mean, and smoothness_se were among the least important, contributing minimally to model performance. These features likely provided less useful information for classification, reaffirming the need to focus on features with higher predictive value.

This two-step approach involving the combination of spearman rank correlation with a tree-based method allowed us to filter out less relevant features while retaining those most critical for improving the model's predictive power. The results emphasise the importance of selecting features that capture key biological characteristics, particularly in contexts like cancer classification, where geometric properties of tumours are pivotal in distinguishing malignant from benign cases.

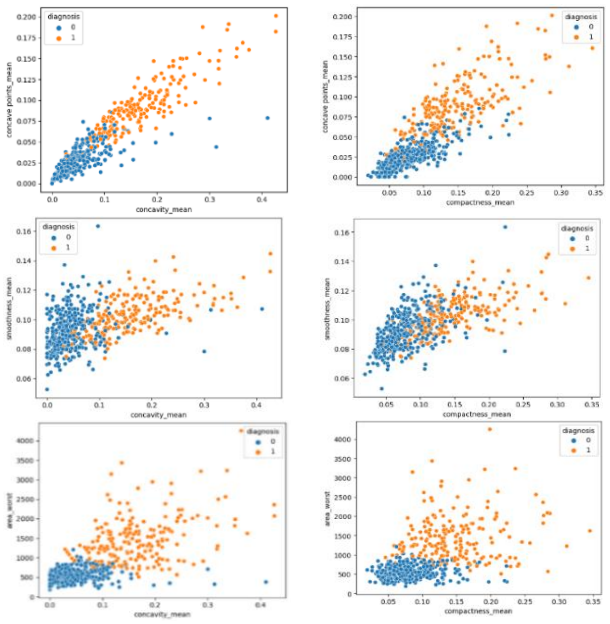

Figure 6. Scatter plot showing relationships between selected features. (Additional views of relationships between other features can be accessed in the GitHub repository [43]).

This method, known for balancing interpretability and computational efficiency while capturing both linear and non-linear relationships, affirmed the chosen features, as shown in Fig. 6, underscoring their significance in model development [42]. The synergy between the two methods ensured a comprehensive and accurate feature selection process, crucial for enhancing the model's predictive capabilities.

Understanding the relationship between the features helped to inform the class of ML algorithms that will be best suited for the classification task.

C. Model Selection and Implementation

In this study, four supervised ML classification algorithms were selected based on their unique attributes and widespread usage in previous research. LR was chosen for its ability to estimate outcome probabilities, making it a suitable choice for binary classification tasks. Its interpretability and computational efficiency further contribute to its popularity, as it provides a balance between performance and simplicity. On the other hand, DT, RF, and GBoost were selected for their ability to partition the data recursively. This recursive approach enables these algorithms to efficiently identify the most relevant features and optimal split points, which is crucial for improving classification accuracy.

The study was conducted using three platforms: KNIME Analytics Platform (Version 4.7.6), Python (Version 3.11.4, JupyterLab) with the Scikit-learn library, and MATLAB R2024a. For each platform, the ML algorithms were trained and tested using default settings, without any parameter tuning.

In KNIME, an exception was made for the RF algorithm, where the default split criterion was modified from "Information Gain Ratio" to "Gini Index." This adjustment was made to align with the default settings used in Scikit-learn, ensuring consistency and fairness in the comparative analysis. No such adjustments were made in MATLAB, as the platform's default configurations were retained for all algorithms. This approach allowed for a standardised comparison of the platforms, providing insights into how each platform handles the same ML models under comparable conditions.

To assess the algorithms' performance, the dataset was divided using an 80:20 train-test split. This split allocated 80% of the data for training, allowing the models to learn from the underlying patterns in the data, while the remaining 20% was used to test their ability to generalise to new, unseen instances. This approach provided a robust framework for evaluating the algorithms' effectiveness in classification tasks.

The source code and data used in this study, are available in a public GitHub repository to facilitate transparency and reproducibility. The methodology employed was designed to allow for a comprehensive evaluation of the selected algorithms while ensuring consistency in the comparative analysis [43].

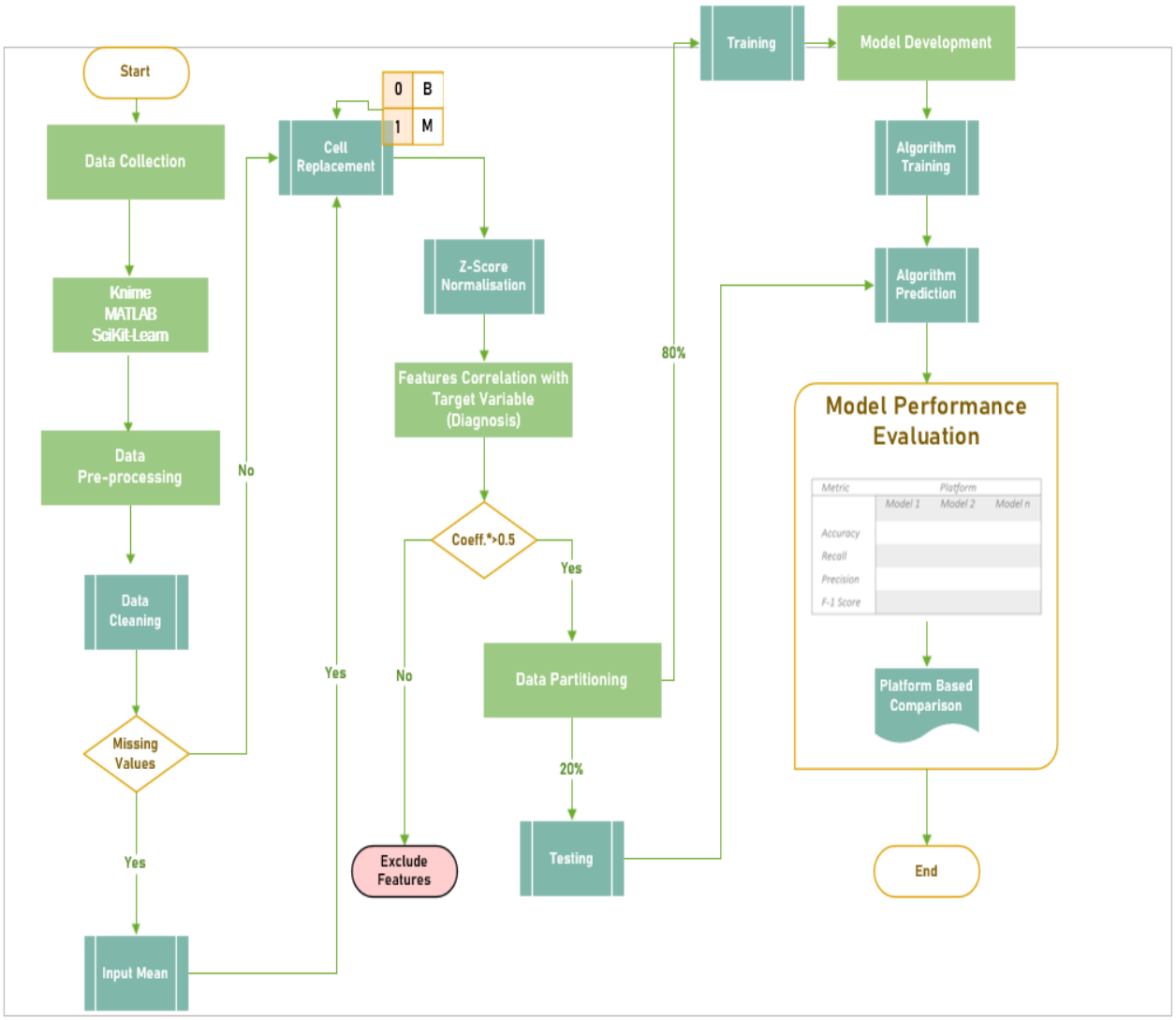

Figure 7. Flowchart illustrating the research methodology employed in this study.

IV. RESULTS AND DISCUSSION

This section outlines the experimental results obtained from implementing the four ML algorithms, LR, DT, RF, and GBoost across three platforms: Scikit-learn, KNIME, and MATLAB. The results are summarised in Table III and visually illustrated in Fig. 8, which depicts how these algorithms performed on the different platforms based on key metrics: Accuracy, Recall, Precision, and F1-Score. These metrics were used to evaluate and analyse the effectiveness of each algorithm in handling classification tasks across the platforms. The implementation of each platform was carefully examined to provide insights into how underlying differences in architecture and execution influence model performance.

A. Results Overview

Beginning with LR, Scikit-learn exhibited the highest overall performance across all metrics. An accuracy of 95.6%, combined with a recall of 0.929, precision being 0.951, and an F1-Score of 0.940, reflects the platform's ability to balance sensitivity and specificity under default settings. The high recall indicates that Scikit-learn's implementation is particularly effective at identifying true positives which is an important characteristic in healthcare scenarios where the misclassification of a malignant tumour as benign could delay treatment. Moreover, the precision score suggests that the platform manages to minimise false positives, which helps avoid unnecessary treatment for benign cases. Given that LR is a foundational algorithm, these results may reflect a strong alignment between the algorithm's mathematical structure and the default handling by Scikit-learn.

TABLE III. COMPARATIVE ASSESSMENT OF MODEL PERFORMANCE ON THE TWO PLATFORMS.

Algorithm	Tool	Accuracy	Recall	Precision	F1-Score
LR	Scikit-learn	0.956	0.929	0.951	0.940
	KNIME	0.921	0.884	0.905	0.894
	Matlab	0.938	0.921	0.897	0.909
DT	Scikit-learn	0.930	0.952	0.870	0.909
	KNIME	0.886	0.907	0.813	0.857
	Matlab	0.903	0.833	0.897	0.864
RF	Scikit-learn	0.947	0.976	0.891	0.932
	KNIME	0.912	0.884	0.884	0.884
	Matlab	0.956	0.925	0.949	0.937
GBoost	Scikit-learn	0.974	0.976	0.953	0.965
	KNIME	0.904	0.861	0.881	0.871
	Matlab	0.965	0.949	0.949	0.949

KNIME's performance for LR was comparatively lower, with an accuracy of 92.1%, recall of 0.884, precision of 0.905, and an F1-Score of 0.894. The lower recall indicates a reduced sensitivity to identifying positive cases, implying a higher rate of missed malignancies, which could have severe consequences in diagnostic applications. The precision, while reasonable, suggests that KNIME's default implementation may produce more false positives than Scikit-learn. This difference in the balance of sensitivity and specificity between platforms could have obvious practical implications in fields where both false negatives and false positives carry significant costs.

MATLAB's implementation of LR on the other hand showed an intermediate performance between the two platforms, with an accuracy of 93.8%, recall of 0.921, precision of 0.897, and an F1-Score of 0.909. Although MATLAB showed better recall than KNIME, indicating improved detection of true positives, its precision was lower than that of Scikit-learn suggesting that MATLAB's LR model may generate a higher number of false positives under default conditions, potentially leading to overdiagnosis in clinical settings. Despite this, the relatively balanced performance across all metrics indicates that MATLAB can still handle classification tasks effectively, albeit with slight trade-offs in sensitivity versus specificity.

The results from the DT algorithm reveal more noticeable disparities between platforms. Scikit-learn achieved an accuracy of 93.0%, recall of 0.952, precision of 0.870, and an F1-Score of 0.909, indicating a robust performance in classifying positive cases. The high recall suggests that Scikit-learn's DT model was able to identify most malignant cases, which is critical in ensuring that no critical diagnoses are overlooked. However, the slightly lower precision score points to a higher rate of false positives, meaning that some benign cases were misclassified as malignant, potentially leading to unnecessary medical interventions.

In contrast, KNIME demonstrated lower overall performance with DT, recording an accuracy of 88.6%, recall of 0.907, precision of 0.813, and an F1-Score of 0.857. The reduction in both precision and recall indicates that KNIME's DT model may struggle more with distinguishing between positive and negative cases under default settings. A lower precision suggests that false positives are more frequent, while the lower recall implies that true positives are being missed. This is particularly concerning in healthcare applications, where both types of errors can have significant consequences for patient care.

MATLAB's DT implementation performed with an accuracy of 90.3%, recall of 0.833, precision of 0.897, and an F1-Score of 0.864. While MATLAB's precision was higher than that of Scikit-learn, indicating fewer false positives, its lower recall shows that it missed more positive cases. This balance suggests that MATLAB's DT model, under default settings, may be more conservative, favoring the reduction of false positives but potentially at the expense of missing some malignant cases. In contexts where it is critical to detect as many positive cases as possible, this trade-off in favor of precision could impact decision-making.

Moving to the RF algorithm, both Scikit-learn and MATLAB exhibited strong performances with accuracy levels of 95.6%. However, Scikit-learn's recall (0.976) was notably higher than MATLAB's (0.925), suggesting that Scikit-learn's implementation was more sensitive to identifying true positives. This higher recall is particularly important in healthcare applications where failing to detect a malignant case could have serious consequences. In contrast, MATLAB's precision (0.949) exceeded that of Scikit-learn

(0.891), implying that MATLAB's RF model produced fewer false positives. This suggests that MATLAB's default RF implementation may prioritise specificity over sensitivity, which could be advantageous in cases where reducing unnecessary medical procedures is critical. The F1-Scores of 0.932 for Scikit-learn and 0.937 for MATLAB reflect a strong overall balance in their respective RF models.

KNIME's RF performance was lower, with an accuracy of 91.2%, recall of 0.884, precision of 0.884, and an F1-Score of 0.884. The equal precision and recall scores suggest that KNIME's RF model maintained a balance between sensitivity and specificity, but both were lower compared to Scikit-learn and MATLAB. The lower recall indicates that KNIME's RF model, under default settings, may miss more malignant cases, while the lower precision points to a higher rate of false positives. This trade-off could be significant in clinical applications where minimising both false positives and false negatives is essential.

TABLE IV. PLATFORM BASED CONFUSION MATRIX OF THE ALGORITHMS.

Scikit-learn

	LR		DT	
	Positive	Negative	Positive	Negative
Positive	70	2	66	6
Negative	3	39	2	40

	RF		GBoost	
	Positive	Negative	Positive	Negative
Positive	67	5	70	2
Negative	1	41	1	44

KNIME

	LR		DT	
	Positive	Negative	Positive	Negative
Positive	67	4	62	67
Negative	5	38	4	5

	RF		GBoost	
	Positive	Negative	Positive	Negative
Positive	66	5	66	5
Negative	5	38	6	37

MATLAB

	LR		DT	
	Positive	Negative	Positive	Negative
Positive	71	3	67	7
Negative	4	35	4	35

	RF		GBoost	
	Positive	Negative	Positive	Negative
Positive	71	3	72	2
Negative	3	36	2	37

The GBoost algorithm exhibited the largest performance differences across platforms. Scikit-learn achieved an accuracy of 97.4%, recall of 0.976, precision of 0.953, and an F1-Score of 0.965. These results suggest that Scikit-learn's default GBoost model is highly sensitive and effective at minimising both false positives and false negatives. High recall ensures that most positive cases are correctly identified, while high precision reduces the number of benign cases misclassified as malignant. This balance makes Scikit-learn's GBoost implementation suitable for applications where both sensitivity and specificity are crucial.

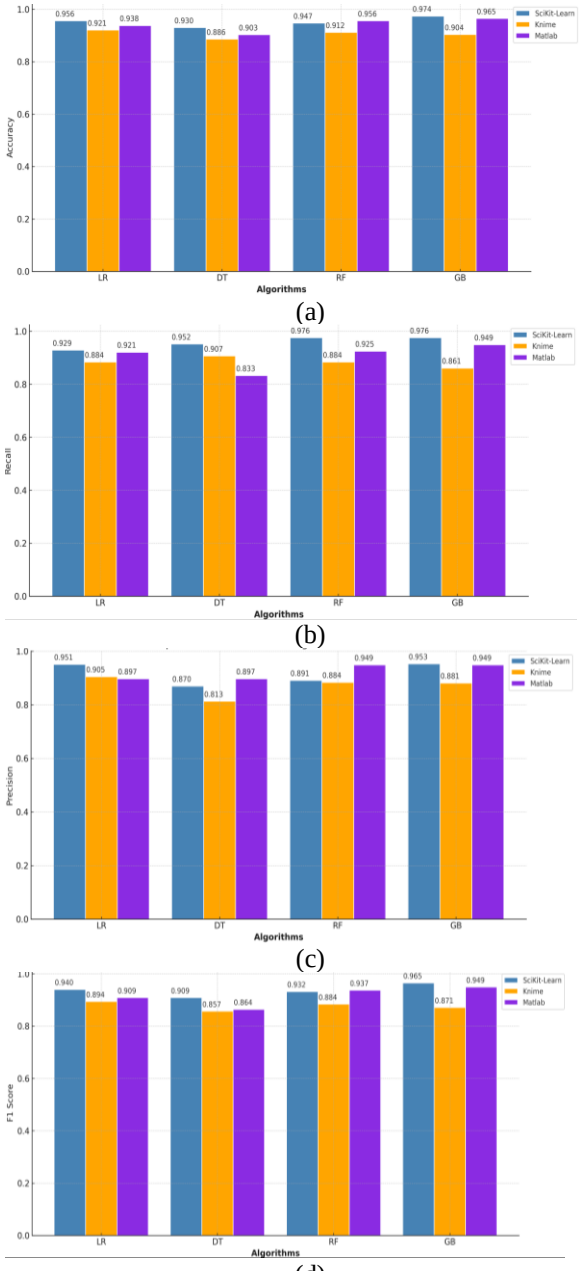

Figure 8. Column chart comparing the performance of all algorithms on the platforms for:
(a) Accuracy, (b) Recall, (c) Precision, and (d) F1-Score.

MATLAB also performed well with GBoost, achieving an accuracy of 96.5%, recall of 0.949, precision of 0.949, and an F1-Score of 0.949. While the results are close to those of Scikit-learn, MATLAB's slightly lower recall suggests that it may miss more positive cases, which could be critical in high-stakes applications like cancer detection. However, its

matching precision indicates that MATLAB is effective at reducing false positives, contributing to its balanced overall performance.

In comparison, KNIME's GBoost model showed lower performance, with an accuracy of 90.4%, recall of 0.861, precision of 0.881, and an F1-Score of 0.871. The lower recall value indicates that KNIME's model may miss more true positives, and the reduced precision suggests a higher rate of false positives compared to Scikit-learn and MATLAB.
This could impact diagnostic accuracy, especially in scenarios where identifying every possible positive case is crucial to patient outcomes.

Further insights into these performance metrics are provided by analysing the confusion matrices. Scikit-learn consistently demonstrated lower false negative (FN) rates compared to KNIME, particularly for RF and GBoost. For example, Scikit-learn's RF and GBoost models reported only 1 false negative each, while KNIME misclassified 5-6 malignant cases as benign. In the context of medical diagnostics, such discrepancies are significant, as false negatives can delay necessary treatments and worsen patient prognosis. Scikit-learn's lower false positive (FP) rates across all algorithms also suggest fewer benign cases misclassified as malignant, reducing the likelihood of unnecessary medical interventions and related costs. This trend was especially evident in the DT and RF models, where KNIME displayed higher FP rates, indicating that platform-specific characteristics might influence error rates in default implementations.

The study reveals performance variations in the algorithms tested across the platforms when executed with default settings. Scikit-learn consistently showed higher recall across all algorithms, particularly in RF and GBoost, where minimising false negatives is critical. MATLAB performed comparably in many instances but generally exhibited slightly lower recall, potentially missing more true positives. KNIME, while maintaining a balance between precision and recall, generally demonstrated lower performance in both metrics, especially in GBoost. These findings underscore the importance of considering the implementation platform when developing ML models, as platform-specific characteristics can influence how models handle classification tasks and the balance between sensitivity and specificity.

In the context of cancer care and ML research, analysing the confusion matrix presented in Table IV, which includes values for true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), provides critical insights into model effectiveness and potential clinical implications. For cancer diagnosis, the priority is to minimise false negatives (FN), as these represent cases where malignant tumours are misclassified as benign. In the presented tables, models implemented in Scikit-learn consistently have lower FN rates compared to KNIME, particularly with RF and GBoost. Scikit-learn's RF and GBoost models show only 1 false negative, whereas KNIME's implementations misclassified 5-6 malignant tumours as benign, raising concerns about its reliability in this critical area.

Equally important is the rate of false positives (FP), where benign tumours are mistakenly classified as malignant. While false positives are less harmful than false negatives, they still pose risks in cancer care by leading to unnecessary treatments, patient anxiety, and potential overtreatment. In this regard, Scikit-learn once again shows better performance, with fewer false positives across models compared to KNIME. For instance, Scikit-learn's LR and GBoost models have just 2 false positives each, whereas KNIME's counterparts show a higher rate, with up to 9 false positives in the DT model.

The true positives (TP) and true negatives (TN) in both platforms indicate the number of correctly classified malignant and benign cases, respectively. High TP values are essential in ensuring that patients with cancer receive timely treatment, while high TN values prevent unnecessary interventions for healthy patients. Scikit-learn demonstrates higher TP and TN rates overall, especially in the RF and GBoost models, where the identification of both malignant and benign cases is nearly flawless. This performance consistency highlights the critical importance of model accuracy and optimisation in ML research for cancer care, where minimising both FN and FP is essential for improving clinical outcomes.

B. Statistical Analysis

The results of this study reveal that the performance of classification algorithms can vary across machine learning platforms, even with consistent datasets and preprocessing steps. To validate these observations, the normality of performance metrics including Accuracy, Recall, Precision, and F1-Score was assessed using the Shapiro-Wilk test. All metrics satisfied the normality assumption, with p-values exceeding 0.05, enabling the use of parametric tests. Repeated measures ANOVA in Table V identified statistically significant differences in Accuracy ($p = 0.0054$) and F1-Score ($p = 0.0107$), suggesting that variations in these metrics are unlikely to be random and may be influenced by platform-specific factors. However, Recall ($p = 0.0730$) and Precision ($p = 0.0757$) did not show significant differences, indicating limited platform-specific effects on these metrics.

TABLE V. ANOVA TEST RESULTS

Metric	F Value	p-value	Significance
Accuracy	14.1037	0.0054	Yes
Recall	4.1794	0.073	No
Precision	4.0927	0.0757	No
F1-Score	10.6302	0.0107	Yes

TABLE VI. FRIEDMAN'S TEST RESULTS

Metric	Friedman Statistic	p-value	Significance
Accuracy	6.5	0.0388	Yes
Recall	6.5	0.0388	Yes
Precision	3.5	0.1738	No
F1-Score	6.5	0.0388	Yes

To further confirm these findings, Friedman's test, a non-parametric alternative, reinforced the ANOVA results by identifying significant variances in Accuracy, Recall, and F1-Score ($p = 0.0388$ for all), while Precision remained statistically insignificant ($p = 0.1738$) as seen in Table VI. Pairwise comparisons using Tukey's HSD test, shown in Table VII indicate that KNIME significantly underperformed compared to Scikit-learn in Accuracy, Recall, and F1-Score ($p = 0.0302$, 0.0311, and 0.0299, respectively). MATLAB exhibited no significant differences when compared to either platform, indicating comparable performance but neither superiority nor inferiority. Precision showed no significant differences across any platform pairs, highlighting its insensitivity to platform-specific factors in this context.

These results underscore that platform-specific characteristics, such as optimisation techniques and library implementations, play a significant role in influencing Accuracy and F1-Score but have minimal impact on Recall and Precision. Scikit-learn's superior Recall and F1-Score, particularly when compared to KNIME, highlight the importance of selecting platforms that consistently demonstrate high sensitivity and specificity in critical applications like healthcare. This study emphasises the importance of platform selection in machine learning research and applications, particularly in high-stakes domains like healthcare. It also sets the stage for further investigations into architectural and algorithmic factors driving these platform-dependent performance differences.

V. CONCLUSION AND FUTURE WORK

This comparative experiment examined the impact of different machine learning (ML) implementation platforms on the performance of classification models, focusing on four commonly used algorithms LR, DT, RF, and Gradient GBoost applied to the WDBC dataset. The analysis involved three platforms: Scikit-learn, KNIME Analytics, and MATLAB, and explored the behavior of these models under default configurations. Significant variations were observed across platforms, with each platform demonstrating unique strengths based on the metrics of accuracy, recall, precision, and F1-Score.

The study's findings highlighted that Scikit-learn consistently achieved high recall across algorithms like DT, RF, and GBoost, which is particularly important in healthcare applications such as cancer diagnosis, where minimising false negatives is crucial. The ability to correctly identify true positive cases ensures that malignant tumors are not overlooked, which is essential for timely treatment. In contrast, KNIME showed strong performance with LR, demonstrating higher accuracy for that algorithm but generally lower recall across other algorithms. This suggests that while KNIME may be effective in specific scenarios, its capacity to handle high-sensitivity tasks such as cancer detection, where recall is critical may be limited under default conditions.

MATLAB, meanwhile, presented a balanced approach, particularly excelling in precision with models like RF and GBoost, suggesting that it may be more suitable for applications where reducing false positives is important, such as in scenarios aiming to minimise unnecessary treatments. However, its lower recall compared to Scikit-learn suggests that it may miss more true positive cases, which could be a concern in healthcare settings which could delay treatment if a positive diagnosis is missed. These results emphasise the importance of selecting the right platform based on the specific objectives of a given ML task. The trade-offs between sensitivity (recall) and specificity (precision) can vary significantly depending on the platform, as demonstrated by the variations in performance across Scikit-learn, KNIME, and MATLAB. For applications such as cancer diagnosis, where both false positives and false negatives carry serious implications, platform choice is not just a technical consideration but a decision that can significantly influence model outcomes and, by extension, patient care.

The statistical analysis further underscores the significance of these platform-specific variations. Using the Shapiro-Wilk test, the normality of the performance metrics was confirmed, allowing parametric tests like repeated measures ANOVA to validate the significance of the observed differences. ANOVA identified statistically significant differences in Accuracy ($p = 0.0054$) and F1-Score ($p = 0.0107$), indicating that the variations in these metrics are unlikely to be random. Conversely, Recall ($p = 0.0730$) and Precision ($p = 0.0757$) did not exhibit significant differences, suggesting that platform-specific factors have minimal influence on these metrics.

These results were further validated using Friedman's test, which supported the significance of variations in Accuracy, Recall, and F1-Score while confirming that Precision remained statistically insignificant. Pairwise comparisons using Tukey's HSD test highlighted that KNIME significantly underperformed compared to Scikit-learn in Accuracy, Recall, and F1-Score, while MATLAB showed no significant differences across any metrics when compared to the other platforms. These findings underscore that platform-specific characteristics, such as optimisation techniques and library implementations, play a significant role in influencing Accuracy and F1-Score but have minimal impact on Recall and Precision.

Scikit-learn's superior recall and F1-Score, particularly when compared to KNIME, highlight the importance of selecting platforms that consistently demonstrate high sensitivity and specificity in critical applications like healthcare.

TABLE VII. TUKEY'S HSD TEST

	Accuracy			Recall			Precision			F1-Score		
Group 1 vs Group 2	Mean Difference	p-value	Significant	Mean Difference	p-value	Significant	Mean Difference	p-value	Significant	Mean Difference	p-value	Significant
KNIME vs MATLAB	0.0347	0.0987	No	0.023	0.6188	No	0.0523	0.1786	No	0.0383	0.1703	No
KNIME vs Scikit-learn	0.046	0.0302	Yes	0.0743	0.0311	Yes	0.0455	0.2557	No	0.06	0.0299	Yes
MATLAB vs Scikit-learn	0.0113	0.7345	No	0.0513	0.1368	No	-0.0068	0.9655	No	0.0218	0.52	No

It is important to highlight that this study does not aim to declare one platform superior to another in absolute terms. Instead, it provides critical insights into how platform architecture and design can influence ML model performance in different contexts. By investigating the inherent disparities in performance due to platform-specific characteristics, this work enables more informed decision-making when selecting platforms for predictive modelling. Ultimately, these findings contribute to a deeper understanding of how the interplay between ML algorithms and their implementation environments affects the reliability, accuracy, and effectiveness of models in real-world applications.

The scope of this study was intentionally focused on evaluating platform-dependent variations in ML classifier performance under default configurations. While this approach provides valuable insights, further research could extend these findings by incorporating additional analyses such as receiver operating characteristic (ROC) curves and precision-recall (PR) analyses. These techniques would enhance the interpretability of results, offering deeper insights into the trade-offs between sensitivity and specificity across platforms.

Another promising avenue for future exploration is classifier fusion, which could combine the strengths of individual classifiers to improve overall model performance. This technique holds potential for enhancing metrics such as accuracy, recall, and precision, especially in applications like cancer diagnosis, where both false positives and false negatives carry critical implications.

Expanding the study to include a broader range of machine learning algorithms, such as Support Vector Machines (SVM) and deep learning models, as well as additional datasets with varying characteristics, could further generalise the findings. Investigating the architectural differences of platforms, such as Scikit-learn, KNIME, and MATLAB, would also shed light on the underlying factors contributing to performance variations.

In conclusion, this study provides insights into how platform-specific characteristics influence ML model performance, offering practical guidance for platform selection in high-stakes applications like healthcare. While this research addresses a significant gap in the literature, it also lays the groundwork for further investigations into the interplay between ML algorithms and their implementation environments, enabling future advancements in predictive analytics and healthcare diagnostics.

REFERENCES

[1] A. S. Olowolayemo, A. Souag, and K. Sirlantzis,"Cancer: Investigating the impact of the implementation platform on machine learning models," The First International Conference on AI-Health (AIHealth 2024) IARIA, Mar. 2024, pp. 20-28, ISBN: 978-1-68558-136-7.

[2] B. S. Chhikara and K. Parang, "Global Cancer Statistics 2022: The trends projection analysis," Chem Biol Lett, vol. 10, pp. 451, Jan. 2023, Accessed: Dec. 01, 2024. [Online]. Available from: https://pubs.thesciencein.org/journal/index.php/cbl/article/view/451.

[3] "CANCER FACT SHEETS - Global Cancer Observatory." Accessed: Dec. 01, 2024. [Online]. Available from: https://gco.iarc.who.int/media/globocan/factsheets/cancers/39-all-cancers-fact-sheet.pdf.

[4] V. D. P. Jasti et al., "Computational technique based on machine learning and image processing for medical image analysis of Breast Cancer diagnosis," Security and Communication Networks, vol. 2022, pp.1-7, Mar. 2022, doi: 10.1155/2022/1918379.

[5] World Health Organization. "Cancer" Accessed: Dec. 01, 2024. [Online]. Available from: https://www.who.int/news-room/fact-sheets/detail/cancer.

[6] J. Boutry et al., "The evolution and ecology of benign tumors," Biochimica et Biophysica Acta (BBA)-Reviews on Cancer, vol. 1877, pp. 188643, Jan. 2022, doi: 10.1016/j.bbcan.2021.188643.

[7] N. Behranvand et al., "Chemotherapy: a double-edged sword in cancer treatment," Cancer immunology, immunotherapy, vol. 71(3), pp. 507-526, Mar. 2022, doi: 10.1007/s00262-021-03013-3.

[8] J. Ko, M. M. Winslow, and J. Sage, "Mechanisms of small cell lung cancer metastasis," EMBO Mol Med, vol. 13, Jan. 2021, doi: 10.15252/emmm.202013122.

[9] S. U. Khan et al., "A machine learning-based approach for the segmentation and classification of malignant cells in breast cytology images using gray level co-occurrence matrix (GLCM) and support vector machine (SVM)," Neural Comp. and Applications, vol. 34, pp. 8365–8372, Jun. 2022, doi: 10.1007/s00521-021-05697-1.

[10] S. A. Alowais et al., "Revolutionizing healthcare: the role of artificial intelligence in clinical practice," BMC medical education, vol. 23, pp. 689, Dec. 2023, doi: 10.1186/s12909-023-04698-z.

[11] Y. Zhang et al., "Machine learning-based prognostic and metastasis models of kidney cancer," Cancer Innovation, vol. 1, pp. 124–134, Aug. 2022, doi: 10.1002/cai2.22.

[12] E. Y. Abbasi et al., "Optimising skin cancer survival prediction with ensemble techniques," Bioengineering, vol. 11, pp. 43, Dec. 2023, doi.org/10.3390/bioengineering11010043.

[13] R. Yang, I. F. Tsigelny, S. Kesari, and V. L. Kouznetsova, “Colorectal cancer detection via metabolites and machine learning,” Curr. Issues in Mol. Biology, vol. 46, pp. 4133–4146, May 2024, doi: 10.3390/cimb46050254.

[14] J.P. Villemin et al., "A cell-to-patient machine learning transfer approach uncovers novel basal-like breast cancer prognostic markers amongst alternative splice variants," BMC Biology, vol. 19, pp. 1-19, Apr. 2021, doi.org/10.1186/s12915-021-01002-7.

[15] K.A. Tran et al., "Deep learning in cancer diagnosis, prognosis and treatment selection,” Genome Medicine, vol. 13, pp. 1-17, Sept. 2021, doi.org/10.1186/s13073-021-00968-x.

[16] J. Kong et al., “Network-based machine learning approach to predict immunotherapy response in cancer patients,” Nature communications, vol. 13, pp. 1-15, Jun. 2022, doi: 10.1038/s41467-022-31535-6.

[17] W. Wolberg, O. Mangasarian, and W. Street, “Breast Cancer Wisconsin (Diagnostic),” UCI Machine Learning Repository, 1993, doi: 10.24432/C5DW2B.

[18] E. Michael, H. Ma, H. Li, and S. Qi, “An optimized framework for breast cancer classification using machine learning,” BioMed Research International, vol. 2022, pp. 8482022, Feb. 2022, doi: 10.1155/2022/8482022.

[19] S. Ara, A. Das, and A. Dey, "Malignant and benign breast cancer classification using machine learning algorithms," 2021 International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 2021, pp. 97-101, doi: 10.1109/ICAI52203.2021.9445249.

[20] M. Kumar, S. Singhal, S. Shekhar, B. Sharma, and G. Srivastava, “Optimized stacking ensemble learning model for breast cancer detection and classification using machine learning,” Sustainability, vol. 14, Jan. 2022, doi: 10.3390/su142113998.

[21] M. Ebrahim, A. A. H. Sedky, and S. Mesbah, “Accuracy assessment of machine learning algorithms used to predict breast cancer,” Data, vol. 8, Feb. 2023, doi: 10.3390/data8020035.

[22] C. Zhu, C. U. Idemudia, and W. Feng, “Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques,” Informatics in Medicine Unlocked, vol. 17, pp. 100179, Jan. 2019, doi: 10.1016/j.imu.2019.100179.

[23] A. K. M. Rahman, F. M. Shamrat, Z. Tasnim, J. Roy, and S. Hossain, “A comparative study on liver disease prediction using supervised machine learning algorithms,” International Journal of Scientific & Technology Research, vol. 8, Nov. 2019, pp. 419–422, ISSN 2277-8616.

[24] M. Minnoor and V. Baths, “Diagnosis of breast cancer using random forests,” Procedia Computer Science, vol. 218, pp. 429–437, Jan. 2023, doi: 10.1016/j.procs.2023.01.025.

[25] W. Li, Y. Yin, X. Quan, and H. Zhang, “Gene expression value prediction based on XGBoost algorithm,” Frontiers in Genetics, vol. 10, pp. 1077, Nov. 2019, doi: 10.3389/fgene.2019.01077.

[26] X. Wan, “Influence of feature scaling on convergence of gradient iterative algorithm,” Journal of physics: Conference series, vol. 1213, pp. 032021, Jun. 2019, doi: 10.1088/1742-6596/1213/3/032021.

[27] X. Yi et al., “Development and External Validation of Machine Learning-Based Models for Predicting Lung Metastasis in Kidney Cancer: A large population-based study,” International Journal of Clinical Practice, vol. 2023, pp. 1-13, Jun. 2023, doi: 10.1155/2023/8001899.

[28] D. A. Omondiagbe, S. Veeramani, and A. S. Sidhu, “Machine learning classification techniques for breast cancer diagnosis,” IOP Conference Series: Materials Science and Engineering, vol. 495, pp. 012033, Apr. 2019, doi: 10.1088/1757-899X/495/1/012033.

[29] R. Shafique et al., “Breast cancer prediction using fine needle aspiration features and upsampling with supervised machine learning,” Cancers, vol. 15, pp. 681, Jan. 2023, doi: 10.3390/cancers15030681.

[30] K. M. M. Uddin, N. Biswas, S. T. Rikta, and S. K. Dey, “Machine learning-based diagnosis of breast cancer utilizing feature optimisation technique,” Computer Methods and Programs in Biomedicine Update, vol. 3, pp. 100098, Jan. 2023, doi: 10.1016/j.cmpbup.2023.100098.

[31] T. Shamu et al., “Cancer incidence among people living with HIV in Zimbabwe: A record linkage study,” Cancer Reports, vol. 5, pp. e1597, 2022, doi: 10.1002/cnr2.1597.

[32] Q. T. N. Nguyen et al., “Machine learning approaches for predicting 5-year breast cancer survival: A multicenter study,” Cancer Science, vol. 114, pp. 4063–4072, Jul. 2023, doi: 10.1111/cas.15917.

[33] T. R. Mahesh et al., “Performance analysis of xGBoost ensemble methods for survivability with the classification of breast cancer,” Journal of Sensors, vol. 2022, pp. 4649510, Sep. 2022, doi: 10.1155/2022/4649510.

[34] Y. Zhang et al., “Machine learning-based prognostic and metastasis models of kidney cancer,” Cancer Innovation, vol. 1, pp. 124–134, Aug. 2022, doi: 10.1002/cai2.22.

[35] S. Aamir et al., “Predicting breast cancer leveraging supervised machine learning techniques,” Computational and Mathematical Methods in Medicine, vol. 2022, pp. 5869529, Aug. 2022, doi: 10.1155/2022/5869529.

[36] İ. Ateş and T. T. Bilgin, “The investigation of the success of different machine learning methods in breast cancer diagnosis,” Konuralp Medical Journal, vol. 13, pp. 347-356, Jun. 2021, doi: 10.18521/ktd.912462.

[37] L. Liu, “Research on logistic regression algorithm of breast cancer diagnose data by machine learning,” In 2018 International Conference on Robots & Intelligent System (ICRIS), May 2018, pp. 157–160, doi: 10.1109/ICRIS.2018.00049.

[38] X. Feng, Y. Cai, and R. Xin, “Optimising diabetes classification with a machine learning-based framework,” BMC Bioinformatics, vol. 24, pp. 428, Nov. 2023, doi: 10.1186/s12859-023-05467-x.

[39] S. Sumin, “The impact of Z-Score transformation scaling on the validity, reliability, and measurement error of instrument SATS-36,” JP3I (Jurnal Pengukuran Psikologi dan Pendidikan Indonesia), vol. 11, pp. 166-180, Nov. 2022, dx.doi.org/10.15408/jp3i.v11i2.26591.

[40] M. Pagan, M. Zarlis, and A. Candra, “Investigating the impact of data scaling on the k-nearest neighbor algorithm,” Computer Science and Information Technologies, vol. 4, pp. 135-142, Jul. 2023, doi: 10.11591/csit. v4i2.p135-142.

[41] J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: A new perspective,” Neurocomputing, vol. 300, pp. 70–79, Jul. 2018, doi: 10.1016/j.neucom.2017.11.077.

[42] G. Alfian et al., “Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method,” Computers, vol. 11, pp. 136, Sep. 2022, doi: 10.3390/computers11090136.
[43] A.Olowolayemo (2023), Cancer3IPMLM GitHub. [Online]. Available from: https://github.com/ProfDee92/Cancer-3IPMLM/blob/main/README.md.
[44] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O’Sullivan, “A review of feature selection methods for machine learning-based disease risk prediction,” Frontiers in Bioinformatics, vol. 2, pp. 927312, Jun. 2022, doi.org/10.3389/fbinf.2022.927312.
[45] I. Kadhim Ajlan, H. Murad, A.A. Salim, and A. Fadhil Bin Yousif, “Extreme Learning machine algorithm for breast cancer diagnosis,” Multimedia Tools and Applications, pp. 1-20, Jun. 2024, doi.org/10.1007/s11042-024-19515-y.