180-Day Mortality Prediction for Critically Ill Patients

Made Izzy Prema Dharma
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
made.izzy@ui.ac.id

Alano Davin Mandagi Awuy
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
alano.davin@ui.ac.id

Kevin Adriano
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
kevin.adriano@ui.ac.id

Darren Marcello Sidabutar
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
darren.marcello@ui.ac.id

Anindiyo Banu Prabasworo
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
anindiyo.banu@ui.ac.id

Fransisco William Sudianto
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
fransisco.william@ui.ac.id

Fariz Darari
Faculty of Computer Science
University of Indonesia
Depok, Indonesia
fariz.darari@ui.ac.id

***Abstract*—Predicting the survival of critically ill patients remains a significant challenge in healthcare, particularly in intensive care settings where patient conditions can fluctuate rapidly. Despite advances in intensive care, six-month mortality rates among critically ill patients remain high, highlighting the need for reliable prognostic models. To address this, we used the SUPPORT dataset—containing data from approximately 9,000 seriously ill adults collected between 1989 and 1994—to develop and evaluate machine learning models capable of predicting in-hospital death within 180 days. Our main objectives are to develop and compare machine learning models that predict in-hospital death within 180 days, to analyze feature importance and uncover clinically meaningful insights, and to assess the applicability of these models in supporting end-of-life clinical decisions. Ultimately, an Optuna-tuned XGBoost classifier achieved the best performance with an accuracy of 94.7%. These results demonstrate the potential of machine learning models to support clinical decision-making.**

Keywords—critically ill patients, machine learning, SUPPORT dataset, XGBoost, prognostic modeling

I. INTRODUCTION

Accurately predicting patient outcomes is one of the most difficult challenges in modern healthcare, especially when dealing with critically ill patients. For patients reaching the end of their lives, prognostic uncertainty can result in needless procedures, extended pain, and a loss of autonomy. To address these issues, the SUPPORT (Study to Understand Prognoses and Preferences for Outcomes and Risks of Treatments) gathered comprehensive clinical, demographic, and physiological data from more than 9,000 patients in five U.S. healthcare facilities from 1989 to 1994, with the aim to enhance critical care decision-making and communication, particularly concerning the appropriateness and timing of end-of-life treatment [1].

The advancement of technology prompts the need to revisit the SUPPORT dataset to develop a machine learning model to predict patient mortality. The aim is to create a model that predicts the 180-day survival outcome of a critically ill patient. The survivability of the patient is based on their physiological, demographic, and clinical characteristics.

The motivation behind this paper is both clinical and technological. Clinicians frequently face difficulties when determining a patient's survival, leading to over-treatment or under-treatment. Machine Learning offers a solution by using a large-scale clinical dataset to generate individualized survival estimates. This project aims to utilize Machine Learning in helping real-world clinicians to predict the patient's 180-day mortality.

II. RELATED WORKS

Mortality prediction in critically ill patients, particularly those admitted to intensive care units (ICUs), has been a critical focus in healthcare research to improve patient management and resource allocation. Traditional severity scoring systems such as APACHE II, SOFA, and LODS have been widely used. However, they often show suboptimal predictive performance due to their limited ability to capture complex nonlinear relationships in clinical data [2].

Recent studies have demonstrated that machine learning (ML) models outperform classical scoring systems in predicting ICU mortality. For instance, a study using Random Forest (RF) on a cohort of 12,747 ICU patients achieved a high predictive accuracy for mortality with an area under the curve (AUC) of 0.945, surpassing conventional methods [3]. This study also identified lactate dehydrogenase (LDH) as a key variable contributing to mortality and length of ICU stay predictions, highlighting the value of ML in discovering important clinical predictors beyond traditional scores.

Comparative analyses of different ML algorithms have further reinforced these findings. For example, gradient-boosted decision trees (LightGBM), logistic

regression with L2 regularization, and multilayer perceptron (MLP) models were developed for mortality prediction in severe pneumonia patients, all significantly outperforming the Simplified Acute Physiology Score II (SAPS II) with AUCs above 0.82 [4]. Similarly, boosting algorithms such as CatBoost have shown superior performance (AUC 0.90) compared to logistic regression and extreme gradient boosting in predicting 30-day mortality in ICU patients from internal medicine departments [5].

In surgical ICU populations, machine learning models have also demonstrated high predictive accuracy. A decision tree classifier achieved an AUC of 0.96 for in-hospital mortality prediction, outperforming neural networks and Bayesian classifiers [6]. This study emphasized the importance of including a broad range of clinical variables (43 variables) to enhance prediction accuracy, suggesting that richer data inputs improve model performance.

Explainability and interpretability of ML models have gained attention to facilitate clinical adoption. Cox proportional hazards models combined with explainable ML techniques have been used to identify risk factors distinguishing survivors from non-survivors in ICU settings, enabling clinicians to understand model reasoning and trust predictions [7]. Moreover, ML models have been extended to predict not only mortality but also 30-day readmission rates post-discharge, indicating their broader applicability in critical care management [8].

The current study aims to build upon these prior works by developing an ML-based mortality prediction model that incorporates a comprehensive set of clinical variables and applies advanced ensemble methods to improve accuracy and robustness. Additionally, this study emphasizes interpretability through feature importance analysis and clustering techniques to stratify patients by mortality risk, thereby addressing both prediction performance and clinical usability.

III. METHODOLOGY

Prompted by a growing and aging population, predicting patient mortality has become essential for ensuring proper patient care. As healthcare systems face increasing strain from demographic shifts, accurate risk prediction is necessary to guide interventions and reduce preventable harm [9]. This project responds to these challenges by building models that forecast patient outcomes using clinical and demographic data.

To structure this process, the CRISP-DM (Cross Industry Standard Process for Data Mining) framework was applied. This methodology enables a systematic approach for transforming raw clinical data into actionable models, including steps such as data understanding, preparation, modeling, and evaluation. Previous work has demonstrated the usefulness of CRISP-DM in clinical environments for identifying high-risk individuals and improving resource planning [10].

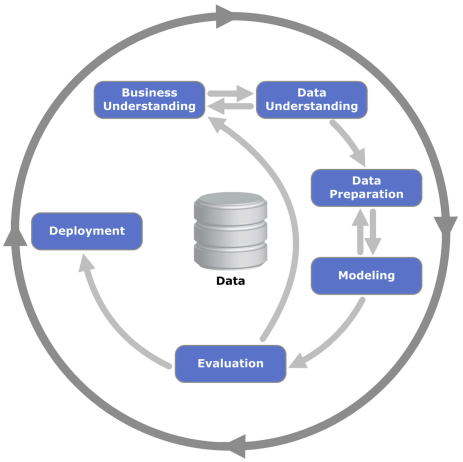

Fig. I. Graph of CRISP-DM

The study focuses on predicting patient mortality to improve healthcare by prioritizing care for high-risk patients, addressing the strain on current healthcare systems due to population growth. Using the CRISP-DM methodology, it emphasizes understanding the business problem and data characteristics to develop effective predictive models. Key stakeholders include medical professionals, patients, and healthcare institutions, benefiting from better resource allocation, treatment planning, and mortality reporting. The model aims to enhance clinical efficiency, reduce overtreatment, and improve patient quality of life while acknowledging challenges such as ethical concerns, potential algorithmic bias, data privacy, and legal responsibilities related to AI-driven decisions. This approach supports clearer communication among clinicians, patients, and families and informs healthcare readiness and policy

However, while predictive models offer considerable value, their outputs must be used to support rather than replace clinical judgment. Over-reliance on algorithmic predictions raises ethical concerns, including potential bias, loss of patient autonomy, and inappropriate care decisions. As such, machine learning predictions should serve as advisory tools within a broader, human-centered decision-making process.

The following research questions guide this project:

1. Can machine learning models accurately predict 180-day mortality in critically ill patients based on clinical and demographic features?
2. Which variables are most predictive of mortality risk?
3. Which machine learning algorithms perform best for this task in terms of accuracy and interpretability?

A. Structures and Data Exploration

The dataset comprises approximately 9,000 critically ill patients with 42 columns capturing detailed medical information, including physical condition, prior medical history, and outcome variables relevant to patient status. Among these patients, about 25.9% died during the study period, indicating a class imbalance with the majority surviving. The patient population is predominantly elderly, and a two-sample t-test confirmed that the age difference between survivors and non-survivors is statistically significant. Other features in the dataset are relatively

balanced. Using Pearson correlation analysis and heatmap visualization, several features were identified as potentially predictive of mortality, with a focus on those having the highest correlation to the target variable or a threshold of approximately 0.4. This exploration lays the groundwork for selecting meaningful predictors to improve mortality prediction models in critically ill patients.

D. Handling missing values

Addressing missing numerical values was done using domain-informed assumptions and statistical imputation when possible. This implies the use of the mean and median of several features to impute missing values in their respective columns. Some features require that it be imputed using the median of other features that existed within the same domain. Additional imputation was done using domain-informed assumptions built on an overall observation from credible sources. Primarily focusing on psychological columns, a generic value was taken to impute said missing values for each respective column.

Categorical features were imputed using their respective mode values, replacing missing entries with the most frequent category. This approach maintains consistency and helps reduce data sparsity without introducing uncommon values.

E. Handling outliers

To avoid misinterpretation and poor accuracy, a structured two-step strategy was applied. First, the interquartile range (IQR) method identified extreme values by flagging observations that fell outside the lower and upper bounds. Second, instead of discarding these points and shrinking the sample, winsorization capped them at those bounds. This technique caps all outliers at the defined upper and lower bounds, effectively reducing the number of outliers to zero while maintaining the number of rows.

F. Feature engineering

With the cleaned dataset, various feature engineering techniques were applied to improve model interpretability and prediction. One-hot encoding was used to handle non-ordinal categorical variables without assuming any order. A target variable was created to indicate whether a patient died within 180 days of hospitalization. Hospital stay duration was transformed into a categorical variable with three groups, short, medium, and long, to reflect different levels of risk. Additional features, such as chronic burden and multi-system failure risk, were created to capture the presence of serious health conditions and signs of organ failure. An advanced disease flag was also added to identify patients with severe or late-stage conditions. These features capture key medical and physical conditions, helping the model better relate them to each patient's 180-day mortality.

G. Data transformation

All numerical features were standardized using the StandardScaler method from sklearn.preprocessing utilizing the Z-score normalization, ensuring each feature has a mean of 0 and a standard deviation of 1, thereby preventing features with large absolute values from dominating the learning process. This will facilitate models that rely upon distance or gradient descent.

Furthermore, Principal Component Analysis (PCA) was applied to reduce dimensionality and enhance computational efficiency in the models. In addition to retaining a core section of data and variance. To facilitate PCA, the elbow method was applied at a target of ~80% variance to retain as much information as possible.

Analysis indicated that retaining 30 components would capture approximately 80% of the total variance, which strikes a balance between information preservation and dimensionality reduction.

H. Data Splitting

The dataset was divided into training and test subsets using an 80/20 split. The training set (80%) was used to train and validate the model, while the test set (20%) was reserved exclusively for final performance evaluation on unseen data. Since it was known that the initial dataset had around ~9000 rows, which directly means around ~9000 patients, splitting 20% into a test set would yield around ~1800 patients for testing, which was sufficient considering the quantity and quality of the data.

IV. EXPERIMENTAL RESULTS

A. Model Selection and Justification

Some strategies were applied to mitigate the impact of class imbalance in our dataset. SMOTE (Synthetic Minority Oversampling Technique) synthetically generates new examples of the minority class, which improves model generalization by helping the model learn from both classes effectively [11]. Random Undersampling techniques balance the dataset by reducing the number of majority class instances, making the model less biased toward the dominant class[12].

Among all the classification models, these machine learning classifiers were the most suitable based on the dataset's data type, complexity, interpretability needs, and computational constraints:

1. Logistic Regression: Simple, interpretable baseline, supports class weights [14]. Estimates the probability of a binary outcome using a logistic function applied to a linear combination of input features.
2. Decision Tree: Captures non-linear patterns, easy to interpret [15]. Recursively splits the data based on feature values to form a tree structure, aiming to maximize information gain or minimize impurity (e.g., Gini index or entropy).
3. SVC (Support Vector Classifier): Good for high-dimensional data, supports class imbalance [16]. Constructs an optimal hyperplane (or set of hyperplanes in kernelized versions) that maximizes the margin between different classes.
4. KNN (K-Nearest Neighbors): Simple, non-parametric, no class weight support, but useful for comparison [17]. Classifies a data point based on the majority class among its k nearest neighbors in the feature space.
5. Random Forest: Strong performance on tabular data, handles imbalance well [18]. An ensemble of decision trees trained on bootstrapped samples with random feature selection, aggregating

predictions via majority voting.

6. XGBoost: High accuracy, good with complex data, supports imbalance with scale_pos_weight [19]. An optimized gradient boosting framework that builds trees sequentially, using second-order gradient information and regularization to prevent overfitting.

B. Performance metrics

Multiple classification metrics were used to evaluate all of the model's performance:

- Accuracy: Measures the overall correctness of the model.
- Precision: Focuses on the correctness of positive predictions.
- Recall: Measures how well the model detects positive instances.
- F1-Score: The harmonic mean of precision and recall.
- ROC AUC: A threshold-independent metric that evaluates the trade-off between true positive and false positive rates.

C. Comparison of Models

These are all the performances of models using both SMOTE and Random Undersampling.

Table I. MODEL PERFORMANCE USING SMOTE AND CLASS WEIGHT

Model	**Performance Metrics**				
	Accuracy	*Precision*	*Recall*	*F1-Score*	*ROC AUC*
Logistic Regression	0.88	0.91	0.88	0.89	0.977
Decision Tree	0.89	0.89	0.89	0.89	0.85
SVC	0.90	0.92	0.91	0.91	0.97
KNN	0.82	0.87	0.82	0.83	0.91
Random Forest	0.92	0.93	0.93	0.93	0.975
XGBoost	0.92	0.93	0.92	0.93	0.977

Table I presents the performance under SMOTE, which aimed to counteract the class imbalance in the dataset. Observing the results, XGBoost outperformed every other model

Table II. MODEL PERFORMANCE USING UNDERSAMPLING

Model	**Performance Metrics**				
	Accuracy	*Precision*	*Recall*	*F1-Score*	*ROC AUC*
Logistic Regression	0.91	0.92	0.91	0.91	0.97
Decision Tree	0.88	0.90	0.88	0.89	0.88
SVC	0.91	0.93	0.92	0.92	0.974
KNN	0.85	0.88	0.86	0.86	0.92
Random Forest	0.92	0.93	0.92	0.93	0.976
XGBoost	0.92	0.93	0.92	0.93	0.979

Table II contains the results from using undersampling to mitigate bias within the models, It can be observed that XGBoost was once again the best-performing model.

Among all the tested models, XGBoost consistently outperformed other models across all evaluation metrics. It achieved an ROC AUC score of 0.9777 and an accuracy of 0.9247, indicating exceptional discriminative power between the two classes.

C. Feature Importance

A critical aspect of our analysis was identifying which features most strongly influence 180-day mortality predictions among critically ill patients. By leveraging the interpretability from our best model, which is XGBoost, we extracted and analyzed feature importance scores to uncover clinically meaningful insights.

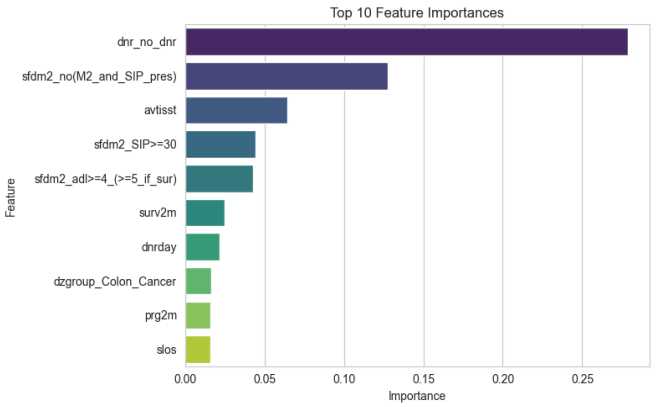

Fig. II. Graph of Feature Importance

The feature importance analysis of our XGBoost model reveals that 180-day mortality among critically ill patients is primarily driven by a combination of care preferences, functional status, and clinical severity. The presence of a Do-Not-Resuscitate (DNR) order (dnr_no_dnr) is the most influential predictor, underscoring the critical role of end-of-life decisions in forecasting outcomes. Patient functionality and engagement in care planning, as captured by sfdm2_no (M2_and_SIP_pres), are also highly predictive, reflecting how preserved independence correlates with better prognosis. The TISS score (avtisst), which quantifies therapeutic interventions, ranks next, directly indicating illness severity and resource intensity. Other significant features include high functional disability (sfdm2_SIP≥30) and lower surrogate-assessed ADL scores (sfdm2_adl≥4), both of which suggest poorer patient capacity and increased vulnerability. Short-term survival estimates (surv2m) and the timing of DNR decisions (dnrday) provide additional prognostic context, while specific diagnoses like colon cancer (dzgroup_Colon_Cancer), physician judgment (prg2m), and hospital length of stay (slos) offer further granularity in risk assessment. Together, these insights affirm that mortality risk is shaped not by any single factor but by a nuanced interplay of clinical severity, functional decline, prognosis estimation, and patient-centered care preferences, highlighting the importance of integrating both objective and subjective data in critical care decision-making.

D. Hyperparameter Tuning

Optuna is used to optimize model performance further. Optuna is an automated hyperparameter optimization framework [13]. We defined a search space involving parameters such as:

- Number of estimators
- Maximum tree depth

- Learning rate
- Subsample ratio
- Column sampling rate
- Regularization parameters (L1 and L2)

Optuna managed to improve the accuracy of the best-performing XGBoost model up to 0.9395. It reached that result by performing 50 trials using the following configurations:

Table III. TABLE OF OPTIMIZED PARAMETERS

Parameter	Value
n_estimators	350
max_depth	7
learning_rate	0.049
sub_sample	0..684
colsample_bytree	0.964
gamma	1.018e-05
reg_alpha	0.01567
reg_lambda	2.8344e-08

The optimized XGBoost model uses 350 trees (n_estimators) with a maximum depth of 7, enabling it to learn moderately complex patterns while maintaining computational efficiency. A learning rate of 0.049 allows for gradual learning, reducing the risk of overfitting by ensuring that each tree has a limited individual impact. The model applies subsampling to approximately 68% of the training data (sub_sample) and uses around 96% of the features per tree (colsample_bytree), introducing randomness that enhances generalization. Regularization parameters (gamma, reg_alpha, and reg_lambda) are set to very small values, allowing flexible tree growth and minimal penalties on model complexity. Overall, these settings reflect a careful balance between underfitting and overfitting, aiming to improve the model's performance in predicting 180-day mortality in critically ill patients.

E. Error Analysis and Limitations

The model had some difficulty distinguishing borderline cases despite strong performance metrics, especially between true positives and false positives. The limitations are:

- Class Imbalance: Although mitigated, real-world imbalance may still affect predictions.
- Feature Importance Bias: Tree-based models may prioritize high cardinality or high variance features. This in turn makes performance poor.
- Logistic Regression Instability: Logistic regression failed to converge even with increased iterations, signaling limited capacity to capture data complexity.

V. CONCLUSION AND FUTURE WORKS

This study revisited the 9,105-patient SUPPORT dataset, demonstrating that modern machine-learning models can accurately predict 180-day mortality for critically ill patients. After data cleaning, feature engineering, and class-imbalance mitigation, an Optuna-tuned XGBoost ensemble achieved 94% accuracy and a 0.98 ROC-AUC, outperforming both traditional severity scores reported in the literature and all baseline algorithms evaluated here (Logistic Regression, Decision Tree, KNN, SVC, and Random Forest).

The study still has several important limitations. The SUPPORT dataset was collected between 1989 and 1994, and since then, treatment standards and documentation practices have changed, which may limit how well the model applies to modern clinical settings. Although methods like SMOTE and undersampling improved class balance, bias may still occur in real-world scenarios. The model also lacked formal calibration and decision-threshold optimization, which are essential for clinical use. Future work should focus on testing the model with newer ICU datasets to assess its generalisability across different hospitals, time periods, and patient groups. Efforts should also be made to recalibrate the model, account for uncertainty, and support decision-making tools that use meaningful thresholds. Additionally, fairness audits should examine model performance across age, sex, insurance status, and ethnicity. Involving clinical professionals can help evaluate the model's practical value. Addressing these areas would help turn the model into a more reliable and supportive tool for improving care in critical settings.

REFERENCES

[1] Knaus, W A et al. "The SUPPORT prognostic model. Objective estimates of survival for seriously ill hospitalized adults. Study to understand prognoses and preferences for outcomes and risks of treatments." Annals of internal medicine vol. 122,3 (1995): 191-203. doi:10.7326/0003-4819-122-3-199502010-00007

[2] B. Tekin, J. Kiliç, G. Taşkin, İ. Solmaz, O. Tezel, and B. B. Başgöz, "Comparison of scoring systems: SOFA, APACHE-II, LODS, MODS, and SAPS-II in critically ill elderly sepsis patients: Journal of Infection in Developing Countries," *Journal of Infection in Developing Countries*, vol. 18, no. 1, pp. 122–130, Jan. 2024, doi: https://doi.org/10.3855/jidc.18526.

[3] S. Iwase *et al.*, "Prediction algorithm for ICU mortality and length of stay using machine learning," *Scientific Reports*, vol. 12, no. 1, Jul. 2022, doi: https://doi.org/10.1038/s41598-022-17091-5.

[4] E.-T. Jeon *et al.*, "Machine learning-based prediction of in-ICU mortality in pneumonia patients," *Scientific Reports*, vol. 13, no. 1, p. 11527, Jul. 2023, doi: https://doi.org/10.1038/s41598-023-38765-8.

[5] A. C. Genç *et al.*, "Comprehensive analyses: Using machine learning models for mortality prediction in the intensive care unit of internal medicine," *Journal of investigative medicine: the official publication of the American Federation for Clinical Research*, p. 10815589251335327, Sep. 2025, doi: https://doi.org/10.1177/10815589251335327.

[6] K. Yun, J. Oh, T. H. Hong, and E. Y. Kim, "Prediction of Mortality in Surgical Intensive Care Unit Patients Using Machine Learning Algorithms," *Frontiers in Medicine*, vol. 8, Mar. 2021, doi: https://doi.org/10.3389/fmed.2021.621861.

[7] A. H. T. Chia *et al.*, "Explainable machine learning prediction of ICU mortality," *Informatics in Medicine Unlocked*, vol. 25, p. 100674, 2021, doi: https://doi.org/10.1016/j.imu.2021.100674.

[8] T.-L. Hu, C.-M. Chao, C.-C. Wu, T.-N. Chien, and C. Li, "Machine Learning-Based Predictions of Mortality and Readmission in Type 2 Diabetes Patients in the ICU," *Applied Sciences*, vol. 14, no. 18, p. 8443, Sep. 2024, doi: https://doi.org/10.3390/app14188443.

[9] Y. Chen *et al.*, "Predicting 1-, 3-, 5-, and 8-year all-cause mortality in a community-dwelling older adult cohort: relevance for predictive, preventive, and personalized medicine," *EPMA Journal*, vol. 14, no. 4, pp. 713–726, Nov. 2023, doi: https://doi.org/10.1007/s13167-023-00342-4.

[10] M. Salimi, P. Bastani, M. Nasiri, M. Karajizadeh, and R. Ravangard, "Predicting readmission of cardiovascular patients admitted to the CCU using data mining techniques," *Open Cardiovascular Medicine Journal*, vol. 17, pp. 1–14, 2023, doi:

https://doi.org/10.2174/18741924-v17-e230627-2022-21.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *arXiv preprint arXiv:1106.1813*, Jun. 2011, doi: https://doi.org/10.1613/jair.953

[12] M. C. Untoro and M. A. N. M. Yusuf, “Evaluate of Random Undersampling Method and Majority Weighted Minority Oversampling Technique in Resolve Imabalanced Dataset,” *IT Journal Research and Development*, vol. 8, no. 1, pp. 1–13, Aug. 2023, doi: https://doi.org/10.25299/itjrd.2023.12412.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, pp. 2623–2631, 2019, doi: 10.1145/3292500.3330701.

[14] Logistic Regression: D. R. Cox, “The regression analysis of binary sequences,” *Journal of the Royal Statistical Society, Series B*, vol. 20, no. 2, pp. 215–232, 1958, doi: 10.1111/j.2517-6161.1958.tb00292.x

[15] J. R. Quinlan, “Induction of decision trees,” Machine Learning, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1023/A:1022643204877

[16] Support Vector Classifier (SVM): C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018

[17] K-Nearest Neighbors (KNN): T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964

[18] Random Forest: L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324

[19] XGBoost (Extreme Gradient Boosting): T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785