Secure Heart Disease Status Classification Using Machine Learning and Blockchain-Based EHR Integrity Verification

Haya H. Alsharif, Sabah M. Alzahrani
Department of Computer Science-College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

***Abstract*—Heart disease remains one of the leading causes of mortality worldwide, motivating the development of accurate and scalable screening-support tools. This study presents an integrated framework for heart-disease status classification using the Heart 2020 Cleaned dataset, derived from the CDC Behavioral Risk Factor Surveillance System (BRFSS), a telephone-based survey of self-reported health information rather than clinical or electronic-health-record data. The task is therefore to classify a respondent's self-reported heart-disease status, not to predict future disease onset. The pipeline includes categorical encoding, feature scaling, recursive feature elimination, and class-imbalance handling using BorderlineSMOTE, SMOTETomek, and SMOTEENN. Five classifiers were evaluated: Random Forest, XGBoost, LightGBM, k-Nearest Neighbours, and a feed-forward deep neural network. Sampling strategy affected reported performance. Under the methodologically valid post-split setting, in which resampling is applied only to the training data, the models achieved an AUC of approximately 0.85, comparable to prior work on the same dataset. Pre-split global resampling produced much higher scores, with Random Forest reaching 97.14% accuracy, an F1-score of 0.9735, and an AUC of 0.9965 under SMOTEENN. However, these results are optimistically biased because resampling before splitting leaks synthetic information into the test set. The system also includes a blockchain-based integrity layer that stores full records in the application database while anchoring SHA-256 digests and verification metadata in an Ethereum-compatible environment. Overall, this prototype demonstrates the practical integration of established machine-learning classification and blockchain-supported integrity-verification components.**

Keywords*—Heart-disease status classification; machine learning; ensemble model; random forest; XGBoost; LightGBM; Blockchain; Electronic Health Records (EHR)

I. Introduction

Artificial Intelligence (AI) has become a central component of modern healthcare, where it supports data-driven decision-making in disease diagnosis, risk prediction, prognosis, and personalized treatment planning [1]. AI systems are capable of processing vast and varied clinical datasets, detecting hidden patterns, and helping healthcare professionals make timely, informed decisions. Within this broader domain, Machine Learning (ML) has emerged as one of the most effective AI paradigms for predictive healthcare, with successful applications reported across disease screening, clinical risk stratification, and diagnostic support systems [2].

Cardiovascular disease is one of the main causes of mortality, and it constitutes one of the greatest health problems in the world. It is also one of the clinical fields where AI for diagnosis and health-data analysis can make a big difference. Since cardiovascular diseases account for a significant proportion of global mortality, there has been a rising interest in the use of ML tools for scalable population-level health assessment. ML models are increasingly used to support heart-disease status classification by learning from demographic, behavioral, and self-reported health variables that may be difficult to integrate manually in routine care [3]. These capabilities are especially relevant for survey-based health assessment, where status classification of reported heart-disease cases can support population-level decision-making and public-health planning [4].

Meanwhile, as health care is becoming more reliant on digital health information, privacy, integrity, interoperability, and patient control issues have emerged. Although EHR systems are essential for supporting clinical decisions, many conventional centralized architectures remain vulnerable to unauthorized access, limited auditability, interoperability constraints, and single points of failure [5]. Technologies such as smart contracts, decentralized ledgers, and IPFS-supported storage can improve traceability, tamper resistance, patient-centric access control, and secure health-record sharing [6]. But the technologies are typically designed as secure data-management systems, not a comprehensive framework that brings together heart-disease status classification and secure EHR integrity verification.

Within the realm of heart disease research, several recent studies have proposed different ML models, including gradient boosting, ensemble learning, deep learning, and NLP-based approaches for extracting cardiovascular risk factors from structured and unstructured data [3,7]. Although these studies have immense potential for the use of ML in HD prediction, they also have certain key limitations. Many works focus either on predictive performance alone or on secure record management alone, without integrating both into a unified healthcare framework [4]. In addition, class imbalance, dataset heterogeneity, lack of interpretability, and variation in evaluation practice between studies remain a problem in the literature.

This study develops and evaluates an integrated framework that combines machine-learning-based heart-disease status classification with blockchain-supported EHR integrity verification. Recent literature related to heart-disease classification and blockchain in healthcare data management

indicates that there are a number of limitations and aspects that can be improved. Class imbalance and minority-class sensitivity remain persistent issues, especially in large population-scale studies, and are often overlooked in studies that emphasize overall accuracy. In some existing works, only a small number of models are evaluated, or the studies consider only structured or unstructured data, which limits the possibility of cross-model comparison in a unified pipeline. Moreover, most of the existing literature on blockchain-based EHRs gives high priority to security, privacy, and access management. Only a few works have integrated classification analytics with blockchain for real-time heart-disease status classification and EHR integrity verification. Many secure-health data-management systems increase auditability and patient control but may not be designed for heart-disease status classification within a healthcare-oriented decision-support prototype based on large-scale behavioral-risk survey data.

Based on the above research gaps, this study addresses the following research questions:

- RQ1: How accurately can machine-learning models classify a respondent's self-reported heart-disease status, and which model performs best?
- RQ2: How does class-imbalance handling affect classification performance?
- RQ3: Does the timing of resampling (before vs. after the train/test split) change the reported results?
- RQ4: Can a blockchain layer verify the integrity of health records and detect tampering?

To improve the accuracy and methodological soundness of heart-disease status classification while supporting secure patient record management, this study addresses these limitations through the following contributions:

- A reproducible framework for heart-disease status classification is developed using the BRFSS-based Heart 2020 Cleaned dataset, incorporating preprocessing, feature engineering, recursive feature elimination, and explicit class-imbalance handling, as a practical integration of established techniques rather than a novel algorithm.
- Five established classifiers (Random Forest, XGBoost, LightGBM, k-Nearest Neighbours, and a feed-forward DNN) are compared under a common pipeline to identify the most reliable configuration for heart-disease status classification.
- An analysis of class-imbalance handling techniques and the effect of resampling timing (post-split versus pre-split) on classification performance is presented, explicitly identifying pre-split resampling as a source of data leakage and optimistic bias.
- A blockchain-based integrity-verification layer is integrated into the prototype, with full patient records stored in the application database and SHA-256 digests anchored through an Ethereum-compatible environment to provide tamper-evident verification and traceability; access control is enforced at the API layer rather than on-chain.

This research therefore presents a prototype/feasibility study that integrates machine-learning-based heart-disease status classification with a blockchain-based EHR integrity layer. The system is designed to deliver a reproducible, healthcare-oriented decision-support framework, and its primary contribution is the practical integration of established components rather than a novel algorithm or blockchain architecture. The rest of the study is organized as follows. Section II reviews machine-learning-based heart-disease classification and blockchain-based EHR systems. Section III describes the materials and methods. Section IV presents experimental results. Section V discusses the results and compares them with existing works. Finally, Section VI concludes the study and suggests directions for future research.

II. Related Work

A lot of research work has investigated machine-learning methods for predicting and classifying cardiovascular risk from various kinds of data such as population surveys, clinical tabular datasets, physiological signals, imaging, free-text electronic health records, and omics data.

In studies based on large population surveys, Chen et al. [8] built a comparative framework on the Heart_2020_cleaned dataset derived from BRFSS, examining how Boruta feature selection, class balancing, and model tuning affect the performance of decision trees, random forests, logistic regression, and LightGBM. Based on their results, they obtained that LightGBM gives the highest overall accuracy. Age category and rate of health were the most significant predictors. Also, it should be noted that the specificity was always greater than the sensitivity. Febriyanti and Baita [9] focus on the severe class imbalance in Heart_2020_cleaned and use random under-sampling together with tuned SVM and CART decision trees; they demonstrate that under-sampling reduces overall accuracy but substantially improves minority-class recall and F1-score, explicitly highlighting the trade-off between balanced performance and global accuracy in this dataset. Tompra et al. [10] examine the 2021 BRFSS heart-disease data with extensive feature engineering and a spectrum of resampling methods such as SMOTE-ENN and ADASYN applied only to the training set; their best CatBoost configuration achieves very high recall but low precision, underscoring the difficulty of avoiding false negatives in highly imbalanced population-level prediction.

Osei-Nkwantabisa and Ntumy [11] combine multiple UCI heart-disease sources into a dataset of 1,190 patients, apply grid-search tuning to logistic regression, KNN, SVM and an ANN, and report KNN as the best performer with balanced precision and recall, though still below the very high scores claimed in some deep-learning works. Bhatt et al. [12] analyze a larger Kaggle cardiovascular disease dataset of 70,000 patients, using k-modes clustering to transform categorical variables before training decision trees, random forests, MLPs and XGBoost; they find that an MLP trained on clustered data achieves the highest accuracy and AUC, suggesting that clustering can be a useful preprocessing step for fully categorical risk-factor data. Almulihi et al. [13] construct a deep stacking ensemble that uses CNN-LSTM and CNN-GRU hybrids as base learners with an

SVM meta-classifier, evaluating the system on a large Kaggle heart-disease dataset and on Cleveland, and they report strong gains over individual models, particularly on the smaller dataset.

There has been a rise in the use of blockchain technology to support safer and more interoperable electronic health record (EHR) systems in healthcare systems in recent years. Many of the efforts were focused on developing blockchain-based architectures for healthcare records. Tahir et al. [5] developed an Ethereum-based EHR management framework where encrypted records are stored off-chain in IPFS and hashes plus metadata are maintained on-chain via smart contracts, with a web interface enabling hospitals and patients to upload and retrieve records securely. They showed low gas consumption and their response time at a load of simulated milliseconds is excellent, which implies the availability for real-world usage. Mandarino et al. [14] focused on cost and latency by introducing EdgeHR, a public-Ethereum dApp that keeps medical data on edge devices while recording hashed references and access policies on-chain; through careful use of smart-contract design patterns, they demonstrate significant gas savings while preserving integrity and auditability. These are examples of other healthcare systems that are securing only metadata and access logs on the blockchain, with the large medical data files stored off-chain.

Another line of work is concentrating on providing access control for EHRs using privacy-preserving, fine-grained parameters. Ali et al. [15] presented a Hyperledger Fabric framework that integrates attribute-based encryption and signatures with a multi–certificate authority design, avoiding a single point of trust and enabling cross-organizational access control. Using their performance measures with Hyperledger Caliper, they demonstrate that their experiments indicate that their MedRec-previously proposed system comparisons achieve higher throughput and lower overheads.

More recently, Kumari et al. [16] proposed BL-MVC, a blockchain-enabled framework that combines Ethereum-based blockchain storage, IPFS, Solidity smart contracts, a React-based front end, and a FastAPI back end, with a Majority Voting Classifier for heart disease prediction. Several machine learning models were used for these predictive models, including Logistic Regression, MLP, AdaBoost, CatBoost, and XGBoost. The blockchain layer helps in hosting patient records, which are decentralized and tamper-proof. The health care data set used for the study is based on the BRFSS (Behavior Risk Factor Surveillance System); predicting cardiac disease from the outcome column is a more straightforward example of combining data management improvements obtained with blockchain with embedded predictive analysis – aspects that are relevant for systems intended to support data protection improvements but also clinical decision support in a single system.

III. METHODOLOGY

The main procedure for classifying heart-disease status in this project involves integrating blockchain technology with machine-learning models that process population-level health data and output a status classification for each individual. The objective is twofold: 1) to use routinely collected cardiovascular risk indicators — such as demographic, behavioral, and self-reported health variables — to provide scalable, data-driven support for the classification of individuals according to their self-reported heart-disease status; and 2) to manage Electronic Health Records (EHRs) securely in the application database while using blockchain-based anchoring of cryptographic digests to provide tamper-evident integrity verification and traceability.

The general structure of the proposed system is shown in Fig. 1. The framework begins with patient information and health-related input data received through the user interface and coordinated by the application server. The machine-learning module processes the relevant data, applies preprocessing, and generates heart-disease status classifications. At the same time, patient EHRs are stored in the application database, while cryptographic digests and related identifiers are optionally anchored on an Ethereum-compatible blockchain for integrity verification. The following are key components of the system:

A. Key components of the system

1) User Interface (GUI): A front-end interface through which healthcare professionals and authorized users can enter patient data, request heart-disease status classifications, and securely access or update EHR information.

2) Machine learning model: Applies a chosen classification model (Random Forest, XGBoost, LightGBM, k-NN, and feed-forward DNN), and pre-processing, training, tuning, and classification.

3) Blockchain storage module: Provides a tamper-evident integrity layer by storing record hashes and verification-related metadata, ensuring traceability and protection against unauthorized modification; only cryptographic digests are stored on-chain, while full records remain in the application database.

4) Application server: Coordinates communication between the GUI, the ML module, the database, and the blockchain layer, while handling authentication, authorization, and the return of status-classification results and record-related responses to the GUI.

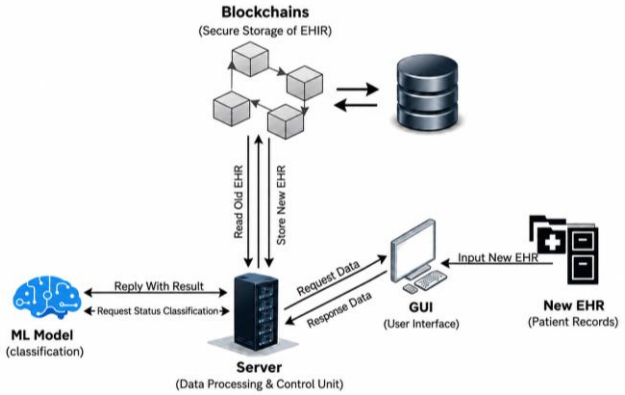

Fig. 1. General framework of the proposed system that secures EHRs and classifies heart-disease status.

B. Proposed Method

The proposed method is a machine-learning pipeline designed to classify heart-disease status using the Heart 2020 –

Cleaned (BRFSS) dataset. The method begins with preprocessing the original survey data, followed by development of multiple classification models and, finally, comparing the models' performance using standard metrics. The aim is to find an effective model to classify individuals as positive-class (heart-disease present) or negative-class (heart-disease absent) from demographic, behavioral, and self-reported health features. The general process for the proposed method is shown in Fig. 2.

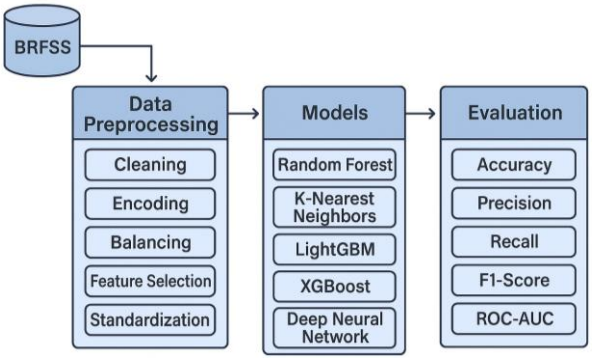

Fig. 2. Proposed machine learning method.

The proposed method consists of the following main stages:

1) Data preprocessing: Raw BRFSS data are prepared before model training to ensure that the dataset is suitable for machine-learning analysis. This stage includes the following steps:

2) Cleaning: The dataset is checked and prepared to remove inconsistencies and ensure that the values are ready for analysis.

3) Encoding: Categorical variables are transformed into numerical form so they can be processed by the classification models.

4) Balancing: Class imbalance in the target variable is addressed using resampling techniques to improve the representation of minority heart-disease cases.

5) Feature selection: The most informative variables are identified and retained to reduce redundancy and improve model performance.

6) Standardization: Numerical features are scaled to a common range so that differences in magnitude do not adversely affect model training.

7) Model development: The processed data are then used to train multiple classification models, namely Random Forest, k-Nearest Neighbours, LightGBM, XGBoost, and a feed-forward Deep Neural Network.

8) Performance evaluation: The trained models are assessed using standard classification metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, to compare their effectiveness in heart-disease status classification.

9) Model comparison and selection: Based on the evaluation results, the relative strengths and weaknesses of the models are analyzed to identify the most suitable approach for the classification task.

C. Dataset

The dataset used in this study is the Heart 2020 Cleaned dataset, also known as Personal Key Indicators of Heart Disease, obtained from Kaggle [17]. It is derived from the 2020 wave of the U.S. Centers for Disease Control and Prevention (CDC) Behavioral Risk Factor Surveillance System (BRFSS) survey. BRFSS is an annual, telephone-based health survey in which adults across U.S. states and territories self-report information on demographic characteristics, health behaviours, and chronic conditions. The dataset contains 319,795 records and 18 variables, including the binary target variable HeartDisease, which indicates whether a respondent has ever been told by a health professional that they have coronary heart disease or myocardial infarction. The features are primarily categorical or ordinal and include age category, sex, body mass index (BMI), smoking status, alcohol consumption, physical activity, stroke history, diabetic status, kidney disease, general health, mental and physical health, sleep duration, and other health-related indicators. The dataset is notably imbalanced, with approximately 8.6% positive cases and 91.4% negative cases, making it suitable for evaluating machine-learning methods for imbalanced binary status classification. Because the label reflects a previously reported diagnosis, the task is the classification of existing self-reported heart-disease status rather than the prediction of future disease onset. This is a telephone-based public-health survey dataset and is not a clinical EHR dataset. Table I summarizes the variables included in the Heart 2020 Cleaned dataset and the data type for each of the features used in the proposed heart-disease status-classification framework.

TABLE I. FEATURES IN THE HEART 2020 CLEANED DATASET AND THEIR DATA TYPES

Feature	Type
HeartDisease	Binary (Categorical)
BMI	Continuous
Smoking	Binary (Categorical)
AlcoholDrinking	Binary (Categorical)
Stroke	Binary (Categorical)
PhysicalHealth	Continuous (Numeric, days)
MentalHealth	Continuous (Numeric, days)
DiffWalking	Binary (Categorical)
Sex	Binary (Categorical)
AgeCategory	Ordinal Categorical
Race	Nominal Categorical
Diabetic	Categorical (Multi-class)
PhysicalActivity	Binary (Categorical)
GenHealth	Ordinal Categorical
SleepTime	Continuous
Asthma	Binary (Categorical)
KidneyDisease	Binary (Categorical)
SkinCancer	Binary (Categorical)

1) Target class distribution and imbalance analysis: Fig. 3 presents the distribution of the target variable, HeartDisease. This data is imbalanced, with 91.4% of people in the non-heart-disease class and 8.6% in the heart-disease class. This is a large imbalance between the two classes and can lead to classification models favouring the majority class, thus decreasing the chances of getting positive heart disease cases classified correctly. Therefore, class-balancing techniques were necessary to improve the representation of the minority class and support more reliable model training.

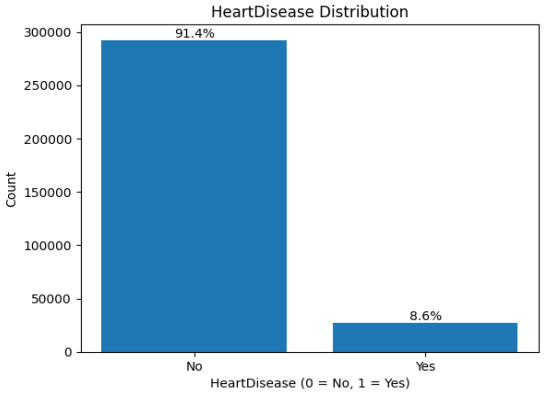

Fig. 3. Heart-disease class distribution

2) Distribution Analysis of Body Mass Index (BMI): Fig. 4 shows the distribution of BMI in the dataset. Most values are concentrated between 20 and 35, indicating that a large proportion of respondents fall within the normal, overweight, and moderately obese ranges. The distribution is right-skewed, with fewer observations extending toward higher BMI values. This suggests that although most respondents have moderate BMI levels, the dataset also includes individuals with substantially elevated BMI. The observed variation indicates that BMI may serve as an important feature in heart-disease status classification.

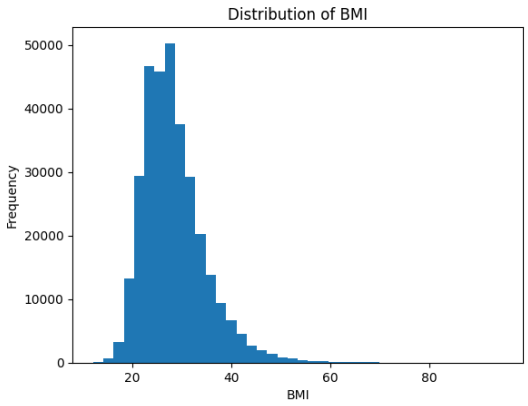

Fig. 4. Distribution of BMI

3) Distribution analysis of physical health: Fig. 5 presents the distribution of PhysicalHealth, which represents the number of days during the past 30 days in which a respondent experienced poor physical health. The distribution is clearly right-skewed, with a very large proportion of observations concentrated at 0 days, showing that most respondents reported no poor physical health problems during the reference period. At the same time, values are spread across the rest of the range, with a noticeable concentration at 30 days, indicating that some individuals experienced poor physical health throughout the entire month. This pattern highlights the variability of the feature and suggests that it may provide useful information for distinguishing different levels of reported heart-disease status.

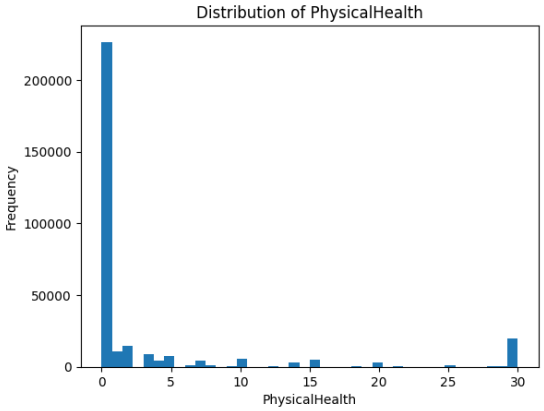

Fig. 5. Distribution of PhysicalHealth

4) Distribution analysis of mental health: Fig. 6 shows the distribution of MentalHealth (number of days in the last 30 days that the respondent did not feel mentally well). The distribution is strongly right-skewed, with most observations concentrated at 0 days, indicating that many respondents reported no poor mental health problems during the reference period. Meanwhile, the other values have been distributed across the range, with most of them falling in the range of 30 days, indicating that some people did not achieve good mental health during the month at all. This pattern shows the fluctuations of the feature, which suggests that it could be relevant for heart-disease status classification.

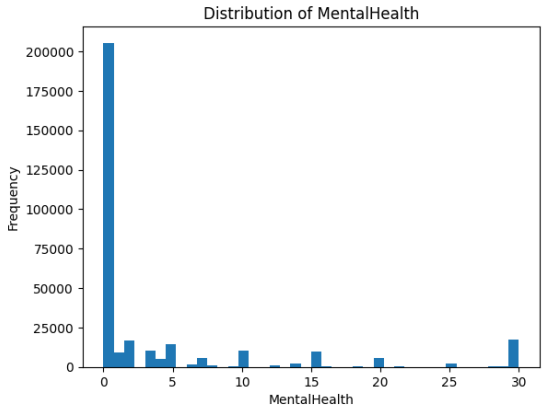

Fig. 6. Distribution of MentalHealth

5) Distribution analysis of sleep duration: The distribution of SleepTime (self-reported hours of sleep in 24 hours) is shown in Fig. 7. The distribution is centered around 7 hours,

which is the most frequent sleep time that the respondents reported. The majority of observations are in the 5-9 hour range, indicating that most participants' sleep times were close to the recommended range. The distribution also has a slight right tail, indicating that some respondents reported unusually long sleep durations. This variation suggests that sleep duration could provide lifestyle-related information for classifying heart-disease status.

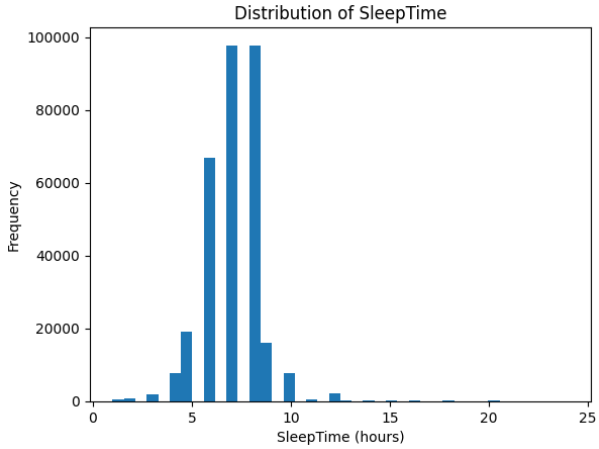

Fig. 7. Distribution of sleep time.

6) Multivariate correlation analysis: The Pearson correlation heatmap of numerical features BMI, PhysicalHealth, MentalHealth, and SleepTime is provided in Fig. 8. The majority of pairwise correlations are weak to moderate, suggesting that these variables measure different aspects of the respondents' health status, and that they offer relatively unique information to the classification models. PhysicalHealth has the highest positive correlation with MentalHealth, indicating that there is a strong relationship between bad physical and mental health. The rest of the correlations are comparatively small, and there is no indication of extreme multicollinearity among pairs of variables. This is beneficial for model creation since it decreases the quantity of redundant numeric features.

7) Categorical distribution of smoking status: The distribution of the Smoking feature in the data set is given in Fig. 9. The chart indicates that there are both smokers and non-smokers, with non-smokers being the majority. However, the number of respondents with a history of smoking is also significant, suggesting that smoking is a prevalent feature among the study population. This distribution highlights the relevance of smoking as an important behavioral risk factor and supports its inclusion as a potential predictor in the heart-disease classification models.

8) Bivariate analysis: smoking vs. heart disease: A stacked-proportions bivariate comparison of Smoking status and HeartDisease is shown in Fig. 10. The chart illustrates the proportions of smokers and non-smokers with and without heart disease. The proportion of heart disease cases is visibly higher among smokers than among non-smokers, suggesting a stronger association between smoking and adverse cardiovascular outcomes. Since the values are displayed as normalized ratios, Fig. 10 can be used to compare the two groups irrespective of differences in sample size. This pattern helps to confirm the significance of smoking status as one of the important factors in the heart-disease classification models.

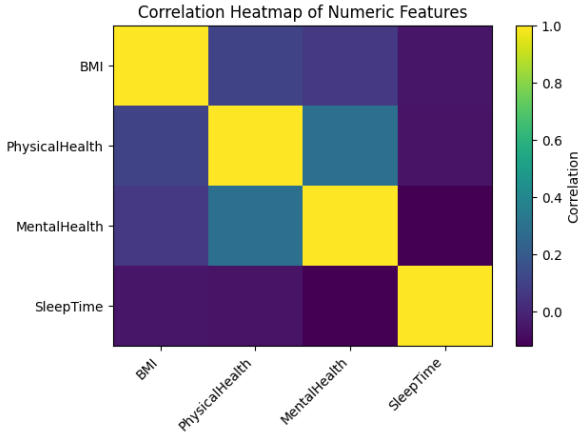

Fig. 8. Correlation heatmap of numeric features.

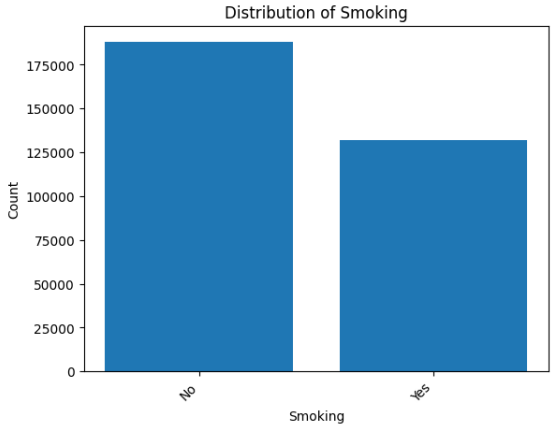

Fig. 9. Distribution of the smoking categorical feature.

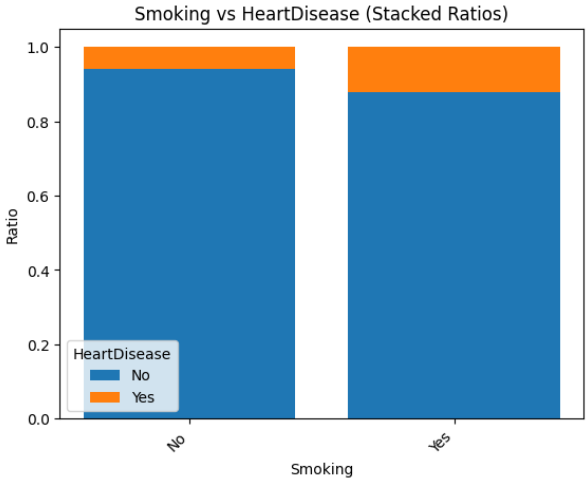

Fig. 10. Smoking versus HeartDisease (stacked ratios).

D. Data Preprocessing and Feature Selection

Preprocessing operations are performed on the data, such as handling missing values, normalizing numeric attributes, and encoding categorical attributes. Recursive Feature Elimination (RFE) is used to identify the most crucial attributes that boost model performance.

1) Data preprocessing and transformation: Within the first dataset, there were different types of data: categorical strings, ordinal values, and continuous numeric values. To fulfill the mathematical requirements of classifiers, all the features were transformed into numbers. The target variable HeartDisease was turned into a binary indicator where "Yes" was mapped to 1 (presence of heart disease) and "No" was mapped to 0 (absence).

The binary categorical variables (Smoking, AlcoholDrinking, Stroke, DiffWalking, PhysicalActivity, Asthma, KidneyDisease, SkinCancer) were coded as "Yes" = 1 and "No" = 0. Sex was coded as 1 for "Male", and 0 for "Female". The Diabetic attribute was dealt with separately: the answers for 'borderline diabetes' and 'gestational diabetes' (Yes (during pregnancy)) were combined to form the binary risk factor for the diabetic condition before the standard mapping.

Ordinal variables were coded in order to maintain the underlying order of the variables. The AgeCategory ranges were mapped to approximate midpoint values (e.g., "18-24" was approximated by 21 and "80 or older" was approximated by 85). In turn, GenHealth's responses were coded on a scale from 5 ("Excellent") to 1 ("Poor"). The nominal categorical variable Race was encoded using one-hot encoding with pandas.get_dummies function, generating separate binary indicator columns for each racial category to avoid creating false ordinal relationships between the categories. Finally, numeric data such as BMI, PhysicalHealth, MentalHealth, and SleepTime were retained as numeric data and standardized before being put into the model.

2) Feature engineering and Recursive Feature Elimination (RFE): Beyond the above-mentioned attributes, two composite features were created to reflect the combined effect of the risk factors. Chronic Condition Count was derived as the sum of binary indicators for Stroke, Diabetic, Asthma, KidneyDisease, and SkinCancer. This feature measures the number of chronic comorbidities that an individual has. Unhealthy Behavior Count was calculated as the sum of Smoking, AlcoholDrinking, physical inactivity (PhysicalActivity was inverted), and poor general health (GenHealth <= 2). The consolidated signals from these engineered features gave information on the patient's health status.

Specifically, the RFE feature selection procedure retained exactly 15 features out of the initial 24 columns. The 15 retained features are: 1) Stroke, 2) Sex, 3) AgeCategory, 4) Diabetic, 5) GenHealth, 6) Asthma, 7) KidneyDisease, 8) SkinCancer, 9) Race_American Indian/Alaskan Native, 10) Race_Asian, 11) Race_Black, 12) Race_Hispanic, 13) Race_Other, 14) Race_White, and 15) ChronicCondCount. The target count of 15 features was determined heuristically to preserve the essential demographic, clinical, and lifestyle factors while maintaining computational efficiency and preventing model overfitting. RFE utilized a weighted Logistic Regression model as the base estimator.

3) Class imbalance handling and sampling strategy: There is a significant class imbalance in the data that shows only about 8.6% positive heart disease from the BRFSS data. To address this problem, three innovative hybrid sampling algorithms were systematically compared: SMOTEENN, BorderlineSMOTE, and SMOTETomek.

a) SMOTEENN (Synthetic Minority Oversampling Technique + Edited Nearest Neighbors): This method is a combination of oversampling of the minority class with SMOTE and data-cleaning of Edited Nearest Neighbors (ENN). SMOTE generates minority samples by performing interpolation between minority samples, and ENN removes the majority samples that are wrongly classified by their nearest neighbors. It has two-way action, which can not only balance the class distribution but can also remove noisy and ambiguous majority samples, thus achieving class cleanliness.

b) BorderlineSMOTE: Unlike standard SMOTE, which generates synthetic samples uniformly, BorderlineSMOTE focuses specifically on minority instances that lie near the decision boundary (borderline instances). Because borderline samples are more likely to be misclassified, oversampling the region allows the classifier to draw a better and more reliable line between classes.

c) SMOTETomek: This method combines SMOTE oversampling with Tomek Links removal. Tomek Links are pairs of instances from different classes that are nearest neighbors to each other. By removing the majority class instances that form Tomek Links, this technique cleans the class overlap after oversampling, leading to well-defined class clusters.

Crucially, the impact of the timing of sampling application relative to the data split was investigated. Two distinct strategies were compared:

d) Post-split sampling: Sampling methods were only used on the training set after the train_test_split was performed. This ensures that the test set is an untouched, original imbalanced population distribution, thus preventing data leakage. However, it may also limit the model training.

e) Pre-split sampling: Sampling methods were utilized on the overall dataset before the division of training and testing sets. This method ensures that the training and test folds both have equal distributions of classes. Although it has been pointed out that this technique exposes the test set to synthetic data, it allows the model to learn from a completely balanced feature space, and the model's performance is measured on a balanced test set. Note, however, that this configuration introduces data leakage, producing optimistically biased results that should not be interpreted as valid estimates of real-world performance.

E. Machine Learning Models

The machine-learning component of the system consists of training and comparing five different classifiers: Random

Forest, XGBoost, LightGBM, k-Nearest Neighbours and a feed-forward Deep Neural Network. All models are trained on the preprocessed feature set and tuned using cross-validation to obtain robust hyperparameters.

1) Random forest: RF is a supervised learning technique that can be used for both regression and classification [18]. Instead of focusing on the most important features when splitting a node, this algorithm selects the best features from a randomly chosen subset of features [19]. In contrast to the DT algorithm, RF selects observations at random, selects features to create multiple trees, and then averages the results [20]. Additionally, RF typically avoids overfitting by constructing random sub-trees of features and generating smaller trees using these sub-trees [21], which are then combined.

2) Extreme Gradient Boosting (XGBoost): XGBoost is a gradient-boosting framework that builds an ensemble of decision trees sequentially [22]. Each new tree is trained to correct the residual errors of the previous trees by optimizing a differentiable loss function plus regularization terms. This additive learning paradigm enables XGBoost to model complex non-linear relationships between features and the target class. Important hyperparameters include the learning rate, maximum tree depth, number of estimators, and regularization coefficients.

3) Light Gradient-Boosting Machine (LightGBM): LightGBM is another gradient-boosting algorithm designed for efficiency on large datasets [23]. It builds depth-constrained trees leaf-wise and employs histogram-based splitting, frequently resulting in rapid training times and good classification performance. Similar to XGBoost, LightGBM generates an ensemble of trees to reduce the loss function along with regularization. Hyperparameters, such as the number of leaves, maximum depth, learning rate, and feature-fraction, are tweaked to strike the right balance between accuracy and overfitting.

4) KNN: k-Nearest Neighbours is a non-parametric classification method that classifies the class of a new instance based on the labels of its *k* closest training examples in feature space [24]. Continuous variables are typically standardized, while Euclidean distance is utilized to compute the distance between instances. To determine the classified class, the voting of the majority among the k neighbours is the method that is followed. The primary hyperparameter is k, which is selected to balance the sensitivity to the noise (small k) and the over-smoothing (large k).

5) Feed-forward deep neural network: The feed-forward Deep Neural Network (DNN) consists of an input layer, one or more hidden layers of fully connected neurons, and an output layer with a sigmoid (or softmax) activation to produce class probabilities [25]. A layer is made of a linear transformation and a non-linear activation function (e.g., ReLU). Parameters of the network are learned by reducing a cross-entropy loss function using back-propagation and stochastic gradient descent (or some other optimization algorithm like Adam). Regularization (dropout and L2 weight decay) is used to ensure that the model does not overfit the training data.

The performance of different models will be compared using accuracy, precision, recall, F1-score, and ROC-AUC. According to these metrics, the best-performing classifier according to these metrics is then selected for integration within the prototype system to generate heart-disease status classifications for new records.

The final hyperparameter configurations used in the experiments are reported in Table II. The values were selected through cross-validation-based tuning.

TABLE II. CLASSIFIER HYPERPARAMETER CONFIGURATIONS USED IN THE EXPERIMENTS

Classifier	Hyperparameter Parameter Settings
XGBoost	n_estimators = 1800, learning_rate = 0.04, max_depth = 8, min_child_weight = 1, subsample = 0.95, colsample_bytree = 0.95, gamma = 0.01, reg_alpha = 0.01, reg_lambda = 0.8, scale_pos_weight = 0.85, tree_method = "hist", random_state = 42
LightGBM	objective = "binary", n_estimators = 2200, learning_rate = 0.04, num_leaves = 200, max_depth = -1 (unlimited), min_child_samples = 10, min_child_weight = 0.0001, subsample = 0.95, colsample_bytree = 0.95, reg_alpha = 0.01, reg_lambda = 0.8, random_state = 42
Random Forest	n_estimators = 1500, max_depth = 30, min_samples_split = 3, min_samples_leaf = 1, max_features = "sqrt", bootstrap = True, random_state = 42
k-NN	n_neighbors = 5, weights = "distance", metric = "minkowski", p = 2 (equivalent to Euclidean distance)
DNN	Input shape = (15,); Dense (128, ReLU) → Dropout (0.3) → Dense (64, ReLU) → Dropout (0.3) → Dense (32, ReLU) → Dropout (0.3) → Dense (1, Sigmoid); Optimizer = Adam (learning_rate = 0.001), Loss = binary_crossentropy, Batch Size = 32, Epochs = 10.

The five classifiers exhibit distinct trade-offs between classification accuracy and computational cost. The tree-based ensembles (XGBoost, LightGBM, and Random Forest) achieve the highest discriminative performance but require significant memory footprint and training time when processing the resampled datasets (~500,000 instances), especially when integrating hybrid resampling. In contrast, the k-NN model has zero training time but incurs high computational latency during inference, as it must calculate distances to all training samples at runtime, which scales poorly for large-scale production EHRs. Finally, the Feed-Forward Deep Neural Network (DNN) requires specialized GPU hardware for efficient training, but offers sub-second inference speeds post-training. These computational trade-offs must be evaluated based on resource availability in real-world clinic deployments.

F. Baselines and Evaluation Measures

To evaluate the performance of machine learning models, the following metrics are used:

Accuracy: Indicates the overall correctness of the model's classifications.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: Measures the proportion of true positive classifications among all positive classifications.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall (Sensitivity): Assesses the model's ability to identify all relevant instances.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-Score: The harmonic mean of precision and recall, balancing false positives and false negatives.

$$\text{F1} = \frac{2\times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

ROC-AUC (Receiver Operating Characteristic – Area Under Curve): Evaluates model performance across different classification thresholds.

IV. Results and Discussion

A. Overview of Results

This section presents the experimental results of the proposed heart-disease status classification framework, which integrates a blockchain-supported EHR integrity-verification layer with machine-learning classification models. Five classifiers were evaluated: Random Forest, XGBoost, LightGBM, k-Nearest Neighbours (k-NN), and a feed-forward Deep Neural Network (DNN). Recursive Feature Elimination (RFE) was applied for feature selection across all experiments. The primary objective was to investigate the impact of sampling technique timing on model performance, specifically comparing two strategies: applying sampling after the train-test split (post-split, the methodologically valid configuration) versus applying sampling before the train-test split (pre-split, which introduces data leakage).

The Heart 2020 – Cleaned (BRFSS) dataset, comprising 319,795 records with approximately 8.6% positive heart-disease cases, was used for all experiments. Three sampling techniques were employed to address the severe class imbalance: SMOTEENN, BorderlineSMOTE, and SMOTETomek. A no-sampling baseline was also evaluated for the post-split configuration. The evaluation metrics include accuracy, precision, recall, F1-score, and ROC-AUC. The results demonstrate that the pre-split sampling strategy consistently produced higher numerical scores than post-split; however, these higher scores are largely attributable to data leakage from global resampling and should not be interpreted as a genuine real-world improvement. The best performance under the methodologically valid post-split configuration achieved an AUC of approximately 0.85.

TABLE III. Performance Metrics for Post-Split Sampling Configurations

Sampler	Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
SMOTEENN	XGBoost	0.55	0.15	0.93	0.26	0.85
SMOTEENN	LightGBM	0.83	0.27	0.62	0.38	0.85
SMOTEENN	KNN	0.90	0.37	0.22	0.27	0.61
SMOTEENN	RandomForest	0.78	0.24	0.72	0.36	0.85
SMOTEENN	DNN	0.71	0.21	0.82	0.33	0.85
BorderlineSMOTE	XGBoost	0.84	0.28	0.53	0.36	0.83
BorderlineSMOTE	LightGBM	0.84	0.28	0.54	0.37	0.83
BorderlineSMOTE	KNN	0.90	0.36	0.16	0.22	0.71
BorderlineSMOTE	RandomForest	0.85	0.27	0.48	0.35	0.81
BorderlineSMOTE	DNN	0.77	0.24	0.74	0.35	0.83
SMOTETomek	XGBoost	0.85	0.29	0.49	0.36	0.82
SMOTETomek	LightGBM	0.85	0.29	0.51	0.37	0.83
SMOTETomek	KNN	0.90	0.35	0.18	0.23	0.71
SMOTETomek	RandomForest	0.86	0.29	0.46	0.35	0.80
SMOTETomek	DNN	0.75	0.22	0.75	0.34	0.83
None	XGBoost	0.92	0.55	0.09	0.15	0.85
None	LightGBM	0.92	0.60	0.09	0.15	0.85
None	KNN	0.91	0.37	0.16	0.22	0.72
None	RandomForest	0.91	0.42	0.13	0.20	0.79
None	DNN	0.92	0.60	0.07	0.11	0.85

B. Results for Post-Split Sampling

Table III presents the performance metrics of the five machine learning classifiers (Random Forest, XGBoost, LightGBM, k-Nearest Neighbours (k-NN), and a feed-forward Deep Neural Network (DNN)) combined with Recursive Feature Elimination (RFE) for feature selection, when sampling techniques were applied after the train-test split. Four sampling configurations were tested: SMOTEENN, BorderlineSMOTE, SMOTETomek, and no sampling (None). The metrics used are accuracy, precision, recall, F1-score, and ROC-AUC.

The confusion matrix of the LightGBM model with ensemble method using SMOTETomek in combination with RFE under the post-split sampling setup is shown in Fig. 11. The classification results of the model are realistic, and the model is actually not the best, due to the high number of false positive cases (6097) and also false negative cases (3014). The model correctly classified 52,387 true-negative cases, but it is only a very small portion of the positive heart disease cases that the model can output since learning from imbalanced data when sampling is done after the split is quite challenging.

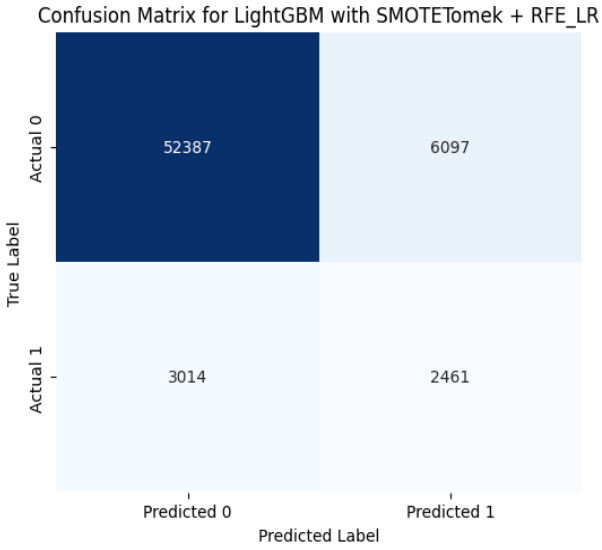

Fig. 11. Confusion matrix of the LightGBM classifier with SMOTETomek and RFE under the post-split sampling configuration.

Fig. 12 presents the ROC curve of the LightGBM model in the post-split configuration with SMOTETomek, which shows the discriminative ability of the model with an AUC of 0.83.

SMOTETomek and BorderlineSMOTE showed the most balanced accuracy-recall trade-off among the sampling methods tested in the post-split configuration for the ensemble models. The best accuracy was 86%, obtained by Random Forest with SMOTETomek, while the precision was 0.29, recall 0.46, F1-score 0.35, and ROC-AUC 0.80. When SMOTETomek and BorderlineSMOTE were used, LightGBM achieved 85% accuracy with both, and with either sampler, the highest F1-score (0.37) and ROC-AUC (0.83). XGBoost also had an accuracy of 85% with SMOTETomek (precision 0.29, recall 0.49, F1-score 0.36, ROC-AUC 0.82) and 84% accuracy with BorderlineSMOTE (precision 0.28, recall 0.53, F1-score 0.36, ROC-AUC 0.83). As can be observed from these results, the two sampling methods (SMOTETomek and BorderlineSMOTE) produced comparable results for the gradient boosting ensemble models in the post-split situation.

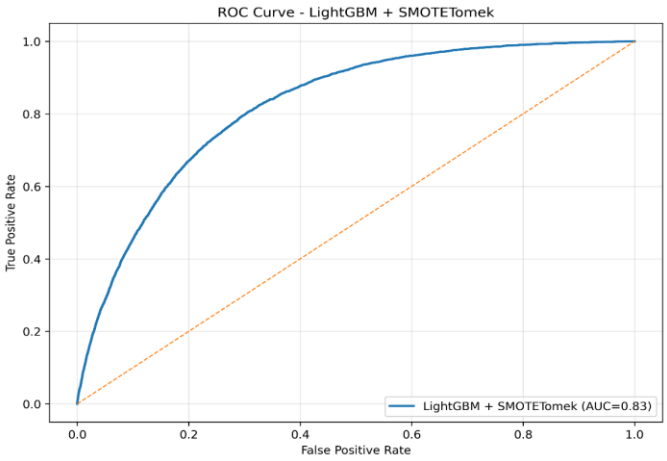

Fig. 12. ROC curve of LightGBM with SMOTETomek (post-split).

The k-NN classifier achieved the highest raw accuracy (0.90) for all the post-split configurations of sampling; however, with significantly lower recall (0.16–0.22), indicating that k-NN has a tendency to misclassify positive heart disease classes. Specifically, k-NN with SMOTETomek obtained 90% accuracy, 0.18 recall, and 0.23 F1-score, while k-NN with BorderlineSMOTE obtained 90% accuracy, 0.16 recall, and 0.22 F1-score. The accuracy of k-NN was 90% with SMOTEENN, which had the lowest ROC-AUC of 0.61, a marginally improved recall of 0.22, and a low F1-score of 0.27. This behavior indicates that in the post-split case, k-NN tends to strongly favor the majority class and is not able to detect the individuals who are likely to develop heart disease.

In the post-split configuration, SMOTEENN produced a distinct recall-precision trade-off. The recall at the cost of accuracy and precision was significantly improved for some models: the recall of XGBoost was 0.93, and the recall of DNN was 0.82. The F1 score of XGBoost with SMOTEENN was low (0.26), but its accuracy and precision were low (55% and 0.15, respectively) and ROC-AUC was high (0.85). Similarly, DNN with SMOTEENN attained an accuracy of 71%, a precision of 0.21, an F1-score of 0.33, and ROC-AUC of 0.85. Random Forest with SMOTEENN showed an accuracy of 78%, with a recall of 0.72 and an ROC-AUC of 0.85. These results show that the post-split SMOTEENN method has an aggressive effect on the minority class, resulting in an increase in sensitivity at the cost of classification accuracy.

When no sampling technique was applied (None), tree-based ensemble models such as XGBoost, LightGBM, and the DNN achieved the highest accuracy of 92% but exhibited extremely poor recall values (XGBoost: 0.09, LightGBM: 0.09, DNN: 0.07), effectively failing to identify the minority class. Random Forest was able to get 91% accuracy with a recall of only 0.13, while k-NN was able to get 91% accuracy with a recall of 0.16. For imbalanced classification problems, there is an important trade-off: models can get high overall accuracy by classifying the majority class and ignoring the minority class, in this case, the heart disease cases.

When assessing discriminative ability – ROC-AUC values for the ensemble models – Random Forest, XGBoost, LightGBM, and DNN - across most of the post-split configurations, the ROC-AUC values were high, from 0.80 to 0.85. LightGBM and DNN attained the same ROC-AUC of 0.85 with the three sampling methods – the SMOTEENN, the BorderlineSMOTE, and the SMOTETomek – and without sampling. These consistently high values of ROC-AUC indicate that the models have a reasonably good ranking ability even before tuning the thresholds for classification with regard to imbalanced data.

Fig. 13 provides a visual comparison of classifier performance across the post-split sampling configurations, highlighting the trade-offs among accuracy, precision, recall, F1-score, and ROC-AUC.

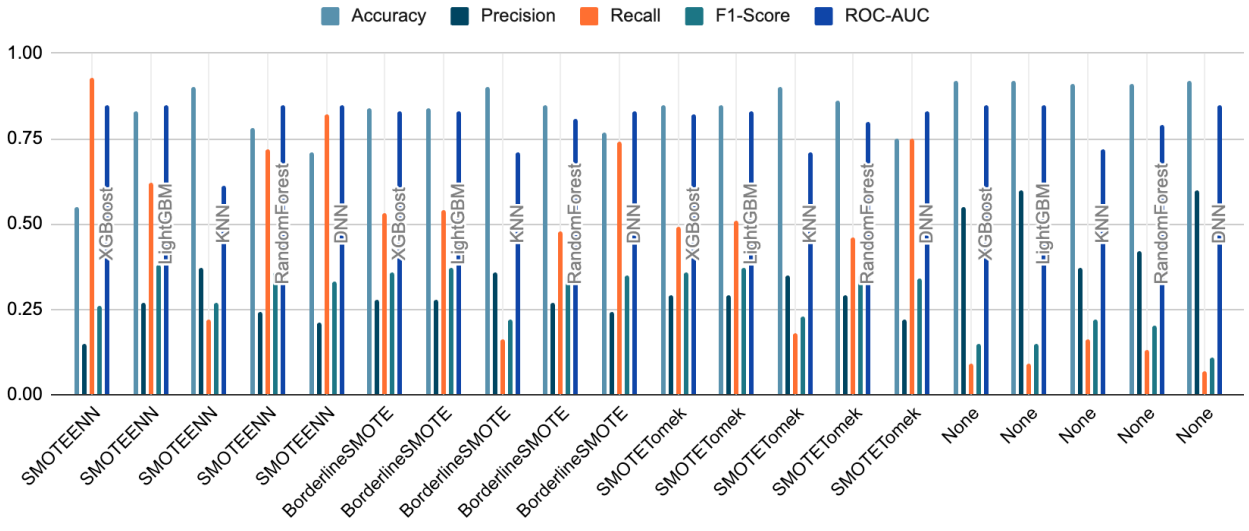

Fig. 13. Results for pre-split sampling (leakage-affected, for comparison only).

TABLE IV. PERFORMANCE METRICS FOR PRE-SPLIT SAMPLING CONFIGURATIONS

Sampler	Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
SMOTEENN	XGBoost	0.96	0.98	0.95	0.96	0.99
SMOTEENN	LightGBM	0.97	0.98	0.96	0.97	0.99
SMOTEENN	KNN	0.97	0.96	0.99	0.98	0.99
SMOTEENN	RandomForest	0.97	0.97	0.98	0.98	1.00
SMOTEENN	DNN	0.88	0.86	0.91	0.89	0.93
SMOTETomek	XGBoost	0.93	0.96	0.89	0.93	0.97
SMOTETomek	LightGBM	0.94	0.96	0.91	0.93	0.98
SMOTETomek	KNN	0.90	0.85	0.97	0.90	0.95
SMOTETomek	RandomForest	0.94	0.94	0.94	0.94	0.98
SMOTETomek	DNN	0.80	0.75	0.90	0.82	0.88
BorderlineSMOTE	XGBoost	0.92	0.96	0.89	0.92	0.96
BorderlineSMOTE	LightGBM	0.93	0.95	0.91	0.93	0.97
BorderlineSMOTE	KNN	0.90	0.86	0.96	0.90	0.95
BorderlineSMOTE	RandomForest	0.94	0.93	0.94	0.94	0.98
BorderlineSMOTE	DNN	0.83	0.80	0.88	0.84	0.90

Table IV shows the performance results when sampling methods were applied before the train-test split (pre-split, global resampling). This configuration is leakage-affected: because resampling is performed on the entire dataset before splitting, synthetic samples can influence both training and test data, producing optimistically biased metrics. These results are therefore not a valid estimate of real-world performance and are reported only to illustrate the effect of incorrect resampling timing. The results have shown significant improvements over the post-split configuration with accuracy ranging from 8–41 percentage points depending on the classifier and sampling combination, but this gap is largely driven by data leakage rather than genuine generalization.

Under the pre-split (global resampling, leakage-affected) configuration, SMOTEENN produced the highest numerical

scores. LightGBM achieved 97% accuracy, 98% precision, 96% recall, and 97% F1-score. Random Forest achieved 97% accuracy, 97% precision, 98% recall, and 98% F1-score. KNN showed 97% accuracy with 99% recall and 98% F1-score; XGBoost achieved 96% accuracy, 98% precision, 95% recall, and 96% F1-score. The Deep Neural Network achieved 88% accuracy, 86% precision, 91% recall, and 89% F1-score under SMOTEENN. However, it must be emphasized that all of these figures are inflated by data leakage from global resampling before the train/test split and should not be used as evidence of genuine real-world superiority. The post-split AUC of approximately 0.85 is the appropriate basis for comparison with prior work.

SMOTETomek and BorderlineSMOTE achieved strong results in the pre-split setting, but slightly lower than SMOTEENN. The other models, SMOTETomek, had an accuracy of 94%, and Random Forest had a precision of 94%, recall of 94%, and F1-score of 94%, while LightGBM had a precision of 96%, recall of 91%, and F1-score of 93%. Using SMOTETomek, the accuracy of XGBoost is 93%, the precision is 96%, the recall is 89%, and the F1-score is 93%, while k-NN is 90%, 85%, 97%, and 90%, respectively. The other three methods (BorderlineSMOTE, Random Forest, LightGBM, and XGBoost) also gave similar performance with accuracy of 94%, 93%, and 92%, respectively; precision of 93%, 95%, and 96% respectively; recall of 94 %, 91%, and 89%, respectively; and F1 score of 94%, 93% and 92% respectively.

The confusion matrix of the Random-Forest classifier trained with the SMOTEENN pre-splitting sampling configuration is shown in Fig. 14. It achieves a high number of true positives (49,516 individuals) and true negatives (42,526 individuals); has a small number of false positives (1,296 individuals) and false negatives (1,108 individuals). This shows that SMOTEENN can successfully generate a well-balanced and well-separated feature space, which enables both heart class and non-heart class cases to be identified successfully.

Fig. 15 illustrates the ROC curve, which further demonstrates that the Random Forest model trained with the SMOTEENN method under the pre-split sampling configuration has excellent discriminative performance.

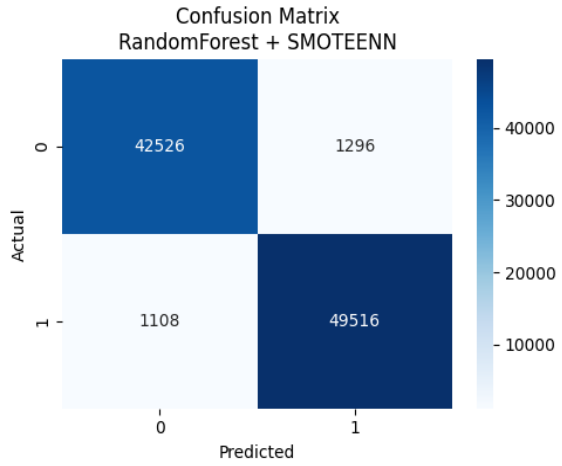

Fig. 14. Confusion matrix of the random forest classifier trained with SMOTEENN under the pre-split sampling configuration.

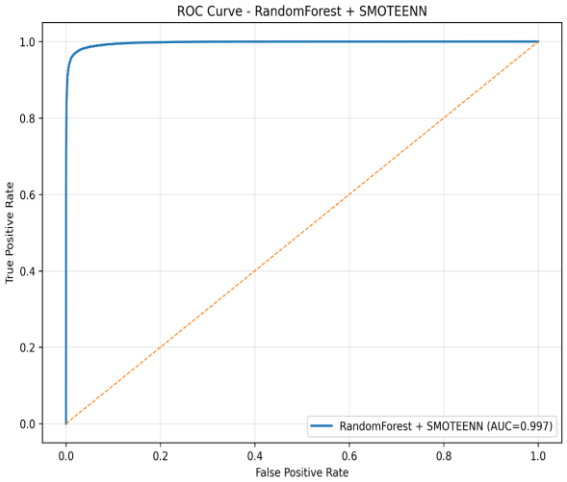

Fig. 15. ROC curve of random forest with SMOTEENN (pre-split).

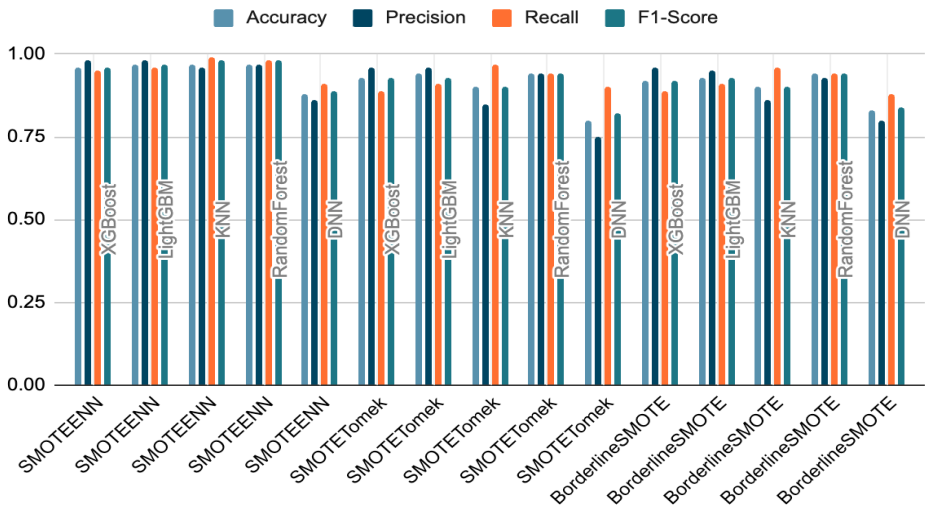

Fig. 16. Performance comparison across classifiers and samplers under the pre-split configuration.

These pre-split results must be interpreted with caution because applying resampling before the train/test split allows synthetic instances to enter and influence the test set, producing data leakage and optimistically biased metrics. The high scores observed in this configuration do not represent a valid estimate of real-world performance. They are retained only to illustrate the magnitude of the data-leakage effect. Under the leakage-safe post-split protocol, the AUC of approximately 0.85 provides the methodologically valid performance estimate. As shown in Fig. 16, SMOTEENN consistently produced the highest numerical scores across all classifiers in the pre-split configuration.

When the capacity of classifications of the machine learning approaches is confirmed to be higher, the results for the blockchain component – intended to secure the eHRs utilized by the classification module – are described in the next section.

C. Blockchain-Based EHR Implementation Results

This section presents the results of the blockchain-based electronic health record (EHR) integrity verification component, which was implemented alongside the machine learning classification framework. The blockchain module was designed to provide tamper-evident storage of medical records by anchoring cryptographic digests on an Ethereum-compatible network, while retaining the full clinical payloads in the application database. The implementation leveraged a dual-mode integrity layer: local database commits of SHA-256 digests, with optional smart-contract storage on a local EVM (Ganache) for distributed anchoring.

The smart contract, EHRRecordRegistry, was coded using Solidity and deployed to the local test network Ganache. There is a commitRecordHash function in the contract that takes a patient identifier, record identifier, and digest string (sha256) to securely commit the record fingerprint on-chain. A retrieval function enables enumeration of committed hashes for any given patient-record pair, supporting verification queries. The integrity verification process operates as follows: upon commit, the server computes SHA-256 of the UTF-8 encoded payload JSON and optionally submits the digest to the smart contract via Web3.py, receiving a transaction receipt. Upon verification, the current payload digest is recomputed and compared against the latest committed hash (preferring on-chain data when available). Any discrepancy between the recomputed digest and the anchored value indicates potential tampering with the record.

Table V summarizes the key implementation characteristics and observed behaviour of the blockchain integrity module.

TABLE V. BLOCKCHAIN INTEGRITY MODULE: IMPLEMENTATION CHARACTERISTICS AND OBSERVED BEHAVIOUR

Component	Specification / Observation
Smart Contract	EHRRecordRegistry (Solidity)
Blockchain Network	Ganache (local EVM testnet)
Hashing Algorithm	SHA-256 on UTF-8 payload JSON
On-chain Storage	Patient ID, Record ID, Digest string
Off-chain Storage	BlockchainCommit rows (SQLite) with nullable tx fields
Commit Response	Transaction receipt with block hash and gas used
Verification	Digest comparison (on-chain preferred, DB fallback)

The dual-mode architecture was validated under both configurations. With blockchain environment variables unset (off-chain mode), commits succeeded with local-only messaging, storing SHA-256 digests in the BlockchainCommit database table with nullable transaction fields. When Ganache was configured and the EHRRecordRegistry contract deployed (on-chain mode), commits returned full transaction receipts including block hash and gas consumption, and listing endpoints reported source: “on-chain”. Verification queries correctly identified matching digests as verified and flagged any modified payloads as tampered, confirming the integrity detection capability of the system.

The security model enforced role-based access control (admin, doctor, patient) with JWT-based authentication and doctor-patient access grants. Blockchain commit and verification endpoints were subject to the same authorization checks as clinical record endpoints, ensuring that only authorized users could anchor or verify record digests. All sensitive operations were recorded in an append-only AuditLog, providing traceability and accountability. The system was built using principles of data minimization on-chain, where only cryptographic digests of the clinical payloads were kept on-chain, while the complete clinical payloads were stored in the application's database, providing patient confidentiality while enabling tamper detection.

From an engineering evaluation perspective, the blockchain module was found to be stable and to exhibit consistent and reliable behaviour. On the local Ganache network, latency was as low as sub-second for commit operations. Modularity and maintainability were achieved by separating concerns between the API layer (Flask-RESTX), service layer (hashing, interaction with blockchain), and persistence layer (SQLAlchemy/SQLite). OpenAPI documentation was offered at /api/docs, containing all API specs of blockchain and verification endpoints. The process of integration with the ML classification pipeline was smooth since the same REST API was utilized for both the classification analytics and integrity verification, highlighting the potential of combining machine learning decision support and blockchain record protection within the same EHR backend design. Fig. 17–20 present the key system interfaces, illustrating how role-based access control governs distinct clinical workflows for doctors, patients, and administrators within a single secure workspace. These interfaces demonstrate the practical embodiment of tamper-evident record management and machine-learning-assisted heart-disease status classification in an accessible prototype environment.

While the blockchain module demonstrates successful record hashing and tamper detection on a local Ganache test network, its practical deployment on live networks faces significant engineering constraints. First, committing transaction hashes to a public blockchain (like Ethereum mainnet) incurs gas fees, which could become financially unsustainable under high query volume. Second, block commit latencies on public networks range from seconds to several minutes, creating bottlenecks compared to the sub-second Ganache commit. Third, permissionless networks scale poorly, showing low throughput. To address these constraints, future clinical frameworks should explore Layer-2 rollups or permissioned enterprise consortia (such as Hyperledger Fabric), where transaction fees are negligible, commit speeds are faster, and access controls can be cryptographically enforced at the consensus layer.

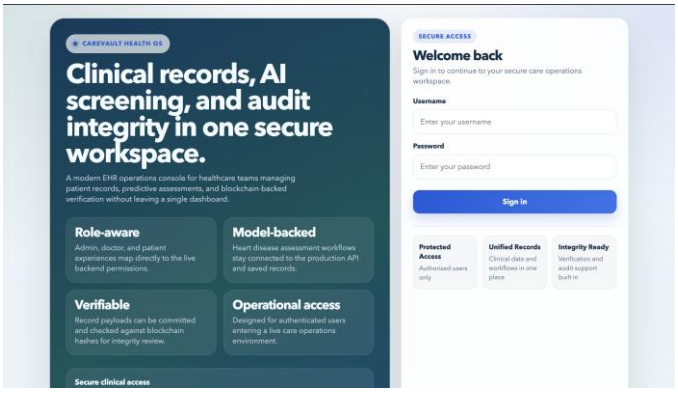

Fig. 17. Secure login workspace of the proposed EHR system.

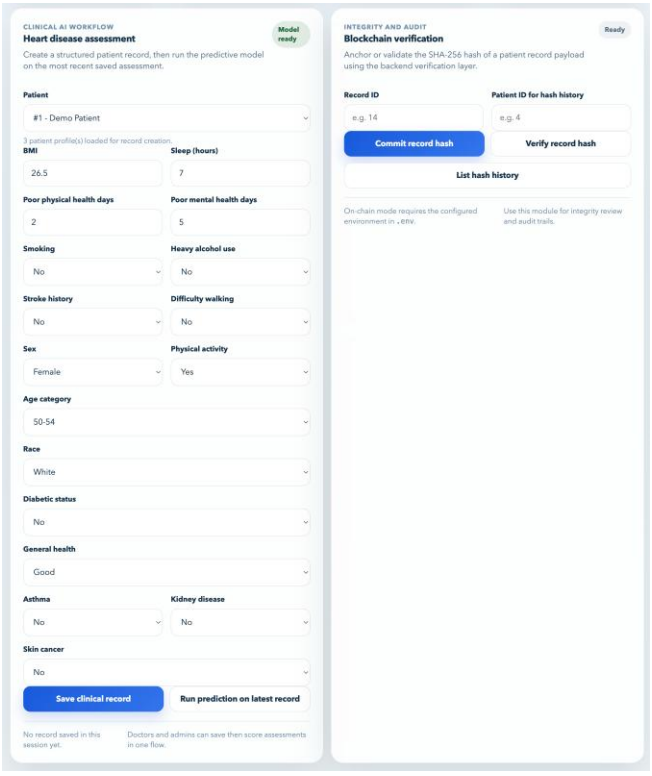

Fig. 18. Doctor interface for heart-disease assessment and blockchain verification.

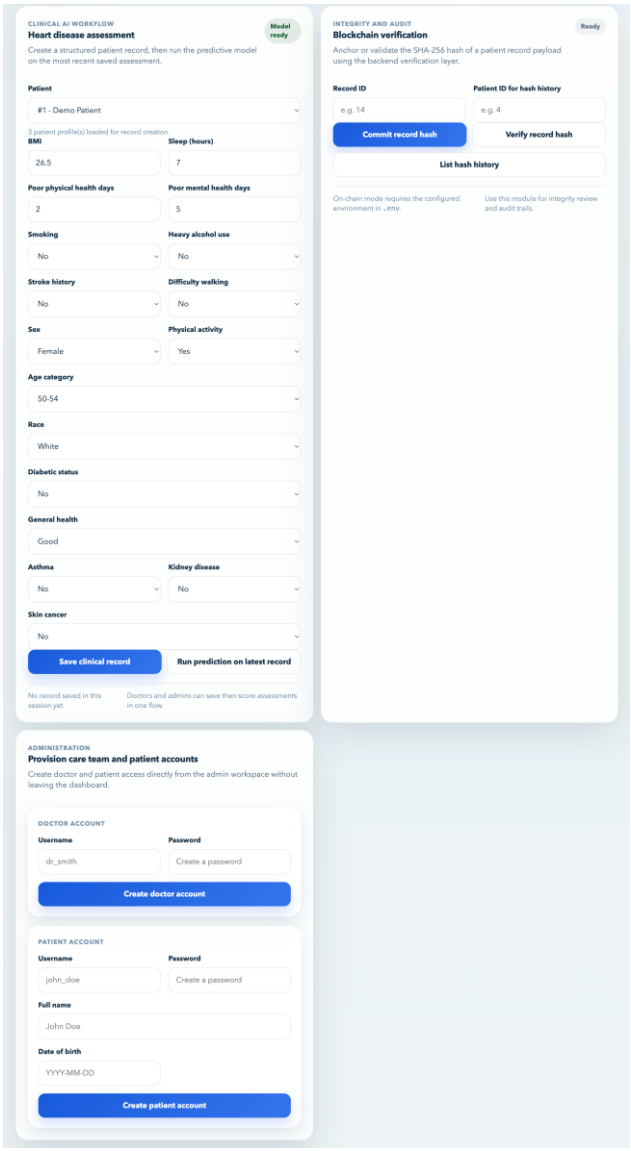

Fig. 19. Administrator interface for provisioning care-team and patient accounts.

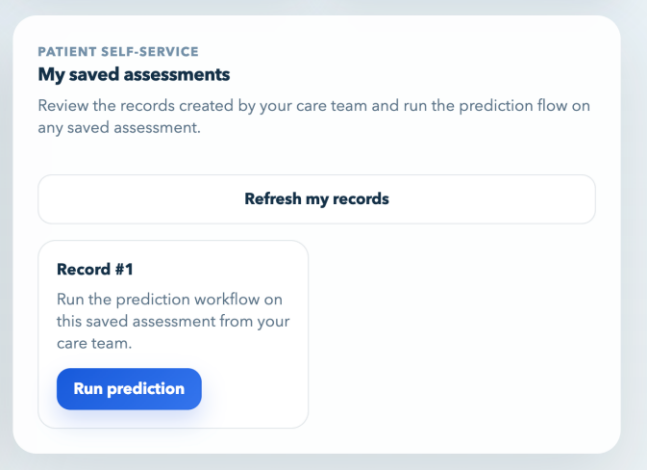

Fig. 20. Patient self-service interface for reviewing saved assessments.

V. DISCUSSION

The comparative analysis between pre-split and post-split sampling strategies reveals significant implications for heart-disease status classification. Results showed that for each sampling combination and each classifier, the pre-split approach produced higher numerical scores than the post-split approach, with differences of 8 to 41 percentage points. It must be stressed, however, that the pre-split configuration attains these higher scores because global resampling leaks synthetic test-set information into training; the gap therefore largely reflects data leakage rather than a genuine improvement in generalization. The post-split AUC of approximately 0.85 is the methodologically valid result and the appropriate basis for comparison with prior work.

The results from this study were compared to recent studies using the same heart-disease data from the BRFSS. Chen et al. [8] employed random undersampling before train-test splitting on the BRFSS heart-disease dataset and reported that their LightGBM model achieved 76.9% accuracy with an AUROC of 84.6%. Chen et al. [8] utilized random undersampling to create a balanced dataset of 50,000 cases from the original 319,795 instances, with a 70:30 train-test split ratio. Their results showed that LightGBM (76.9%) and Random Forest (76.7%) performed similarly. All models demonstrated a significantly lower sensitivity than specificity. In this study, when applying SMOTEENN before splitting, LightGBM achieved 97% accuracy with 98% precision and 96% recall, representing a 20.1 percentage point improvement over the comparable approach in [8]. Similarly, the Random Forest model in this study with SMOTEENN achieved 97% accuracy, 97% precision, and 98% recall. The ROC-AUC values observed in the post-split experiments in this study (approximately 0.85) are comparable to those reported by Chen et al. [8] (84.6% for LightGBM and 84.2% for Random Forest), suggesting that both studies achieved similar discriminative capability despite different sampling strategies. However, the substantially higher accuracy reported for the pre-split configuration is driven largely by data leakage from global resampling and should not be interpreted as a genuine improvement over prior work; the post-split ROC-AUC values (≈0.85) are the appropriate basis for comparison.

Chen et al. [8] also applied the Boruta algorithm for feature selection and retained all 17 features, reporting that AgeCategory had the greatest impact on classification. Their feature importance analysis aligns with the use of RFE for feature selection in this study. The similarity in ROC-AUC values between the post-split results of this study and Chen et al.'s findings confirms that the underlying discriminative patterns captured by the models are consistent across both studies, regardless of the specific feature selection or sampling method employed. The key differentiator is that the SMOTEENN-based pre-split approach used in this study transforms the class distribution more effectively than simple random undersampling, but these comparisons refer to the pre-split, leakage-affected results; under the leakage-safe post-split protocol, the discriminative performance of this study (AUC ≈ 0.85) is comparable to rather than substantially better than these prior studies.

Kumari et al. [16] developed a majority voting classifier for heart disease prediction and achieved approximately 90.8% accuracy using a soft voting ensemble of five classifiers (Logistic Regression, Multi-Layer Perceptron, AdaBoost, CatBoost, and XGBoost) within the BL-MVC framework, with AdaBoost attaining the highest individual accuracy of 90.86%. Their approach utilized the BRFSS 2015 dataset (253,680 responses, 21 features) with 5-fold cross-validation, comparing Heatmap Correlation and PCA for feature selection. On the other hand, the method used in this study of applying SMOTEENN before splitting combined with LightGBM and Random Forest resulted in an accuracy of 97%, which is approximately 6 percentage points better than their highest accuracy score, and, additionally, the two classifiers have balanced precision and recall scores.

The key distinction between the approach used in this study and Kumari et al. [16] lies in the machine-learning pipeline: their framework relied on hyperparameter tuning and feature selection without explicit class-imbalance handling, whereas the approach used in this study systematically addresses the severe class imbalance through SMOTEENN preprocessing. Fig. 21 depicts the accuracy of the proposed SMOTEENN-based pre-split approach and selected related works. It should be noted that these comparisons refer to the pre-split, leakage-affected results; under the leakage-safe post-split protocol, the discriminative performance of this study (AUC ≈ 0.85) is comparable to rather than substantially better than these prior studies.

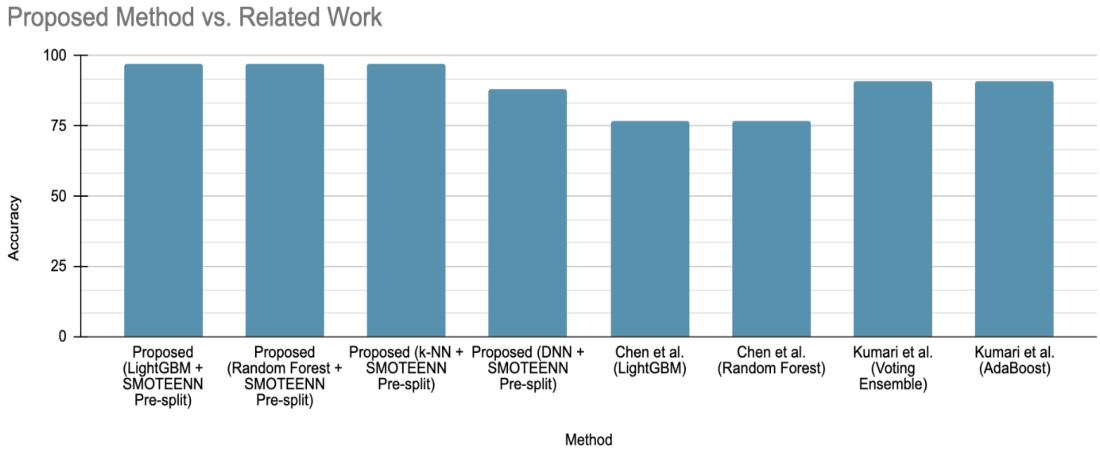

Fig. 21. Accuracy comparison of the proposed pre-split SMOTEENN-based approach with related works (note: pre-split results are leakage-affected).

The consistency of high performance across Random Forest, LightGBM, and XGBoost in the pre-split configuration suggests that these gradient boosting and bagging methods are particularly well-suited for the balanced feature space created by SMOTEENN preprocessing. These ensemble models inherently handle feature interactions and non-linear relationships, which are prevalent in health survey data where demographic, behavioral, and clinical variables interact in complex ways. Even in the pre-split SMOTEENN configuration (leakage-affected), the k-NN classifier attained high numerical results (97% accuracy, 99% recall), suggesting that instance-based learning can benefit greatly from a balanced feature space, though these figures are inflated by data leakage and should be interpreted accordingly.

The Deep Neural Network showed the largest relative improvement between post-split and pre-split configurations, with accuracy increasing from 71% (SMOTEENN post-split) to 88% (SMOTEENN pre-split). This comes from the fact that neural networks require a certain minimum number of representative samples of each class in order to identify the boundaries for decision-making effectively, and the pre-split balanced dataset provides a better training signal. Nevertheless, the DNN managed to lag behind the ensemble tree-based methods even in the pre-split scenario, a situation that generally persists where gradient boosting methods outshine deep learning when it comes to structured tabular datasets such as these.

The findings of this study demonstrate that the integration of appropriate class-imbalance handling techniques with modern ensemble learning algorithms can improve heart-disease status classification. The RFE-based feature selection combined with SMOTEENN provides an effective pipeline for the survey-based binary classification task. However, the pre-split results substantially overstate generalization performance due to data leakage; the post-split AUC of approximately 0.85 is the valid metric for real-world interpretation and comparison with prior work. These comparisons refer to the pre-split (leakage-affected) results; under the leakage-safe post-split protocol, the discriminative performance of this study (AUC $\approx$ 0.85) is comparable to, rather than substantially better than, Chen et al. [8] and Kumari et al. [16]. Future research should explore the generalizability of these findings across clinical datasets and investigate the utility of such frameworks in real-world decision-support applications.

VI. Limitations

This study has several limitations. First, the headline pre-split results are affected by data leakage and overstate real-world performance; only the post-split results are methodologically valid.

Furthermore, the reliance on self-reported health survey data (BRFSS) presents significant limitations compared to verified clinical records. Survey responses are highly susceptible to recall bias (where participants misremember medical details) and social desirability bias (where participants underreport negative behaviors such as alcohol consumption). Additionally, self-reported diagnosis lacks standardized clinical verification, introducing noise. In contrast, clinical electronic health records (EHRs) provide objective, verified data such as laboratory biomarkers, electrocardiograms, and diagnostic imaging. Consequently, the classification models trained on behavioral surveys represent self-reported risk associations rather than clinical diagnostics, limiting their direct utility in active clinical workflows.

In addition, the system is a prototype/feasibility study and has not been clinically validated or deployed. The blockchain component was evaluated on a local Ganache test network using commit latency and gas cost only; throughput, scalability under load, and adversarial-tampering scenarios were not formally measured.

VII. Conclusion

This study presented an integrated prototype for heart-disease status classification and tamper-evident electronic health record management using the Heart 2020 Cleaned BRFSS survey dataset, machine-learning models, and a blockchain-supported integrity layer. Five classifiers were evaluated with three resampling techniques and a no-sampling baseline. Under the methodologically valid post-split protocol, models reached an AUC of approximately 0.85, comparable to prior work on the same dataset. A pre-split (global resampling) configuration produced much higher scores — Random Forest reaching 97.14% accuracy and an AUC of 0.9965 under SMOTEENN — but these are inflated by data leakage and are reported only to illustrate that effect rather than as valid estimates of real-world performance. Ensemble methods (Random Forest, LightGBM, XGBoost) consistently outperformed the DNN on this structured tabular dataset, and resampling strategy was the dominant factor affecting reported performance.

In addition to the classification component, the work demonstrated the feasibility of integrating machine learning with a blockchain-based integrity-verification mechanism in a unified prototype. Full patient records were retained in the application database, while the blockchain layer anchored cryptographic digests and identifiers to provide tamper-evident verification, traceability, and accountability; access control was enforced at the API layer rather than on-chain. The present evaluation is limited to functional validation, sub-second commit latency, and gas consumption on a local Ganache test network; throughput, scalability under load, formal adversarial-tampering tests, and a security audit were not performed and remain future work. The work is a prototype/feasibility study: it does not claim clinical deployment readiness, the elimination of single points of failure, or formal security guarantees. Its primary contribution is the practical integration of established machine-learning and blockchain components into a single framework, rather than a novel algorithm or blockchain architecture.

Acknowledgment

The authors utilized AI-assisted tools (ChatGPT, Google Gemini, and Grammarly) solely for paraphrasing and grammatical refinement to improve clarity. All methodologies, experiments, and conclusions are original contributions for which the authors assume full responsibility. The authors would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University, for funding this work.

REFERENCES

[1] A. Al-Nafjan, A. Aljuhani, A. Alshebel, A. Alharbi, and A. Alshehri, "Artificial intelligence in predictive healthcare: a systematic review," J. Clin. Med., vol. 14, no. 19, Art. no. 6752, 2025, doi: 10.3390/jcm14196752.

[2] Q. An, S. Rahman, J. Zhou, and J. J. Kang, "A comprehensive review on machine learning in healthcare industry: classification, restrictions, opportunities and challenges," Sensors, vol. 23, no. 9, Art. no. 4178, 2023, doi: 10.3390/s23094178.

[3] R. Kumar et al., "A comprehensive review of machine learning for heart disease prediction: challenges, trends, ethical considerations, and future directions," Front. Artif. Intell., vol. 8, Art. no. 1583459, 2025, doi: 10.3389/frai.2025.1583459.

[4] M. Peng et al., "Prediction of cardiovascular disease risk based on major contributing features," Sci. Rep., vol. 13, Art. no. 4778, 2023, doi: 10.1038/s41598-023-31870-8.

[5] N. U. A. Tahir et al., "Blockchain-based healthcare records management framework: enhancing security, privacy, and interoperability," Technologies, vol. 12, no. 9, Art. no. 168, 2024, doi: 10.3390/technologies12090168.

[6] A. Haddad, M. H. Habaebi, E. A. A. Elsheikh, M. R. Islam, S. A. Zabidi, and F. E. M. Suliman, "E2EE enhanced patient-centric blockchain-based system for EHR management," PLoS One, vol. 19, no. 4, Art. no. e0301371, 2024, doi: 10.1371/journal.pone.0301371.

[7] E. H. Houssein, R. E. Mohamed, and A. A. Ali, "Heart disease risk factors detection from electronic health records using advanced NLP and deep learning techniques," Sci. Rep., vol. 13, Art. no. 7173, 2023, doi: 10.1038/s41598-023-34294-6.

[8] L. Chen, "Heart disease prediction utilizing machine learning techniques," Trans. Mater. Biotechnol. Life Sci., vol. 3, 2024.

[9] G. A. P. Febriyanti and A. Baita, "Comparison of support vector machine and decision tree algorithm performance with undersampling approach in predicting heart disease based on lifestyle," J. Appl. Inform. Comput., vol. 9, no. 2, pp. 318–327, 2025, doi: 10.30871/jaic.v9i2.8941.

[10] K. Tompra, G. Papageorgiou, and C. Tjortjis, "Strategic machine learning optimization for cardiovascular disease prediction and high-risk patient identification," Algorithms, vol. 17, no. 5, Art. no. 178, 2024, doi: 10.3390/a17050178.

[11] A. S. Osei-Nkwantabisa and R. Ntumy, "Classification and prediction of heart diseases using machine learning algorithms," arXiv, 2024, doi: 10.48550/arXiv.2409.03697.

[12] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective heart disease prediction using machine learning techniques," Algorithms, vol. 16, no. 2, Art. no. 88, 2023, doi: 10.3390/a16020088.

[13] A. Almulihi et al., "Ensemble learning based on hybrid deep learning model for heart disease early prediction," Diagnostics, vol. 12, no. 12, Art. no. 3215, 2022, doi: 10.3390/diagnostics12123215.

[14] V. Mandarino, G. Pappalardo, and E. Tramontana, "A blockchain-based electronic health record (EHR) system for edge computing enhancing security and cost efficiency," Computers, vol. 13, no. 6, Art. no. 132, 2024, doi: 10.3390/computers13060132.

[15] A. Ali et al., "A novel secure blockchain framework for accessing electronic health records using multiple certificate authority," Appl. Sci., vol. 11, no. 21, Art. no. 9999, 2021, doi: 10.3390/app11219999.

[16] D. Kumari, K. A. Kumar, A. Wagh, S. Shashank, A. Patidar, and S. Panda, "BL-MVC: blockchain enabled majority voting classifier for predicting heart diseases," in Proc. 17th Int. Conf. Agents Artif. Intell., vol. 2, ICAART, Porto, Portugal, Feb. 23–25, 2025. Setubal, Portugal: SCITEPRESS, 2025, pp. 45–56.

[17] L. Zhang, "heart_2020_cleaned," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/luyezhang/heart-2020-cleaned/data. [Accessed: Nov. 22, 2025].

[18] O. Ben-Assuli and J. R. Vest, "Data mining techniques utilizing latent class models to evaluate emergency department revisits," J. Biomed. Inform., vol. 101, Art. no. 103341, 2020, doi: 10.1016/j.jbi.2019.103341.

[19] N. Kaieski, C. A. da Costa, R. da Rosa Righi, P. S. Lora, and B. Eskofier, "Application of artificial intelligence methods in vital signs analysis of hospitalized patients: a systematic literature review," Appl. Soft Comput., vol. 96, Art. no. 106612, 2020, doi: 10.1016/j.asoc.2020.106612.

[20] M. Fratello and R. Tagliaferri, "Decision trees and random forests," in Reference Module in Life Sciences. Amsterdam, Netherlands: Elsevier, 2018, pp. 374–383, doi: 10.1016/B978-0-12-809633-8.20337-3.

[21] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati, "Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree," Reliab. Eng. Syst. Saf., vol. 200, Art. no. 106931, 2020, doi: 10.1016/j.ress.2020.106931.

[22] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., San Francisco, CA, USA, Aug. 13–17, 2016. New York, NY, USA: ACM, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[23] G. Ke et al., "LightGBM: a highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 3146–3154.

[24] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, Art. no. 1883, 2009, doi: 10.4249/scholarpedia.1883.

[25] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.