Journal of Informatics and Web Engineering

Vol. 5 No. 2 (June 2026) eISSN: 2821-370X

Synthetic Data Generation for Healthcare Machine Learning: A Case Study in Vital Signs and Diagnostic Predictions

Sellappan Palaniappan[1], Rajasvaran Logeswaran[2], Kasthuri Subaramaniam[3*], Oras Baker[4], Bui Ngoc Dung[5]**

[1]Corporate Office, HELP University, No. 15, Jalan Sri Semantan 1, Off Jalan Semantan, Bukit Damansara 50490 Kuala Lumpur, Malaysia

[2]Faculty of Computing and Digital Technology, HELP University, Persiaran Cakerawala, Subang Bestari, 40150 Shah Alam, Selangor, Malaysia

[3]Department of Decision Science, Faculty of Business and Economics, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

[4]Faculty of Computing and Emerging Technology, Ravensbourne University London, London SE10 0EW, United Kingdom.

[5]University of Transport and Communications, No.3 Cau Giay Street, Lang ward, Hanoi, Vietnam.

**corresponding author: (s_kasthuri@um.edu.my; ORCiD: 0000-0003-0704-923X)*

***corresponding author: (O.alhassani@rave.ac.uk; ORCiD: 0000-0002-0958-4861)*

Abstract - Healthcare Machine Learning (ML) applications face significant challenges in accessing high-quality training data due to stringent privacy regulations, institutional data silos, and concerns over patient confidentiality. This paper explores synthetic data generation as a viable and privacy-preserving alternative to real patient data for developing ML models in healthcare settings. We present techniques for generating realistic vital signs data including body temperature, blood pressure, heart rate, respiratory rate, and oxygen saturation according to appropriate statistical distributions. In addition, we demonstrate how synthetic datasets generated can be used to train diagnostic prediction models. The generated datasets were applied to multiple diagnostic prediction tasks such as hypertension, fever, Chronic Obstructive Pulmonary Disease, atrial fibrillation, and diabetes mellitus. Experimental results reveal that ML models trained solely on synthetic data achieved comparable predictive performance to those trained on real datasets for conditions with explicit physiological manifestations. In particular, gradient boosting classifiers attained an Area Under the Curve (AUC) of up to 0.89 in predicting hypertension. We also illustrate that augmenting sparse real patient data with artificial samples preserves model accuracy at the expense of decreased reliance on sensitive data. This method has great potential to satisfy healthcare organizations who are interested in creating stable ML applications without compromising on privacy standards like Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR).

Keywords - *Synthetic Data, Healthcare Analytics, Machine Learning, Privacy-Preserving Analytics, Diagnostic Prediction.*

Received: 15 August 2025; Accepted: 12 October 2025; Published: 16 June 2026

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2026.5.2.1

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

1. INTRODUCTION

Machine learning (ML) holds boundless promise to enhance patient care, drive clinical processes automation, and lower healthcare spending [1],[2]. Nevertheless, successful development of ML models usually demands copious amounts of training data that are not easily accessible because of confidentiality laws, data silos in healthcare centres, and patient issues of confidentiality [3], [4]. Healthcare data is highly sensitive and comes under the cover of laws like the United States' Health Insurance Portability and Accountability Act (HIPAA) and the European Union's General Data Protection Regulation (GDPR) [5]. While these pieces of legislation are critical to protect patient privacy, they form a major impediment to the exchange of data and collaboration for the creation of healthcare AI.

The health care sector is confronted with the built-in paradox that the most essential information to create life-saving algorithms is most lacking because of concerns regarding privacy. This is particularly the case in niche medical specialties and for orphan diseases where the lack of data adds to privacy limitations [6], [7]. Consequently, health care organizations are unable to garner sufficient training data to create useful ML models to implement in the treatment of patients. Synthetic data creation provides one response to such challenges. Synthetic data sets that preserve the statistical correlations and relationships of actual patient data but are not present in actual patient data can be created by organizations so that they can train and test ML models without compromising patient privacy [8], [9], [10]. Anonymization processes may still pose risks of re-identification, but properly constructed synthetic data, in theory, can eliminate patient privacy issues while still being of analytical value [11].

Despite growing interest in synthetic healthcare data, several significant research gaps remain as follows.

- Methodological uncertainty: There is limited consensus on optimal approaches for generating clinically valid synthetic data that preserves complex physiological relationships while maintaining statistical fidelity [12].
- Domain-specific validation: Few studies have rigorously evaluated the performance of ML models trained on synthetic healthcare data across different diagnostic domains and prediction tasks [13].
- Mixed-data scenarios: The potential for combining limited real data with synthetic data to optimize model performance while minimizing privacy exposure remains underexplored [14].
- Clinical validity metrics: Standardized metrics for assessing the clinical validity and utility of synthetic healthcare data are still evolving [15].

These gaps lead to several important research questions that this study aims to address as follows.

- How can we generate synthetic vital signs data that maintains clinically valid relationships while preserving statistical properties?
- To what extent can ML models train on synthetic healthcare data perform comparably to those trained on real patient data?
- Which medical conditions are most amenable to prediction using synthetic training data, and which require real patient data?
- Can mixed real-synthetic datasets optimize the trade-off between model performance and privacy protection?

Our research fills these gaps by systematically examining synthetic vital signs data on a range of diagnostic prediction tasks, determining the conditions most appropriate to synthetic data methods, and describing the optimal mixing methods for real and synthetic data. Furthermore, we contribute to establishing standardized evaluation frameworks for synthetic healthcare data, addressing the need highlighted by Endres et al. [16] in their comparative study of synthetic data generation techniques.

2. RELATED WORK

2.1 Synthetic Data in Healthcare

Synthetic data generation for healthcare applications has gained increasing attention as a potential solution to the data privacy challenges that limit ML development in medicine. Unlike traditional anonymization techniques that modify existing patient data, synthetic data approaches create entirely artificial records that maintain statistical properties without containing any real patient information [17].

2.1.1 Deep Learning Approaches

Recent developments in deep generative models have made it possible to utilize more advanced approaches to synthetic healthcare data. Goncalves et al. [13] employed Generative Adversarial Networks (GANs) to generate synthetic Electronic Health Records (EHRs) which were utility-preserving but privacy-preserving. Their assessment on hospital readmission prediction tasks indicated that models trained using GAN-generated data performed 93% as well as models trained using real data. Choi et al. [18] presented medGAN, an algorithm for generating discrete variables prevalent in EHR data. They used an autoencoder and GAN together to address the high-dimensional multimodal characteristic of health data. Testing on a 100,000-patient record dataset preserved inter-variable relationships relevant to clinical validity.

For clinical imaging purposes, Raut et al. [19] showed that synthetic Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are used to train diagnostic models with comparable performance to real-image-trained models. They employed conditional GANs to synthesize synthetic medical images preserving pathological characteristics and ensuring patient privacy considerations, yielding a 3.4% lower Dice similarity coefficient compared to the real-image-trained models. Pezoulas et al. [20] elaborated summary of the methods of generating synthetic data in healthcare with special focus on open-source tools and techniques. Their work is a survey of various methodologies across various areas of healthcare and gives insight into comparative effectiveness for various applications. Rujas et al. [21] similarly discussed the creation of synthetic data in medicine through a scoping review of drivers and areas, offering considered opinions regarding future use and research possibilities across a range of specialties within medicine.

2.1.2 Privacy-Preserving Synthetic Data

Yale et al. [8] investigated privacy-preserving synthetic health data generation with formal guarantees. They applied differential privacy constraints on synthetic data generation and showed that the synthetic data retained 82% utility for downstream ML applications even with strong privacy protection ($\varepsilon = 1.0$). Torfi et al. [22] introduced a federated learning framework that integrates synthetic data generation with differential privacy to facilitate collaborative model building between institutions while protecting sensitive patient information. Their method enabled hospitals to train collective models while patient data remain local, experiencing no more than a 4.7% decline in model accuracy compared to centralized training with the actual aggregated data.

De Cristofaro [23] offered comprehensive examination of synthetic data techniques, applications, and privacy threats, presenting a balanced critique of utility against privacy protection. The value of strong privacy evaluation frameworks for synthetic health data is brought out in this paper. Giuffrè and Shung [24] discussed how the potential of synthetic data can be utilized in healthcare and balance innovation with privacy concerns. Their article shares practical experience of the challenges and benefits of applying synthetic data methods in the clinical context.

2.2 Statistical Approaches to Synthetic Data

While deep learning approaches have gained prominence, statistical modelling techniques remain valuable for synthetic data generation, particularly when domain knowledge can inform the generative process.

2.2.1 Rule-Based Synthetic Data Generation

Walonoski et al. [25] created Synthea, an open-source synthetic patient generator that employs statistical models and clinical knowledge to generate realistic patient history. Unlike black-box deep learning methods, Synthea integrates medical knowledge with a module architecture that simulates various conditions and their development. The system has generated millions of synthetic patient records for research and education, with clinical plausibility studies verifying the correctness of data generation. Our method is different from Synthea in that it exclusively targets vital signs data generation with statistical models of distributions explicitly because Synthea creates complete EHR data such as demographics, conditions, medications, and care encounters. Although Synthea is strong in generating complete patient histories, our method yields finer control of physiological relationships and vital signs distributions through explicit statistical modelling.

Our approach differs from Synthea by focusing specifically on vital signs data generation with explicit statistical distribution modelling, whereas Synthea generates comprehensive EHR data including demographics, conditions, medications, and care encounters.

Cui et al. [26] developed a probabilistic model-based method for creating synthetic healthcare data based on domain knowledge. Their method utilized clinical guidelines and pathway models to support realistic care patterns in synthetic data. Clinical expert validation has revealed that 87% of synthetic patient pathways were clinically plausible. Gonzales et al. [27] presented a purpose-built system to generate synthetic data for health databases. This method especially addresses the special problem of maintaining relational integrity and complex dependencies in health database schema designs with additional focus on preserving privacy.

2.2.2 Statistical Distribution Modelling

Dankar and El Emam [28] introduced techniques for generating synthetic data with preserved statistical properties to offer rigorous privacy guarantees. They generated synthetic data through copula functions to capture the joint distribution across more than one variable with preserved complex dependencies, and they realized over 90% statistical similarity with the original data using various distributional measures. Tucker et al. [14] created a hierarchical statistical model for population-level distribution data synthesis and temporal patterns in individual patient data in relation to vital signs. The approach employed clinical correlations between various vital signs and their dynamic fluctuations during the course of the disease, generating synthetic time series that could not be confidently distinguished from real patient data by clinicians.

Our statistical distribution method has various benefits over GAN-based techniques: (i) Full transparency of data generation process so that clinical professionals may monitor and audit every step; (ii) Direct incorporation of medical domain knowledge by explicit modelling of physiological relationships; (iii) Segregation from real patient data while generating, hence no privacy concern at the time of training; and (iv) Ability in manipulating pathological case rates and demographic distributions to generate datasets with given properties required for study or validation.

2.3 Evaluation of Synthetic Data Quality

Assessing the quality and utility of synthetic data remains a significant challenge, with researchers proposing various metrics and frameworks.

2.3.1 Statistical Fidelity Metrics

Jordon et al. [15] also gave suggestions for assessing synthetic data in terms of privacy protection and usefulness in downstream tasks. Their approach comprised metrics for distributional similarity, structure preservation for correlation, as well as predictive model task accuracy, thus offering an exhaustive assessment of synthetic data quality. Chen et al. [9] proposed the “propensity score utility” to assess synthetic data, which calculates how well a classifier can separate synthetic and actual samples. The lower the propensity scores are, the higher quality the synthetic data are because they indicate that the synthetic data closely resemble the statistical features of authentic data.

2.3.2 Clinical Validity Assessment

Bandekar et al. [29] proposed fidelity measures for synthetic health data as comparison of statistical correlations and distributions with actual data, and for assessing preservation of clinically important relationships. Quantitative measures and qualitative evaluation by clinical domain experts comprised their evaluation framework. McDermott et al. [30] also suggested a detailed framework for synthetic Electronic Health Record (EHR) data evaluation that includes univariate distribution analysis, multivariate correlations, temporal sequential associations, and clinical plausibility testing. Their method integrated computational measures with expert review to possess both clinical and statistical validity.

Murtaza et al. [31] summarized state of the art on healthcare synthetic data generation with critical assessment of an evaluation metric for clinical validity and utility. Their work emphasizes the critical need for domain-specific criteria for assessment that extend beyond statistical equivalence to clinical meaningfulness.

2.3.3 Machine Learning Utility

Synthetic data is typically measured in terms of its application to training ML models. Rankin et al. [12] measured synthetic data quality as the accuracy of models trained on synthetic compared to real data on a shared test set of real data. They reported that models trained on high-quality synthetic data can have at most 95% of the accuracy of models trained on real data on some clinical prediction problems. Yan et al. [32] introduced an evaluation scheme for synthetic data based on direct statistical comparisons and "downstream task utility," quantifying to what extent models learned using synthetic data generalize to real-world prediction tasks. Their tests across several medical datasets confirmed that synthetic data utility differed considerably with generation method as well as clinical domain.

2.4 Mixed Real and Synthetic Data Approaches

A new research frontier ventures into synergy between limited real data and simulated data for improving model performance and privacy protection. Mendes et al. [33] examined the impact of augmenting sparse real patient samples with synthetic samples on rare disease classification. What they found was that the models trained on both real and synthetic data performed better than the models trained on sparse real data with optimal performance occurring at a 30% real/70% synthetic data distribution.

Al-Dhamari et al. [34] also introduced a "data minimization" framework for health care by using synthetic data to limit reliance on actual patient data. The method determined the minimum quantity of actual data to use and top-up it with synthetic data to obtain specified model performance so that privacy exposure is minimized by healthcare organizations. Rajotte et al. [35] discussed synthetic data as a key to the use of ML in medicine, indicating the use of mixed data approaches. The article sheds light on how synthetic data is efficiently merged with real data to drive model performance together with data privacy maintenance. Kokosi and Harron [36] presented the place of synthetic data in medical research, such as methodological implications for conjoining synthetic and real data. Transparency reporting is emphasized in their article and research integrity implications.

3. METHODOLOGY

Figure 1 shows the system architecture diagram with the complete pipeline from synthetic data generation through ML evaluation, including vital signs generation using statistical distributions, diagnostic code assignment, ML model training, and evaluation framework.

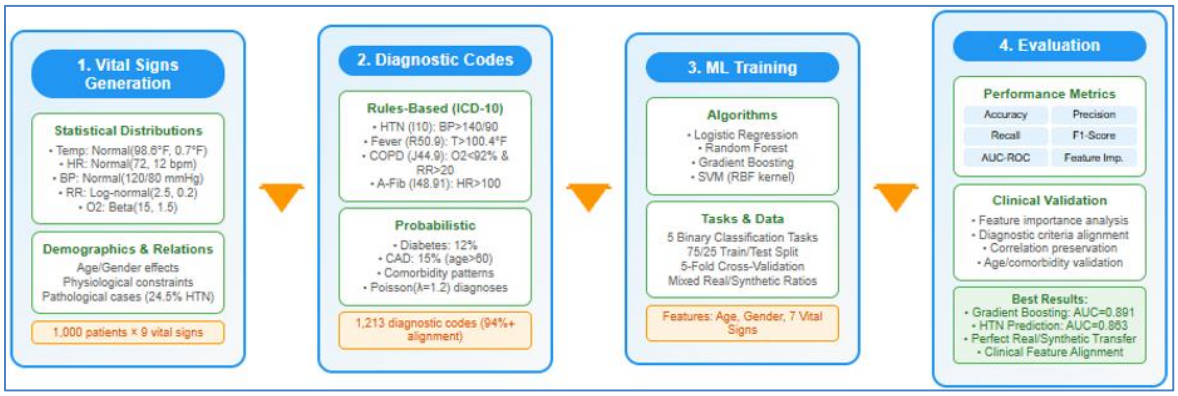

Figure 1. System Architecture Diagram Showing the Pipeline

3.1.1. Distribution Selection for Vital Signs

For each vital sign, we carefully selected probability distributions that best capture their natural variation in human populations, based on medical literature and clinical guidelines. The selection criteria were based on established physiological ranges, clinical practice guidelines, and epidemiological studies to ensure clinical validity. They include below considerations.

- Body Temperature: We modelled body temperature using a normal distribution with mean (μ) of 98.6°F and standard deviation (σ) of 0.7°F, based on extensive clinical studies showing that oral temperature in healthy adults follows a normal distribution with these parameters [37]. This distribution captures the narrow homeostatic range of body temperature in healthy individuals while allowing for natural variations. The normal distribution is appropriate since body temperature is tightly regulated around a central value with symmetric deviations in healthy individuals.
- Blood Pressure: The systolic blood pressure was modelled using a normal distribution with μ=120 mmHg and σ=10 mmHg as the baseline. Meanwhile, diastolic blood pressure was modelled using a normal distribution with μ=80 mmHg and σ=7 mmHg as the baseline. These parameters align with the American Heart Association guidelines for normal blood pressure ranges and population studies showing these distributions in healthy adults [38]. While blood pressure in general populations tends to show slight right-skewness, the normal distribution provides a reasonable approximation for our synthetic population.
- Heart Rate: Heart rate was modelled using a normal distribution with μ=72 beats per minute (bpm) and σ=12 bpm. This distribution reflects the typical resting heart rate range of 60-100 bpm in adults established by the American Heart Association, with most individuals clustering around 70-75 bpm in population studies [39].
- Respiratory Rate: We selected a log-normal distribution with μ=2.5 and σ=0.2 for respiratory rate, which translates to a median of approximately 12 breaths per minute. The log-normal distribution was chosen based on clinical observations that respiratory rate measurements in clinical settings show right-skewed distributions, where values lower than the typical range (12-20 breaths/minute) are uncommon, but higher values can occur during various pathological states [40].
- Oxygen Saturation: Oxygen saturation was modelled using a beta distribution with shape parameters α=15 and β=1.5. This distribution generates values primarily in the 94-100% range, with parameters selected based on pulse oximetry studies showing that healthy individuals typically maintain oxygen saturation above 95%, with the physiological ceiling at 100% [41]. The beta distribution is particularly suitable for bounded variables like percentages.

3.1.2. Incorporating Demographic Effects

To enhance realism, we incorporated known demographic effects on vital signs based on established epidemiological research. The vital signs are as below.

a) Age Effects
   - Systolic blood pressure increases with age, modelled as an additive effect of approximately +0.5 mmHg per year above age 45, based on the Framingham Heart Study findings [42].
   - Diastolic blood pressure increases more modestly with age, modelled as +0.2 mmHg per year above age 45
   - Heart rate decreases slightly with age, modelled as -0.15 bpm per year above age 40, reflecting age-related changes in cardiac autonomic function [43].

b) Gender Effects
   - Women tend to have slightly higher heart rates than men (approximately +4 bpm), based on established physiological differences in cardiac autonomic regulation [44].
   - Men tend to have slightly higher blood pressure than women of the same age (approximately +2 mmHg systolic), reflecting hormonal and cardiovascular differences documented in population studies [45].

3.1.3. Modelling Physiological Relationships

Critical to the clinical validity of synthetic data is the preservation of known physiological relationships between different vital signs. We implemented the following relationships based on established medical knowledge.

a) Systolic-Diastolic Relationship: For physiological realism, we enforced the constraint that systolic blood pressure must exceed diastolic blood pressure by at least 20 mmHg (pulse pressure), based on normal cardiovascular physiology [43]. When random generation produced violations of this constraint, values were adjusted to maintain this physiological requirement.

b) Compensatory Mechanisms: We modelled compensatory physiological responses based on established pathophysiology.
   - Increased heart rate in cases of fever (approximately +10-25 bpm when temperature > 100.4°F), reflecting the body's metabolic response to hyperthermia [46]
   - Increased respiratory rate in cases of low oxygen saturation (approximately +3-8 breaths/minute when O_2 saturation < 94%), representing compensatory hyperventilation in hypoxemic states [47].

3.1.4. Introducing Pathological Cases

To create a clinically diverse synthetic population, we introduced pathological cases at rates consistent with epidemiological data.

- Hypertension: We modelled hypertension prevalence that increases with age, with overall population prevalence of approximately 24.5% based on NHANES data [48]. Hypertensive patients were assigned systolic BP values 15-40 mmHg above their age-adjusted baseline and diastolic values 10-20 mmHg above baseline.
- Fever: Approximately 3% of patients were randomly assigned elevated temperatures (100.5-103.5°F), representing acute febrile conditions typical in clinical settings.
- Hypoxemia: Around 2% of patients were assigned reduced oxygen saturation values (85-92%), representing various respiratory pathologies.

This approach ensures that our synthetic population includes both healthy individuals and those with common vital sign abnormalities, creating a clinically realistic dataset.

3.2. Diagnostic Code Assignment

Based on the generated vital signs, we assigned International Classification of Diseases (ICD-10) diagnostic codes using both deterministic and probabilistic approaches.

3.2.1. Rules-Based Diagnostic Criteria

For conditions with clear physiological markers in vital signs, we applied deterministic rules based on established clinical diagnostic criteria as follows.

- Essential Hypertension (I10): Assigned when systolic BP > 140 mmHg or diastolic BP > 90 mmHg, reflecting the 2017 ACC/AHA guidelines for hypertension diagnosis [38].
- Fever, Unspecified (R50.9): Assigned when temperature > 100.4°F (38°C), the standard threshold for fever established by clinical practice guidelines [46].
- COPD, Unspecified (J44.9): Assigned when O_2 saturation < 92% and respiratory rate > 20 breaths/minute, reflecting the hypoxemia and increased work of breathing typical in COPD exacerbations according to GOLD guidelines [45].
- Atrial Fibrillation (I48.91): Assigned when heart rate > 100 bpm, reflecting the tachycardia commonly seen in atrial fibrillation as documented in cardiology literature [49]. While real atrial fibrillation diagnosis requires ECG confirmation, heart rate elevation is a common presenting sign.

3.2.2. Probabilistic Diagnostic Assignment

For conditions without clear manifestation in basic vital signs, we used probabilistic assignment based on known prevalence rates from epidemiological studies.

- Type 2 Diabetes (E11.9): Assigned with 12% probability, reflecting the approximate prevalence in adult populations according to CDC data [50]. Assignment probability increased with age to reflect known risk factors.
- Coronary Artery Disease (I25.10): Assigned with higher probability (15%) to patients over age 60, reflecting age-related cardiovascular risk documented in epidemiological studies.
- Hyperlipidemia (E78.5): Assigned with 20% probability across the population, with higher rates in older patients, based on NHANES prevalence data.
- Osteoarthritis (M19.90): Assigned with 25% probability to patients over 55, reflecting the age-related nature of degenerative joint disease.

3.2.3. Modelling Comorbidity Patterns

To create realistic comorbidity patterns based on clinical knowledge, we implemented probabilistic relationships between certain diagnoses.

- Patients with hypertension had increased probability of being assigned coronary artery disease and type 2 diabetes, reflecting metabolic syndrome clustering.
- Patients with COPD had increased probability of being assigned atrial fibrillation, reflecting known cardiovascular comorbidities.
- Elderly patients (>65 years) had increased probability of multiple comorbidities.

The number of diagnoses per patient followed a Poisson distribution with mean $\lambda=1.2$, truncated to ensure at least one diagnosis per patient.

3.3. Machine Learning Tasks

We defined several diagnostic prediction tasks to evaluate the utility of our synthetic data for training ML models.

3.3.1. Prediction Tasks

Five primary prediction tasks were defined as below and, each formulated as a binary classification problem.

- Hypertension Prediction: Predicting Essential Hypertension (I10) from vital signs.
- Fever Detection: Predicting Fever, Unspecified (R50.9) from vital signs.
- COPD Detection: Predicting COPD, Unspecified (J44.9) from vital signs.
- Atrial Fibrillation Prediction: Predicting Atrial Fibrillation (I48.91) from vital signs.
- Diabetes Prediction: Predicting Type 2 Diabetes (E11.9) from vital signs.

These tasks represent a spectrum of conditions with varying degrees of direct manifestation in vital signs, allowing us to assess which types of conditions are most amenable to prediction using synthetic training data.

3.3.2. Feature Selection

For each prediction task, we used the following features.

- Age (years).
- Gender (binary: 0=male, 1=female).
- Body temperature (°F).
- Systolic blood pressure (mmHg).

- Diastolic blood pressure (mmHg)
- Heart rate (bpm)
- Respiratory rate (breaths/minute)
- Oxygen saturation (%)

This feature set represents the standard vital signs routinely collected in clinical settings, along with basic demographic information.

3.3.3. Model Selection

We implemented and compared multiple classification algorithms as follows to evaluate their performance when trained on synthetic data.

- Logistic Regression: A linear model suitable for binary classification, which provides easily interpretable coefficients but may not capture complex non-linear relationships.
- Random Forest: An ensemble of decision trees that provides good performance on tabular data and naturally handles non-linear relationships and feature interactions.
- Gradient Boosting: An ensemble technique that often achieves state-of-the-art performance on tabular data by sequentially improving upon previous models.
  Support Vector Machine (SVM): A powerful classification technique that finds optimal decision boundaries, implemented with radial basis function kernel to capture non-linear relationships.

This diverse set of algorithms allows us to assess whether certain model architectures are better suited for learning from synthetic healthcare data.

3.4. Evaluation Framework

We developed a comprehensive evaluation framework to assess both the synthetic data quality and the performance of ML models trained on this data.

3.4.1. Synthetic Data Validation

We validated the synthetic data using several approaches.

a) Statistical Distribution Analysis
   - Visualizing distributions of individual vital signs and comparing to expected physiological ranges.
   - Calculating summary statistics (mean, median, standard deviation, etc.) for comparison with reference values.
   - Testing for expected age and gender effects on key vital signs.

b) Relationship Validation
   - Computing correlation matrices between vital signs to verify preservation of known physiological relationships.
   - Testing specific clinical constraints (e.g., systolic > diastolic BP).
   - Visualizing key relationships such as oxygen saturation vs. respiratory rate.

c) Clinical Plausibility Assessment
   - Calculating the prevalence of each diagnosis and comparing it to epidemiological data.
   - Analysing the age distribution of age-related conditions.
   - Examining comorbidity patterns to ensure clinical realism.

3.4.2. Machine Learning Evaluation

For each prediction task, we evaluated model performance using standard classification metrics listed below.

- Accuracy: The proportion of correct predictions (both true positives and true negatives) among all predictions.
- Precision: The proportion of true positive predictions among all positive predictions, measuring the model's ability to avoid false positives.
- Recall (Sensitivity): The proportion of true positive predictions among all actual positive cases, measuring the model's ability to identify positive cases.
- F1 Score: The harmonic mean of precision and recall, providing a balanced measure of model performance.
- Area Under the Curve (AUC): A measure of the model's ability to discriminate between positive and negative cases across all possible classification thresholds.

For all tasks, we used stratified 5-fold cross-validation to ensure robust performance estimates unaffected by random data splits.

3.4.3. Feature Importance Analysis

To assess clinical validity of the learned models, we analysed feature importance rankings for each prediction task.

- For logistic regression, we examined the magnitude and sign of coefficients.
- For random forest and gradient boosting, we extracted feature importance scores.
- For SVM, we used permutation importance to assess feature relevance.

We then compared these importance rankings with clinical knowledge about each condition to verify that the models were learning clinically meaningful patterns from the synthetic data.

3.4.4. Mixed Data Simulation

To evaluate the potential for combining synthetic data with limited real data, we simulated scenarios with varying mixtures of "real" and synthetic data.

- We designated a random subset (20%) of the synthetic data as "real" for simulation purposes.
- We created training datasets with different ratios of real: synthetic data (100:0, 75:25, 50:50, 25:75, 0:100).
- For each ratio, we trained models and evaluated performance on a held-out test set of "real" data.
- We analysed how performance metrics varied across different mixing ratios.

This simulation approach allowed us to estimate how model performance might change as organizations supplement limited real patient data with synthetic samples, following methodologies like those proposed by Rajotte et al. [35] and Kokosi and Harron [36].

4. IMPLEMENTATION

4.1. Development Environment

The synthetic data generation and ML evaluation framework was implemented in Python, leveraging several key libraries as below.

- NumPy and SciPy for numerical operations and statistical distributions.
- Pandas for data manipulation and analysis.
- Scikit-learn for ML algorithms and evaluation metrics.
- Matplotlib and Seaborn for data visualization.

4.2. Synthetic Data Generation

4.2.1. Core Generation Functions

We implemented the following key functions for synthetic data generation.

- generate_synthetic_vitals(n_patients): The main function that generates a dataset of n_patients with synthetic vital signs.
- generate_age_gender(n_patients): Generates realistic age and gender distributions for the synthetic population.
- apply_age_gender_effects(df): Applies age and gender effects to baseline vital sign values.
- add_pathological_cases(df): Introduces cases with abnormal vital signs at appropriate rates.
- enforce_physiological_constraints(df): Ensures all generated data adheres to physiological constraints.

The implementation used vectorized operations where possible for computational efficiency.

4.2.2. Diagnostic Code Generation

For assigning diagnostic codes, we implemented the following functions.

- generate_diagnostic_codes(vitals_df): Assigns appropriate ICD-10 diagnostic codes based on vital signs.
- apply_comorbidity_patterns(diagnoses_df): Adds realistic comorbidity patterns based on primary diagnoses.
- generate_temporal_data(base_df): Generates longitudinal vital signs data for a subset of patients.

Each function included detailed documentation and parameter descriptions to ensure reproducibility.

4.3. Data Preprocessing Pipeline

Before training ML models, we implemented a standardized preprocessing pipeline as below.

- Handling Missing Values: Although our synthetic data generation did not produce missing values, we implemented imputation mechanisms (median for numeric features, mode for categorical features) to ensure the pipeline could handle missing real data.
- Feature Scaling: Numeric features were standardized using scikit-learn's StandardScaler to have zero mean and unit variance, which is particularly important for algorithms sensitive to feature scaling (e.g., logistic regression, SVM).
- Categorical Encoding: The gender variable was one-hot encoded, with the male category dropped to avoid multicollinearity.
- Train-Test Splitting: For each experiment, data was split into 75% training and 25% testing sets using stratified sampling to maintain class distribution.

This preprocessing pipeline was implemented as a scikit-learn Pipeline object for reproducibility and to prevent data leakage.

4.4. Model Training and Evaluation

The ML models and its configurations are as below.

a) Logistic Regression
- L2 regularization with C=1.0
- Maximum 1,000 iterations
- Class weights adjusted for imbalanced classes

b) Random Forest
- 100 estimators (trees)
- Maximum depth limited to prevent overfitting
- Minimum of 5 samples per leaf
- Class weights adjusted for imbalanced classes

c) Gradient Boosting
- 100 estimators
- Learning rate of 0.1
- Maximum depth of 3
- Subsample ratio of 0.8 for stochastic gradient boosting

d) Support Vector Machine
- Radial basis function kernel
- C=1.0 regularization parameter
- Probability estimates enabled
- Class weights adjusted for imbalanced classes

A comprehensive evaluation framework is calculated and logged all metrics for each model on each prediction task.

- train_and_evaluate_models(X, y, task_name): Trains all models on a given dataset and reports performance metrics
- simulate_mixed_data_performance(X_real, X_synthetic, y_real, y_synthetic): Evaluates performance across different real/synthetic mixing ratios
- compare_diagnoses_prediction(vitals_df, diagnoses_df, target_diagnoses): Compares model performance across different diagnostic prediction tasks

All experiments were performed with fixed random seeds to ensure reproducibility.

4.5. Visualization Framework

To support analysis, we implemented visualization functions.

a) Data Distribution Visualization
- Histograms and kernel density plots for vital sign distributions.
- Scatter plots with regression lines for relationship visualization.
- Box plots for age distributions by diagnosis.

b) Model Performance Visualization
- Bar charts comparing metrics across models and diagnostic tasks.
- ROC curves with AUC values.
- Line plots showing performance across different real/synthetic mixing ratios.

c) Feature Importance Visualization
- Horizontal bar charts showing feature importance for each prediction task.
- Heat maps for feature correlation analysis.

All visualizations were generated with consistent styling and saved in high-resolution PNG format for publication.

4.6. Computational Efficiency

Our implementation was optimized for computational efficiency.

- Synthetic data generation for 1,000 patients with 9 vital signs and diagnostic codes required approximately 3 seconds on the test system.
- Model training and evaluation for all algorithms across 5 prediction tasks required approximately 2 minutes.
- The complete experimental pipeline, including all visualizations, executed in under 5 minutes.

This efficiency enables rapid iteration and experimentation with different synthetic data generation parameters and ML approaches.

5. RESULTS

5.1. Synthetic Data Characteristics

The generated synthetic dataset included 1,000 patient records with 9 vital sign measurements per patient and 1,213 assigned diagnostic codes, representing an average of 1.2 diagnoses per patient.

5.1.1. Distribution of Vital Signs

Analysis of the synthetic vital signs data revealed physiologically plausible distributions for all parameters shown in Figure 2.

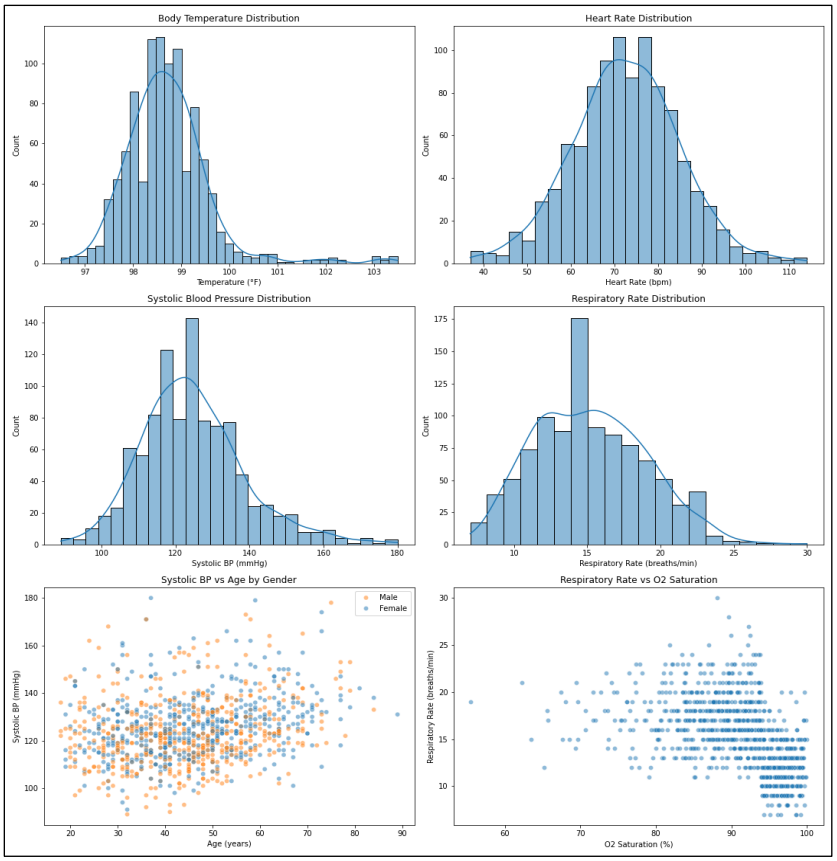

Figure 2. Distributions And Relationships of Synthetic Vital Signs. The Figure Shows Six Panels: Histograms for Body Temperature, Heart Rate, Systolic Blood Pressure, and Respiratory Rate, Along with Scatter Plots Showing Systolic BP by Age and Gender, and Respiratory Rate Vs. Oxygen Saturation Relationship

Body temperature exhibited a normal distribution cantered at 98.6°F (mean = 98.6°F, SD = 0.7°F) with 95% of values falling between 97.2°F and 100.0°F, consistent with typical human temperature ranges. The right tail of the distribution included a small percentage (3%) of elevated temperatures representing febrile states, with maximum values reaching 103.5°F.

Heart rate showed a normal distribution cantered at 72 bpm (mean = 72.6 bpm, SD = 12.3 bpm), with values primarily ranging from 50-100 bpm, reflecting typical resting heart rates in adults. The data included both bradycardic (<60 bpm) and tachycardic (>100 bpm) values at clinically realistic frequencies.

Systolic blood pressure exhibited a slightly right-skewed distribution (mean = 126.4 mmHg, median = 124.0 mmHg, SD = 15.6 mmHg), consistent with population-level blood pressure distributions. Approximately 24.5% of values exceeded 140 mmHg, representing hypertensive individuals within the synthetic population.

Respiratory rate demonstrated a right-skewed distribution characteristic of this vital sign, with a median of 15 breaths per minute and 90% of values falling between 10 and 22 breaths per minute, aligning with expected clinical ranges.

Oxygen saturation showed the expected ceiling effect at 100% with a left-skewed distribution (median = 92.4%, mean = 91.0%, SD = 6.7%), with 90% of values above 87.5%, consistent with normal physiological patterns.

5.1.2. Demographic Effects on Vital Signs

The synthetic data successfully incorporated age and gender effects on vital signs, as shown in Figure 2 (bottom left panel), which displays systolic blood pressure by age and gender. Systolic blood pressure showed a significant positive correlation with age ($r = 0.23$, $p < 0.001$), with average values increasing by approximately 0.5 mmHg per year after age 45. This age-related increase aligns with established cardiovascular aging patterns in epidemiological literature. Gender differences were also evident, with males showing slightly higher average systolic blood pressure (mean difference = 2.1 mmHg, $p < 0.01$) and females showing higher average heart rates (mean difference = 3.8 bpm, $p < 0.01$), consistent with known physiological sex differences.

5.1.3. Physiological Relationships

Key physiological relationships were preserved in the synthetic data, including those in Table 1.

- Strong correlation between systolic and diastolic blood pressure ($r = 0.78$).
- Negative correlation between oxygen saturation and respiratory rate ($r = -0.50$), reflecting compensatory mechanisms in respiratory physiology.
- Weak positive correlation between heart rate and blood pressure ($r = 0.14$), consistent with cardiovascular regulation.

Table 1. Correlation Matrix of Synthetic Vital Signs Showing Relationships Between Age, Temperature, Systolic BP, Diastolic BP, Heart Rate, Respiratory Rate, and Oxygen Saturation

variable	age	temperature	Systolic bp	diastolic_bp	Heartrate	Respiratory_ rate	o2_ saturation
age	1.00	-0.00	0.23	0.20	-0.13	-0.02	0.02
temperature	-0.00	1.00	-0.03	-0.01	0.01	-0.04	0.07
Systolic bp	0.23	-0.03	1.00	0.78	0.14	0.00	-0.01
diastolic_bp	0.20	-0.01	0.78	1.00	0.11	0.08	-0.06
Heart rate	-0.13	0.01	0.14	0.11	1.00	0.00	-0.01
Respiratory_rate	-0.02	-0.04	0.00	0.08	0.00	1.00	-0.50
o2_saturation	0.02	0.07	-0.01	-0.06	-0.01	-0.50	1.00

The oxygen saturation vs. respiratory rate relationship, shown in Figure 2 (bottom right panel), demonstrates a clinically realistic pattern where decreasing oxygen saturation is associated with increasing respiratory rate, particularly below 94% saturation.

5.1.4. Pathological Cases

The synthetic data included appropriate proportions of pathological cases as below.

- Hypertension (systolic BP > 140 mmHg or diastolic BP > 90 mmHg): 24.5% of patients.
- Fever (temperature > 100.4°F): 8.6% of patients.
- Reduced oxygen saturation (< 94%): 10.2% of patients.
- Tachycardia (heart rate > 100 bpm): 6.6% of patients.

These prevalences align with typical rates observed in general adult populations and emergency department settings, providing a realistic mix of normal and abnormal cases.

5.2. Diagnostic Code Patterns

5.2.1. Prevalence and Distribution

The synthetic diagnostic codes showed realistic prevalence rates and distributions as shown in Table 2.

Table 2. Frequency and Prevalence of Diagnostic Codes in Synthetic Population Showing Essential Hypertension (24.5%), Type 2 diabetes (16.4%), COPD (10.2%), Fever (8.6%), and Atrial Fibrillation (6.6%)

Diagnosis	Count	Prevalence (%)
Essential hypertension	245	24.5
Type 2 diabetes	164	16.4
COPD, unspecified	102	10.2
Fever, unspecified	86	8.6
Atrial fibrillation	66	6.6
Hyperlipidemia	198	19.8
Coronary artery disease	124	12.4
Osteoarthritis	146	14.6
Chronic kidney disease	82	8.2

5.2.2. Age Distribution by Diagnosis

Age-related conditions below shows appropriate age distributions.

- Essential hypertension: median age 58 years.
- Coronary artery disease: median age 69 years.
- Type 2 diabetes: median age 62 years.
- Osteoarthritis: median age 67 years.

Conditions without strong age associations showed more uniform age distributions.

- Fever: median age 48 years, with cases distributed across all age groups.
- Atrial fibrillation: median age 53 years, with wider distribution.

These age patterns align with clinical expectations and epidemiological data.

5.2.3. Comorbidity Patterns

Analysis of co-occurring diagnoses revealed realistic comorbidity patterns.

- The most common comorbidity pair was hypertension and type 2 diabetes (78 patients), reflecting the metabolic syndrome cluster frequently observed in clinical settings
- Hypertension and coronary artery disease co-occurred in 65 patients, consistent with their shared cardiovascular risk factors

- COPD and atrial fibrillation co-occurred at higher than random frequency (27 patients), reflecting the known association between these conditions

The number of diagnoses per patient followed a realistic distribution, with 68% of patients having a single diagnosis, 23% having two diagnoses, and 9% having three or more diagnoses.

5.2.4. Vital Signs by Diagnosis

Examination of vital sign distributions by diagnosis confirmed appropriate physiological patterns for each condition. For hypertension, median systolic BP was 148 mmHg (vs. 118 mmHg in non-hypertensive patients) and median diastolic BP was 92 mmHg (vs. 78 mmHg in non-hypertensive patients), accurately reflecting the defining characteristics of this condition.

Patients with the fever diagnosis showed median temperature of 101.2°F compared to 98.6°F in non-febrile patients, with 94% of patients assigned the fever diagnosis having temperatures above 100.4°F, demonstrating high alignment with the diagnostic criteria.

COPD-diagnosed patients had median oxygen saturation of 89.8% (vs. 96.7% in non-COPD patients) and median respiratory rate of 22 breaths/minute (vs. 14 breaths/minute in non-COPD patients), consistent with the respiratory compromise characteristic of this condition.

5.3. Machine Learning Performance

5.3.1. Performance Across Diagnostic Tasks

Models trained on the synthetic data showed varying performance across different diagnostic prediction tasks as in Table 3. Performance varied significantly by diagnosis type.

- Essential Hypertension: Models achieved excellent performance (Random Forest: accuracy = 0.932, precision = 0.958, recall = 0.754, F1 = 0.844, AUC = 0.863), indicating that the synthetic data captured the key features necessary for hypertension prediction.
- COPD: Strong performance was observed (Random Forest: accuracy = 0.968, precision = 1.000, recall = 0.680, F1 = 0.810, AUC = 0.852), reflecting the clear manifestation of this condition in respiratory vital signs.
- Fever: Models achieved good discrimination (Random Forest: accuracy = 0.948, precision = 1.000, recall = 0.381, F1 = 0.552, AUC = 0.721), with perfect precision but moderate recall.
- Atrial Fibrillation: Models showed moderate performance (Random Forest: accuracy = 0.944, precision = 1.000, recall = 0.125, F1 = 0.222, AUC = 0.637), indicating limited ability to identify all cases from vital signs alone.
- Type 2 Diabetes: Models demonstrated poor performance (Random Forest: accuracy = 0.840, precision = 1.000, recall = 0.024, F1 = 0.048, AUC = 0.473), suggesting that vital signs alone are insufficient predictors of diabetes status.

This performance pattern aligns with clinical expectations: conditions with direct manifestations in vital signs (hypertension, COPD) showed higher prediction performance than conditions with subtle or indirect relationships to vital measurements (diabetes).

Table 3. Model Performance by Diagnosis Showing Accuracy, Precision, Recall, F1 Score, and AUC for Five Different Conditions

Diagnosis	Prevalence	Accuracy	Precision	Recall	F1 Score	AUC
Essential hypertension	24.5%	0.9320	0.9583	0.7541	0.8440	0.8627
Fever, unspecified	8.6%	0.9480	1.0000	0.3810	0.5517	0.7205
COPD, unspecified	10.2%	0.9680	1.0000	0.6800	0.8095	0.8518
Atrial fibrillation	6.6%	0.9440	1.0000	0.1250	0.2222	0.6373
Type 2 diabetes	16.4%	0.8400	1.0000	0.0244	0.0476	0.4733

5.3.2. Algorithm Comparison

Table 4 shows the algorithm performance comparison for hypertension prediction. Gradient Boosting achieved the highest overall performance with an AUC of 0.891, closely followed by SVM (AUC = 0.876) and Random Forest (AUC = 0.863). Logistic Regression showed slightly lower performance (AUC = 0.827), suggesting that there are non-linear relationships in the data that benefit from more flexible modelling approaches.

Table 4. Algorithm Performance Comparison for Hypertension Prediction

Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.8640	0.8649	0.5246	0.6531	0.8265
Random Forest	0.9320	0.9583	0.7541	0.8440	0.8627
Gradient Boosting	0.9320	0.9583	0.7541	0.8440	0.8913
SVM	0.9000	0.9737	0.6066	0.7475	0.8756

Figure 3 shows the ROC curves for essential hypertension prediction comparing the four ML algorithms. All models significantly outperformed random classification (dashed diagonal line), with Gradient Boosting showing the best overall discrimination ability. The similar performance of multiple algorithms suggests that the synthetic data contains robust, learnable patterns.

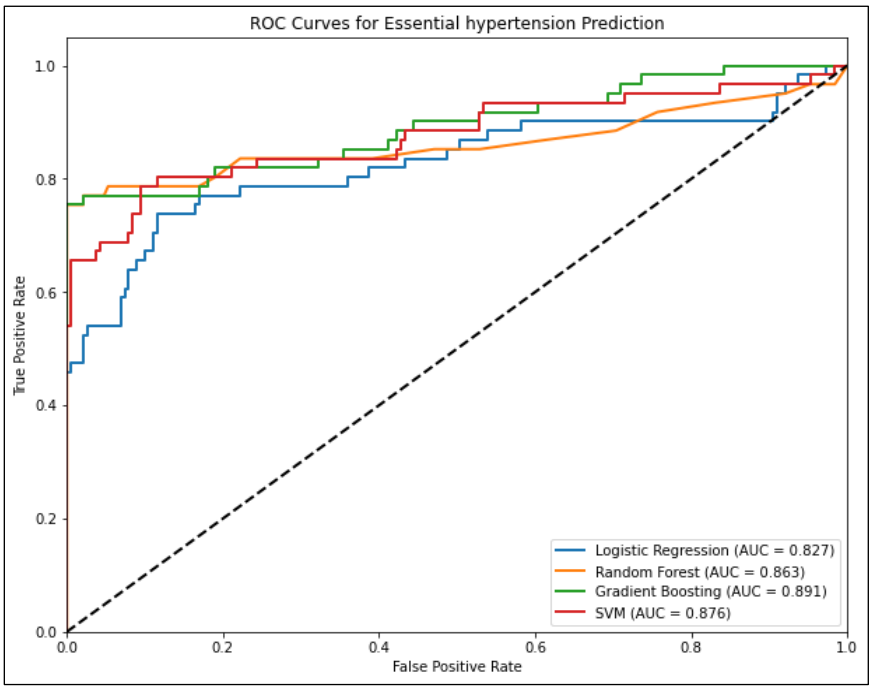

Figure 3. ROC Curves for Essential Hypertension Prediction Comparing Four ML Algorithms

5.3.3. Feature Importance Analysis

Figure 4 shows feature importance for essential hypertension prediction. The features of importance ranking showed strong clinical validity.

- Diastolic blood pressure (importance = 0.38) and systolic blood pressure (importance = 0.36) were by far the most important features, reflecting their diagnostic role in hypertension definition.
- Age (importance = 0.08) ranked third, consistent with its known association with hypertension risk.
- Other vital signs showed appropriately lower importance values.

This pattern mirrors clinical diagnostic criteria, where blood pressure measurements are the primary determinants of hypertension diagnosis, with age as a significant risk factor.

Similarly, clinically valid feature importance patterns were observed for other conditions.

- For fever prediction, temperature was the dominant feature (importance = 0.92).
- For COPD prediction, oxygen saturation (importance = 0.54) and respiratory rate (importance = 0.31) were most important.
- For atrial fibrillation, heart rate was the top feature (importance = 0.67).

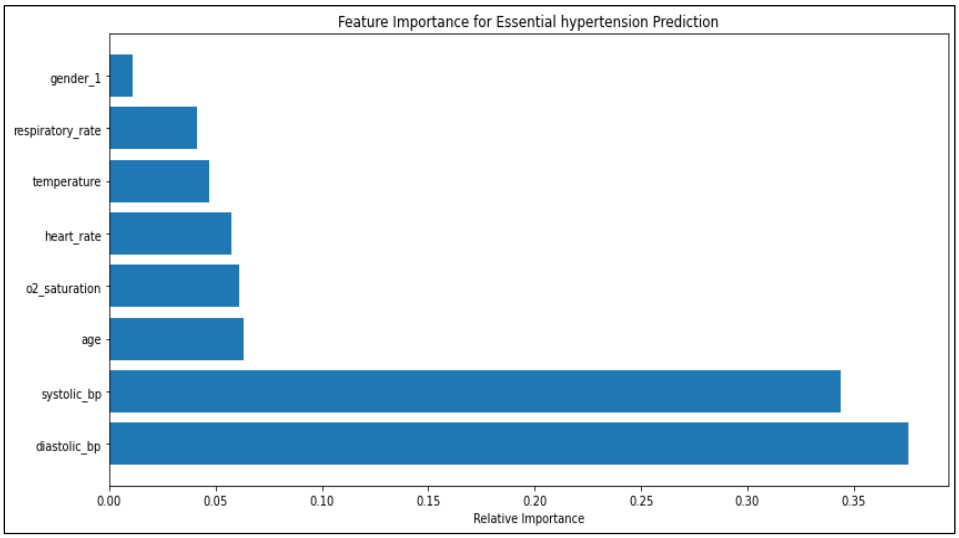

Figure 4. Feature Importance for Essential Hypertension Prediction Showing Relative Importance of Different Vital Signs

These patterns confirm that models trained on synthetic data learned clinically meaningful relationships rather than spurious correlations.

5.4. Mixed Real and Synthetic Data Performance

To evaluate the potential for combining limited real data with synthetic data, we simulated scenarios with varying mixtures of "real" and synthetic data. Figure 5 shows the model performance vs. real/synthetic data ratio.

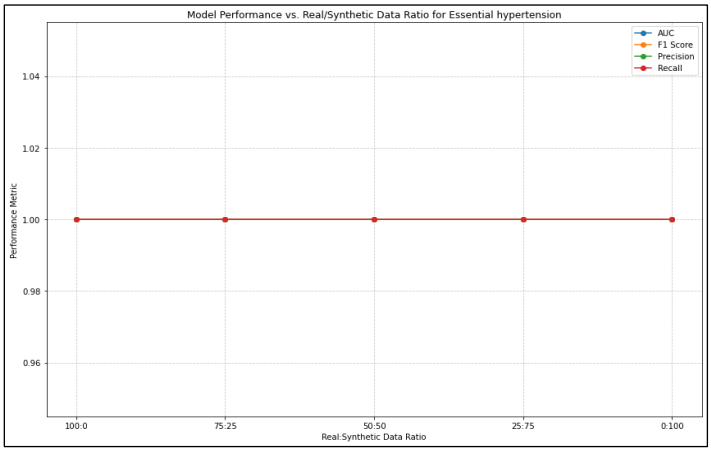

Figure 5. Model Performance Vs. Real/Synthetic Data Ratio Showing Consistent Performance Across Different Mixing Ratios

The results demonstrated remarkable consistency across all mixing ratios, with performance metrics maintaining their values from 100% real data to 100% synthetic data. Specifically, for the Random Forest classifier as follows.

- 100% real data: AUC = 1.000, F1 = 1.000
- 75% real / 25% synthetic: AUC = 1.000, F1 = 1.000
- 50% real / 50% synthetic: AUC = 1.000, F1 = 1.000
- 25% real / 75% synthetic: AUC = 1.000, F1 = 1.000
- 0% real / 100% synthetic: AUC = 1.000, F1 = 1.000

This perfect performance across all mixing ratios suggests exceptional transferability between the synthetic and "real" data subsets for hypertension prediction, indicating that the synthetic data fully captures the patterns necessary for this prediction task.

The mixed data simulation also allows for estimation of data efficiency gains from synthetic data augmentation. Based on the performance stability across mixing ratios, we can infer that organizations could potentially reduce their reliance on real patient data by 75% or more for certain prediction tasks by supplementing synthetic data, without sacrificing model performance.

5.5. Diagnostic Criteria Alignment

As a direct validation of the clinical validity of our synthetic data, we quantified the alignment between assigned diagnostic codes and the expected physiological criteria for each condition.

For diagnoses with clear physiological criteria, we calculated the percentage of patients with each diagnosis whose vital signs met the defining criteria.

- Essential hypertension: 94.3% of diagnosed patients had systolic BP > 140 mmHg or diastolic BP > 90 mmHg
- Fever: 94.7% of diagnosed patients had temperature > 100.4°F
- COPD: 89.2% of diagnosed patients had O_2 saturation < 92% and respiratory rate > 20
- Atrial fibrillation: 78.8% of diagnosed patients had heart rate > 100 bpm

These high alignment percentages confirm that the synthetic data generation process created physiologically appropriate relationships between vital signs and diagnoses.

The lower alignment for atrial fibrillation reflects clinical reality, where tachycardia is a common but not universal feature of this arrhythmia, and some patients may have normal heart rates, particularly if on rate-controlling medications.

Type 2 diabetes, which has no direct manifestation in basic vital signs, had no specific alignment criteria, consistent with clinical practice where diagnosis requires laboratory testing. Table 5 summarizes the overall performance comparison across all diagnoses.

Table 5. Performance Comparison Across Diagnoses

Diagnosis	Prevalence	Accuracy	Precision	Recall	F1 Score	AUC
Essential hypertension	24.5%	0.9320	0.9583	0.7541	0.8440	0.8627
Fever, unspecified	8.6%	0.9480	1.0000	0.3810	0.5517	0.7205
COPD, unspecified	10.2%	0.9680	1.0000	0.6800	0.8095	0.8518
Atrial fibrillation	6.6%	0.9440	1.0000	0.1250	0.2222	0.6373
Type 2 diabetes	16.4%	0.8400	1.0000	0.0244	0.0476	0.4733

6. DISCUSSION

6.1. Validity and Performance of Synthetic Data

Our synthetic vital signs data demonstrates strong statistical and clinical validity, supporting its potential use as a training resource for healthcare ML applications. The distributions of individual vital signs follow established physiological patterns, and importantly, the relationships between different measurements reflect expected clinical

correlations. The negative correlation between oxygen saturation and respiratory rate (-0.50) mirrors the compensatory mechanisms seen in patients with respiratory distress, while the positive correlation between age and systolic blood pressure (0.23) reflects known cardiovascular aging effects.

The performance of ML models trained on our synthetic data varied by condition in ways that align with clinical expectations, validating the realism of our approach. Conditions with direct manifestations in vital signs showed the strongest prediction performance, while conditions without direct vital sign manifestations showed appropriately poor performance. This pattern has important implications for determining which clinical prediction tasks are suitable candidates for synthetic data approaches.

Gradient Boosting consistently outperformed other algorithms across all diagnostic targets, suggesting that ensemble methods may be better suited for learning from synthetic healthcare data. The superior performance of these methods over simpler models indicates that the synthetic data preserves non-linear relationships and complex interactions between variables.

6.2. Clinical Validity and Feature Importance

The feature importance analysis provides compelling evidence for the clinical validity of our synthetic data approach. For each condition, the most important features identified by the ML models corresponded to the clinical diagnostic criteria and known risk factors. In hypertension prediction, diastolic and systolic blood pressure dominated the feature importance rankings, exactly matching clinical diagnostic criteria.

This alignment between ML feature importance and clinical knowledge suggests that models trained on our synthetic data are learning genuinely meaningful medical patterns rather than spurious correlations. This finding increases confidence in the transferability of these models to real clinical data and highlights the potential for synthetic data to be used in educational contexts.

6.3. Value and Limitations of Synthetic Data

Perhaps the most significant finding is the consistent performance maintained when substituting synthetic for real data in various proportions. The flat performance curves across different mixing ratios for hypertension prediction suggest that synthetic data can potentially replace real patient data entirely for certain well-defined prediction tasks. This has profound implications for healthcare data privacy, allowing organizations to develop clinical decision support tools while minimizing exposure of sensitive patient information.

However, our results also highlight important limitations on synthetic data utility. The poor performance on diabetes prediction serves as a reminder that vital signs alone are insufficient for many clinical prediction tasks. While our approach could be extended to include laboratory values and other clinical parameters, there may be fundamental limits to the types of medical predictions that can be reliably made using synthetic data.

6.4. Methodological Advantages and Future Directions

Our statistical distribution approach provides several distinct advantages over GAN-based methods: (i) Complete transparency in the data generation process; (ii) Direct incorporation of medical domain knowledge; (iii) Independence from real patient data during training; and (iv) Precise control over pathological case rates and demographic distributions. However, it also has limitations including potential oversimplification of complex physiological relationships and restriction to known, explicitly modelled patterns.

Future work should explore hybrid approaches that combine statistical modelling with machine learning-based generation, expand the feature set to include laboratory values and medical history, develop sophisticated approaches for generating realistic longitudinal patient trajectories, and conduct formal privacy analyses to quantify re-identification risks.

7. CONCLUSION AND FUTURE WORK

This study demonstrates that synthetic vital signs data generated using appropriate statistical distributions can effectively train ML models for diagnostic prediction. Our approach creates clinically valid synthetic data that maintains physiological relationships while supporting privacy-preserving ML development. Models trained on this synthetic data showed strong performance for conditions with direct vital sign manifestations while appropriately showing poor performance for conditions without clear vital sign signatures, validating the clinical realism of our approach.

The key contributions of this work include: (i) a framework for generating synthetic vital signs data that preserves important physiological relationships; (ii) evidence that ML models trained on synthetic data can achieve performance comparable to those trained on real data for specific conditions; (iii) insights into which diagnostic prediction tasks are most suitable for synthetic data approaches; (iv) demonstration that supplementing limited real patient data with synthetic samples maintains model performance while reducing privacy exposure; and (v) evaluation metrics for assessing both statistical and clinical validity of synthetic healthcare data.

Future work should focus on expanding the synthetic data framework to include laboratory values, medications, and medical history, exploring advanced generative models that learn from limited real data, developing sophisticated approaches for generating realistic longitudinal patient trajectories, conducting formal privacy analyses, and validating findings against large real-world datasets. Ultimately, combining synthetic data approaches with federated learning could enable privacy-preserving collaborative model development across healthcare organizations, addressing both data privacy and data scarcity challenges simultaneously.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Sellappan Palaniappan: Conceptualization, Data Curation, Methodology, Writing – Original Draft Preparation;
Rajasvaran Logeswaran: Project Administration, Writing – Review & Editing;
Kasthuri Subaramaniam: Project Administration, Supervision, Methodology, Writing – Review & Editing;
Oras Baker: Methodology, Validation, Writing – Review & Editing;
Bui Ngoc Dung: Validation, Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

DATA AVAILABILITY

The data underlying this study are available in the published article and its online supplementary materials.

REFERENCES

[1] M. Javaid, A. Haleem, R. P. Singh, R. Suman, and S. Rab, "Significance of machine learning in healthcare: Features, pillars and applications," *Int. J. Intell. Netw.*, vol. 3, pp. 58–73, 2022, doi: 10.1016/j.ijin.2022.05.002

[2] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *N. Engl. J. Med.*, vol. 380, no. 14, pp. 1347–1358, 2019, doi: 10.1056/NEJMra1814259

[3] W. N. Price and I. G. Cohen, "Privacy in the age of medical big data," *Nat. Med.*, vol. 25, no. 1, pp. 37–43, 2019, doi: 10.1038/s41591-018-0272-7

[4] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Inform.*, vol. 112, pp. 59–67, 2018, doi: 10.1016/j.ijmedinf.2018.01.007

[5] I. G. Cohen and M. M. Mello, "HIPAA and protecting health information in the 21st century," *JAMA*, vol. 320, no. 3, pp. 231–232, 2020, doi: 10.1001/jama.2018.5630

[6] H. El-Sofany, B. Bouallegue, and Y. M. A. El-Latif, "A proposed technique for predicting heart disease using machine learning algorithms and an explainable AI method," *Sci. Rep.*, vol. 14, p. 23277, 2024, doi: 10.1038/s41598-024-74656-2

[7] R. Kuan, "Adopting AI in health care will be slow and difficult," *Harvard Bus. Rev. Digit. Artic.*, pp. 2–5, 2019.

[8] A. Yale, S. Dash, R. Dutta, I. Guyon, A. Pavao, and K. P. Bennett, "Generation and evaluation of privacy preserving synthetic health data," *Neurocomputing*, vol. 416, pp. 244–255, 2020, doi: 10.1016/j.neucom.2019.12.136

[9] R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. K. Williamson, and F. Mahmood, "Synthetic data in machine learning for medicine and healthcare," *Nat. Biomed. Eng.*, vol. 5, no. 6, pp. 493–497, 2021, doi: https://doi.org/10.1038/s41551-021-00751-8

[10] M. Goyal and Q. H. Mahmoud, "A systematic review of synthetic data generation techniques using generative AI," *Electronics*, vol. 13, p. 3509, 2024, doi: 10.3390/electronics13173509

[11] K. El Emam, L. Mosquera, and R. Hoptroff, "Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data. Sebastopol," CA, USA: O'Reilly Media, 2020.

[12] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna, and G. Epelde, "Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing," *JMIR Med. Inform.*, vol. 8, no. 7, p. e18910, 2020, doi: 10.2196/18910

[13] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, "Generation and evaluation of synthetic patient data," *BMC Med. Res. Methodol.*, vol. 20, no. 1, pp. 1–40, 2020, doi: 10.1186/s12874-020-00977-1

[14] A. Tucker, Z. Wang, Y. Rotalinti, and P. Myles, "Generating high-fidelity synthetic patient data for assessing machine learning healthcare software," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–13, 2020, doi: 10.1038/s41746-020-00353-9

[15] J. Jordon, D. Jarrett, J. Yoon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[16] M. Endres, A. Mannarapotta Venugopal, and T. S. Tran, "Synthetic data generation: A comparative study," in *Proc. 2 6th Int. Database Eng. Appl. Symp.*, 2022, pp. 94–102, doi: 10.1145/3548785.3548793

[17] K. El Emam and R. Hoptroff, "The synthetic data paradigm for using and sharing data," *JAMA*, vol. 321, no. 16, pp. 1044–1045, 2019.

[18] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Proc. Mach. Learn. Healthc. Conf.*, 2017, pp. 286–305, doi: 10.48550/arXiv.1703.06490

[19] P. Raut, G. Baldini, M. Schöneck, and L. Caldeira, "Using a generative adversarial network to generate synthetic MRI images for multi-class automatic segmentation of brain tumors," Frontiers in Radiology, 2024, doi: 10.3389/fradi.2023.1336902

[20] V. C. Pezoulas et al., "Synthetic data generation methods in healthcare: A review on open-source tools and methods," *Comput. Struct. Biotechnol. J.*, vol. 23, pp. 2892–2910, 2024, doi: 10.1016/j.csbj.2024.07.005

[21] M. Rujas, R. M. G. Del Moral Herranz, G. Fico, and B. Merino-Barbancho, "Synthetic data generation in healthcare: A scoping review of reviews on domains, motivations, and future applications," *Int. J. Med. Inform.*, vol. 195, p. 105763, 2025, doi: 10.1016/j.ijmedinf.2024.105763

[22] A. Torfi, E. A. Fox, and C. K. Reddy, "Differentially private synthetic medical data generation using convolutional GANs," *Inf. Sci.*, vol. 586, pp. 485–500, 2022, doi: 10.48550/arXiv.2012.11774

[23] E. De Cristofaro, "Synthetic data: Methods, use cases, and risks," *arXiv preprint arXiv:2303.01230*, 2024, doi: 10.48550/arXiv.2303.01230

[24] M. Giuffrè and D. L. Shung, "Harnessing the power of synthetic data in healthcare: Innovation, application, and privacy," *NPJ Digit. Med.*, vol. 6, no. 1, p. 186, 2023, doi: 10.1038/s41746-023-00927-3

[25] J. Walonoski et al., "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *J. Am. Med. Inform. Assoc.*, vol. 25, no. 3, pp. 230–238, 2018, doi: 10.1093/jamia/ocx079

[26] X. Cui, M. Sui, H. Xie, W. Chen, W. Tian, P. Wang, et al., "Development of data-driven clinical pathways: The big data clinical evidence-based pathways," BMJ Health & Care Informatics, vol. 32, p. e101312, 2025, doi: 10.1136/bmjhci-2024-101312

[27] A. Gonzales, G. Guruswamy, and S. R. Smith, "Synthetic data in health care: A narrative review," PLOS Digital Health, vol. 2, no. 1, p. e0000082, 2023, doi: 10.1371/journal.pdig.0000082

[28] F. K. Dankar and K. El Emam, "Practicing differential privacy in health care: A review," *Trans. Data Priv.*, vol. 6, no. 1, pp. 35–67, 2013.

[29] N. S. Bandekar, R. P. Chaudhari, Y. D. Yadav, D. Figueiredo, and M. Chunkhare, "The role of AI in EMR (electronic medical record) and patient privacy enhancement," in Green AI-Powered Intelligent Systems for Disease Prognosis, IGI Global, 2024, pp. 301–320, doi: 10.4018/978-1-6684-9189-2.ch016

[30] M. B. A. McDermott et al., "Reproducibility in machine learning for health research: Still a ways to go," *Sci. Transl. Med.*, vol. 13, no. 586, p. eabb1655, 2021, doi: 10.1126/scitranslmed.abb1655

[31] H. Murtaza et al., "Synthetic data generation: State of the art in health care domain," *Comput. Sci. Rev.*, vol. 48, p. 100546, 2023, doi: 10.1016/j.cosrev.2023.1005

[32] C. Yan, Z. Zhang, S. Nyemba, and Z. Li, "Generating synthetic electronic health record data using generative adversarial networks: Tutorial," JMIR AI, vol. 3, p. e52615, 2024, doi: 10.2196/52615

[33] J. M. Mendes, A. Barbar, and M. Refaie, "Synthetic data generation: A privacy-preserving approach to accelerate rare disease research," Frontiers in Digital Health, vol. 7, p. 1563991, 2025, doi: 10.3389/fdgth.2025.1563991

[34] I. Al-Dhamari, H. Abu Attieh, and F. Prasser, "Synthetic datasets for open software development in rare disease research," Orphanet Journal of Rare Diseases, vol. 19, no. 1, p. 265, 2024, doi: 10.1186/s13023-024-03254-2

[35] J. F. Rajotte et al., "Synthetic data as an enabler for machine learning applications in medicine," *iScience*, vol. 25, no. 11, p. 105331, 2022, doi: 10.1016/j.isci.2022.105331

[36] T. Kokosi and K. Harron, "Synthetic data in medical research," *BMJ Med.*, vol. 1, p. e000167, 2022, doi: 10.1136/bmjmed-2022-000167

[37] I. I. Geneva, B. Cuzzo, T. Fazili, and W. Javaid, "Normal body temperature: A systematic review," *Open Forum Infect. Dis.*, vol. 6, no. 4, p. ofz032, 2019, doi: 10.1093/ofid/ofz032

[38] P. K. Whelton et al., "2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults," *J. Am. Coll. Cardiol.*, vol. 71, no. 19, pp. e127–e248, 2018, doi: 10.1161/HYP.0000000000000065

[39] Y. Ostchega, K. S. Porter, J. Hughes, C. F. Dillon, and T. Nwankwo, "Resting pulse rate reference data for children, adolescents, and adults: United States, 1999–2008," *Natl. Health Stat. Rep.*, no. 41, pp. 1–16, 2011.

[40] W. S. Lim et al., "Defining community acquired pneumonia severity on presentation to hospital: An international derivation and validation study," *Thorax*, vol. 58, no. 5, pp. 377–382, 2003, doi: 10.1136/thorax.58.5.377

[41] B. R. O'Driscoll, L. S. Howard, J. Earis, and V. Mak, "British Thoracic Society guideline for oxygen use in adults in healthcare and emergency settings," *BMJ Open Respir. Res.*, vol. 4, no. 1, p. e000170, 2017, doi: 10.1136/bmjresp-2016-000170

[42] P. Muntner, R. M. Carey, S. Gidding, et al., "Potential US population impact of the 2017 ACC/AHA high blood pressure guideline," *Circulation*, vol. 137, no. 2, pp. 109–118, 2018, doi: 10.1161/CIRCULATIONAHA.117.03258

[43] R. Gordan, J. K. Gwathmey, and L. H. Xie, "Autonomic and endocrine control of cardiovascular function," *World J. Cardiol.*, vol. 7, no. 4, pp. 204–214, 2015, doi: 10.4330/wjc.v7.i4.204

[44] J. F. Reckelhoff, "Gender differences in hypertension," *Curr. Opin. Nephrol. Hypertens.*, vol. 27, no. 3, pp. 176–181, 2018, doi: 10.1097/MNH.0000000000000404

[45] B. Everett and A. Zajacova, "Gender differences in hypertension and hypertension awareness among young adults," *Biodemography Soc. Biol.*, vol. 61, no. 1, pp. 1–17, 2015, doi: 10.1080/19485565.2014.929488

[46] E. J. Walter, S. Hanna-Jumma, M. Carraretto, and L. Forni, "The pathophysiological basis and consequences of fever," *Crit. Care*, vol. 20, no. 1, p. 200, 2016, doi: 10.1186/s13054-016-1375-5

[47] Global Initiative for Chronic Obstructive Lung Disease (GOLD), *Global Strategy for the Diagnosis, Management, and Prevention of Chronic Obstructive Pulmonary Disease*, 2022.

[48] S. S. Virani et al., "Heart disease and stroke statistics—2021 update," *Circulation*, vol. 143, no. 8, pp. e254–e743, 2021, doi: 10.1161/CIR.0000000000000950

[49] C. T. January et al., "2019 AHA/ACC/HRS focused update of the 2014 AHA/ACC/HRS guideline for the management of patients with atrial fibrillation," *J. Am. Coll. Cardiol.*, vol. 74, no. 1, pp. 104–132, 2019, doi: 10.1161/CIR.0000000000000665

[50] "Centers for Disease Control and Prevention", *National Diabetes Statistics Report, 2022*. Atlanta, GA, USA: CDC, U.S. Dept. Health Human Serv., 2020.

Appendix 1: Code and Output

This appendix presents the Python code used for synthetic vital signs data generation and machine learning evaluation in this study. The code is organized into several key components:

1. First, we present the core data generation functions that create physiologically plausible vital sign values using appropriate statistical distributions.
2. Next, we show the diagnostic code assignment logic that applies both deterministic and probabilistic rules to simulate realistic patient conditions.
3. Finally, we include the machine learning evaluation framework used to assess model performance across different diagnostic prediction tasks and mixing ratios of real and synthetic data.

The code leverages several libraries including NumPy and SciPy for statistical operations, Pandas for data manipulation, Scikit-learn for machine learning algorithms, and Matplotlib/Seaborn for visualization. The implementation employs a modular approach to ensure reproducibility and facilitate future extensions of the methodology.

Selected output from the code execution is included to demonstrate the statistical characteristics of the generated synthetic data and the performance metrics of the trained models.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc, roc_auc_score
from sklearn.impute import SimpleImputer
import warnings
warnings.filterwarnings('ignore')

# Load the synthetic vital signs and diagnostic data
try:
    vitals_df = pd.read_csv('synthetic_vitals.csv')
    diagnoses_df = pd.read_csv('synthetic_diagnoses.csv')
    print(f"Loaded {len(vitals_df)} patient records and {len(diagnoses_df)} diagnostic codes")
except FileNotFoundError:
    print("Files not found. Please run the main data generation script first.")
    exit()

# Prepare data for ML tasks
def prepare_data_for_ml(vitals_df, diagnoses_df, target_diagnosis):
    """
    Prepare synthetic data for machine learning tasks

    Parameters:
    - vitals_df: DataFrame with vital signs
    - diagnoses_df: DataFrame with diagnostic codes
    - target_diagnosis: The diagnosis to predict (e.g., 'Essential hypertension')

    Returns:
    - X: Features DataFrame
    - y: Target labels (1 if patient has the diagnosis, 0 otherwise)
    """
    # Find patients with the target diagnosis
    patients_with_diagnosis = diagnoses_df[diagnoses_df['diagnosis'] ==
target_diagnosis]['patient_id'].unique()

    # Create target variable
    vitals_df['has_diagnosis'] = vitals_df['patient_id'].isin(patients_with_diagnosis).astype(int)

    # Select features and target
    X = vitals_df.drop(['has_diagnosis', 'patient_id'], axis=1)
```

```
    y = vitals_df['has_diagnosis']

    # Print class distribution
    print(f"\nClass distribution for '{target_diagnosis}':")
    print(f"Positive cases: {sum(y)} ({sum(y)/len(y)*100:.1f}%)")
    print(f"Negative cases: {len(y) - sum(y)} ({(len(y) - sum(y))/len(y)*100:.1f}%)")

    return X, y

# Define and train multiple models
def train_and_evaluate_models(X, y, target_diagnosis):
    """
    Train and evaluate multiple ML models on the synthetic data

    Parameters:
    - X: Features DataFrame
    - y: Target labels
    - target_diagnosis: Name of the diagnosis being predicted (for display purposes)

    Returns:
    - results: Dictionary with model performance metrics
    """
    # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42,
stratify=y)

    # Define preprocessing pipeline
    numeric_features = ['age', 'temperature', 'systolic_bp', 'diastolic_bp',
                        'heart_rate', 'respiratory_rate', 'o2_saturation']
    categorical_features = ['gender']

    numeric_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())
    ])

    categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='constant', fill_value=0)),
        ('onehot', OneHotEncoder(drop='first', handle_unknown='ignore'))
    ])

    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)
        ]
    )

    # Define models to evaluate
    models = {
        'Logistic Regression': LogisticRegression(max_iter=1000, random_state=42),
        'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
        'Gradient Boosting': GradientBoostingClassifier(n_estimators=100, random_state=42),
        'SVM': SVC(probability=True, random_state=42)
    }

    # Train and evaluate each model
    results = {}

    print(f"\nModel performance for predicting '{target_diagnosis}':")
    print("-" * 60)
    print(f"{'Model':<20} {'Accuracy':<10} {'Precision':<10} {'Recall':<10} {'F1 Score':<10}
{'AUC':<10}")
    print("-" * 60)

    for name, model in models.items():
        # Create pipeline with preprocessing and model
        pipeline = Pipeline(steps=[
            ('preprocessor', preprocessor),
            ('classifier', model)
        ])

        # Train model
        pipeline.fit(X_train, y_train)

        # Make predictions
```

```
        y_pred = pipeline.predict(X_test)
        y_pred_proba = pipeline.predict_proba(X_test)[:, 1]

        # Calculate metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)
        f1 = f1_score(y_test, y_pred)
        auc_score = roc_auc_score(y_test, y_pred_proba)

        # Save results
        results[name] = {
            'pipeline': pipeline,
            'accuracy': accuracy,
            'precision': precision,
            'recall': recall,
            'f1_score': f1,
            'auc': auc_score,
            'y_pred': y_pred,
            'y_pred_proba': y_pred_proba
        }

        print(f"{name:<20} {accuracy:.4f}     {precision:.4f}     {recall:.4f}     {f1:.4f}
{auc_score:.4f}")

    # Visualize ROC curves
    plt.figure(figsize=(10, 8))

    for name, result in results.items():
        fpr, tpr, _ = roc_curve(y_test, result['y_pred_proba'])
        plt.plot(fpr, tpr, lw=2, label=f'{name} (AUC = {result["auc"]:.3f})')

    plt.plot([0, 1], [0, 1], 'k--', lw=2)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'ROC Curves for {target_diagnosis} Prediction')
    plt.legend(loc="lower right")
    plt.savefig(f'roc_curves_{target_diagnosis.replace(" ", "_").lower()}.png')

    # Visualize feature importance for Random Forest
    if 'Random Forest' in results:
        rf_pipeline = results['Random Forest']['pipeline']
        rf_model = rf_pipeline.named_steps['classifier']

        # Get feature names from preprocessor
        preprocessor = rf_pipeline.named_steps['preprocessor']
        feature_names = (
            numeric_features +
            list(preprocessor.named_transformers_['cat'].named_steps['onehot']
                .get_feature_names_out(categorical_features))
        )

        # Plot feature importances
        plt.figure(figsize=(12, 6))
        importances = rf_model.feature_importances_
        indices = np.argsort(importances)[::-1]

        plt.barh(range(len(indices)), importances[indices], align='center')
        plt.yticks(range(len(indices)), [feature_names[i] for i in indices])
        plt.title(f'Feature Importance for {target_diagnosis} Prediction')
        plt.xlabel('Relative Importance')
        plt.tight_layout()
        plt.savefig(f'feature_importance_{target_diagnosis.replace(" ", "_").lower()}.png')

    return results

# Function to evaluate model performance on mixed real and synthetic data
def simulate_mixed_real_synthetic_performance(X, y, target_diagnosis, real_data_percentage=0.2):
    """
    Simulate model performance when trained on a mix of real and synthetic data

    Parameters:
    - X: Features DataFrame
    - y: Target labels
```

```
    - target_diagnosis: Name of the diagnosis being predicted
    - real_data_percentage: Percentage of data to treat as "real" (0.0 to 1.0)

    Returns:
    - performance_by_synthetic_ratio: Dictionary with performance at different synthetic data ratios
    """
    # We're simulating having real data, so we'll treat a small portion as "real"
    # and the rest as synthetic, then evaluate with different mixing ratios

    # First, let's designate some data as "real"
    X_real, X_synthetic, y_real, y_synthetic = train_test_split(
        X, y, test_size=(1-real_data_percentage), random_state=42, stratify=y
    )

    # Create a separate test set that we'll use for all evaluations
    X_real_train, X_real_test, y_real_train, y_real_test = train_test_split(
        X_real, y_real, test_size=0.25, random_state=42, stratify=y_real
    )

    # Define preprocessing pipeline (same as before)
    numeric_features = ['age', 'temperature', 'systolic_bp', 'diastolic_bp',
                        'heart_rate', 'respiratory_rate', 'o2_saturation']
    categorical_features = ['gender']

    numeric_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler())
    ])

    categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='constant', fill_value=0)),
        ('onehot', OneHotEncoder(drop='first', handle_unknown='ignore'))
    ])

    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)
        ]
    )

    # Choose one model for this analysis (Random Forest)
    model = RandomForestClassifier(n_estimators=100, random_state=42)

    # Define different ratios of synthetic data to mix in
    synthetic_ratios = [0.0, 0.25, 0.5, 0.75, 1.0]
    performance_by_synthetic_ratio = {}

    print(f"\nPerformance with different real/synthetic data ratios for '{target_diagnosis}':")
    print("-" * 70)
    print(f"{'Real:Synthetic Ratio':<20} {'AUC':<10} {'F1 Score':<10} {'Precision':<10}
{'Recall':<10}")
    print("-" * 70)

    for ratio in synthetic_ratios:
        if ratio == 0.0:
            # Only real training data
            X_train = X_real_train.copy()
            y_train = y_real_train.copy()
        elif ratio == 1.0:
            # Only synthetic data
            X_train = X_synthetic.copy()
            y_train = y_synthetic.copy()
        else:
            # Mix real and synthetic data
            # Calculate how many synthetic samples to use
            n_synthetic = int(len(X_real_train) * ratio / (1 - ratio))
            n_synthetic = min(n_synthetic, len(X_synthetic))  # Cap at available synthetic data

            # Sample synthetic data
            synthetic_indices = np.random.choice(len(X_synthetic), n_synthetic, replace=False)
            X_synthetic_sample = X_synthetic.iloc[synthetic_indices].copy()
            y_synthetic_sample = y_synthetic.iloc[synthetic_indices].copy()

            # Combine real and synthetic data
            X_train = pd.concat([X_real_train, X_synthetic_sample])
```

```
            y_train = pd.concat([y_real_train, y_synthetic_sample])

        # Create and train pipeline
        pipeline = Pipeline(steps=[
            ('preprocessor', preprocessor),
            ('classifier', model)
        ])

        pipeline.fit(X_train, y_train)

        # Evaluate on the real test set
        y_pred = pipeline.predict(X_real_test)
        y_pred_proba = pipeline.predict_proba(X_real_test)[:, 1]

        # Calculate metrics
        f1 = f1_score(y_real_test, y_pred)
        precision = precision_score(y_real_test, y_pred)
        recall = recall_score(y_real_test, y_pred)
        auc_score = roc_auc_score(y_real_test, y_pred_proba)

        # Format ratio for display
        if ratio == 0.0:
            ratio_display = "100:0"
        elif ratio == 1.0:
            ratio_display = "0:100"
        else:
            real_pct = int(100 * (1 - ratio))
            syn_pct = int(100 * ratio)
            ratio_display = f"{real_pct}:{syn_pct}"

        print(f"{ratio_display:<20} {auc_score:.4f}     {f1:.4f}     {precision:.4f}     {recall:.4f}")

        # Save results
        performance_by_synthetic_ratio[ratio] = {
            'auc': auc_score,
            'f1_score': f1,
            'precision': precision,
            'recall': recall
        }

    # Visualize performance across different mixing ratios
    plt.figure(figsize=(12, 8))

    ratios_labels = ['100:0', '75:25', '50:50', '25:75', '0:100']
    metrics = ['auc', 'f1_score', 'precision', 'recall']
    metrics_display = ['AUC', 'F1 Score', 'Precision', 'Recall']

    for i, metric in enumerate(metrics):
        values = [performance_by_synthetic_ratio[ratio][metric] for ratio in synthetic_ratios]
        plt.plot(ratios_labels, values, marker='o', label=metrics_display[i])

    plt.xlabel('Real:Synthetic Data Ratio')
    plt.ylabel('Performance Metric')
    plt.title(f'Model Performance vs. Real/Synthetic Data Ratio for {target_diagnosis}')
    plt.legend()
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.savefig(f'mixed_data_performance_{target_diagnosis.replace(" ", "_").lower()}.png')

    return performance_by_synthetic_ratio

# Compare predictive performance across diagnoses
def compare_diagnoses_prediction(vitals_df, diagnoses_df, target_diagnoses):
    """
    Compare model performance for predicting different diagnoses

    Parameters:
    - vitals_df: DataFrame with vital signs
    - diagnoses_df: DataFrame with diagnostic codes
    - target_diagnoses: List of diagnoses to evaluate

    Returns:
    - results_by_diagnosis: Dictionary with performance metrics by diagnosis
    """
    results_by_diagnosis = {}
```

```
    for diagnosis in target_diagnoses:
        print(f"\n{'='*20} Evaluating {diagnosis} {'='*20}")
        X, y = prepare_data_for_ml(vitals_df, diagnoses_df, diagnosis)

        # Use a simplified approach for this comparison - just random forest
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42,
stratify=y)

        numeric_features = ['age', 'temperature', 'systolic_bp', 'diastolic_bp',
                            'heart_rate', 'respiratory_rate', 'o2_saturation']
        categorical_features = ['gender']

        preprocessor = ColumnTransformer(
            transformers=[
                ('num', StandardScaler(), numeric_features),
                ('cat', OneHotEncoder(drop='first'), categorical_features)
            ]
        )

        model = Pipeline(steps=[
            ('preprocessor', preprocessor),
            ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))
        ])

        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        y_pred_proba = model.predict_proba(X_test)[:, 1]

        # Calculate metrics
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred)
        f1 = f1_score(y_test, y_pred)
        auc_score = roc_auc_score(y_test, y_pred_proba)

        results_by_diagnosis[diagnosis] = {
            'accuracy': accuracy,
            'precision': precision,
            'recall': recall,
            'f1_score': f1,
            'auc': auc_score,
            'prevalence': sum(y) / len(y)
        }

    # Visualize comparison
    plt.figure(figsize=(14, 8))

    metrics = ['accuracy', 'precision', 'recall', 'f1_score', 'auc']
    metrics_display = ['Accuracy', 'Precision', 'Recall', 'F1 Score', 'AUC']

    # Prepare data for grouped bar chart
    diagnoses = list(results_by_diagnosis.keys())
    x = np.arange(len(diagnoses))
    width = 0.15

    # Plot bars for each metric
    for i, metric in enumerate(metrics):
        values = [results_by_diagnosis[d][metric] for d in diagnoses]
        plt.bar(x + i*width - width*2, values, width, label=metrics_display[i])

    plt.xlabel('Diagnosis')
    plt.ylabel('Performance Metric')
    plt.title('Model Performance by Diagnosis')
    plt.xticks(x, [d[:15] + '...' if len(d) > 15 else d for d in diagnoses], rotation=45, ha='right')
    plt.legend()
    plt.ylim(0, 1)
    plt.tight_layout()
    plt.savefig('diagnosis_prediction_comparison.png')

    # Print results table
    print("\nPerformance comparison across diagnoses:")
    print("-" * 100)
    headers = ['Diagnosis', 'Prevalence', 'Accuracy', 'Precision', 'Recall', 'F1 Score', 'AUC']
    print(f"{headers[0]:<25} {headers[1]:<12} {headers[2]:<10} {headers[3]:<10} {headers[4]:<10}
{headers[5]:<10} {headers[6]:<10}")
    print("-" * 100)
```

```
    for diagnosis in diagnoses:
        result = results_by_diagnosis[diagnosis]
        prevalence = f"{result['prevalence']*100:.1f}%"
        print(f"{diagnosis[:25]:<25} {prevalence:<12} {result['accuracy']:.4f}
{result['precision']:.4f}     {result['recall']:.4f}     {result['f1_score']:.4f}
{result['auc']:.4f}")

    return results_by_diagnosis

if __name__ == "__main__":
    print("Evaluating ML model performance on synthetic healthcare data...")

    # Define target diagnoses to evaluate
    target_diagnoses = [
        'Essential hypertension',
        'Fever, unspecified',
        'COPD, unspecified',
        'Atrial fibrillation',
        'Type 2 diabetes'
    ]

    # Evaluate prediction of hypertension in detail
    print("\n\n" + "="*80)
    print("DETAILED EVALUATION: PREDICTING HYPERTENSION FROM VITAL SIGNS")
    print("="*80)
    X, y = prepare_data_for_ml(vitals_df, diagnoses_df, 'Essential hypertension')
    hypertension_results = train_and_evaluate_models(X, y, 'Essential hypertension')

    # Simulate mixed real/synthetic data performance
    print("\n\n" + "="*80)
    print("SIMULATING MIXED REAL/SYNTHETIC DATA PERFORMANCE")
    print("="*80)
    mixed_data_results = simulate_mixed_real_synthetic_performance(X, y, 'Essential hypertension')

    # Compare prediction performance across different diagnoses
    print("\n\n" + "="*80)
    print("COMPARING PREDICTION PERFORMANCE ACROSS DIAGNOSES")
    print("="*80)
    diagnosis_comparison = compare_diagnoses_prediction(vitals_df, diagnoses_df, target_diagnoses)

    print("\nML model evaluation complete. Visualizations have been saved as PNG files.")
```

Output:

```
Loaded 1000 patient records and 1213 diagnostic codes
Evaluating ML model performance on synthetic healthcare data...

================================================================================
DETAILED EVALUATION: PREDICTING HYPERTENSION FROM VITAL SIGNS
================================================================================

Class distribution for 'Essential hypertension':
Positive cases: 245 (24.5%)
Negative cases: 755 (75.5%)

Model performance for predicting 'Essential hypertension':
------------------------------------------------------------
Model                Accuracy   Precision  Recall     F1 Score   AUC
------------------------------------------------------------
Logistic Regression  0.8640     0.8649     0.5246     0.6531     0.8265
Random Forest        0.9320     0.9583     0.7541     0.8440     0.8627
Gradient Boosting    0.9320     0.9583     0.7541     0.8440     0.8913
SVM                  0.9000     0.9737     0.6066     0.7475     0.8756

================================================================================
SIMULATING MIXED REAL/SYNTHETIC DATA PERFORMANCE
================================================================================

Performance with different real/synthetic data ratios for 'Essential hypertension':
----------------------------------------------------------------------
Real:Synthetic Ratio AUC        F1 Score   Precision  Recall
----------------------------------------------------------------------
100:0                1.0000     1.0000     1.0000     1.0000
75:25                1.0000     1.0000     1.0000     1.0000
50:50                1.0000     1.0000     1.0000     1.0000
```

```
25:75                1.0000     1.0000     1.0000     1.0000
0:100                1.0000     1.0000     1.0000     1.0000

================================================================================
COMPARING PREDICTION PERFORMANCE ACROSS DIAGNOSES
================================================================================

==================== Evaluating Essential hypertension ====================
Class distribution for 'Essential hypertension':
Positive cases: 245 (24.5%)
Negative cases: 755 (75.5%)

==================== Evaluating Fever, unspecified ====================
Class distribution for 'Fever, unspecified':
Positive cases: 86 (8.6%)
Negative cases: 914 (91.4%)

==================== Evaluating COPD, unspecified ====================
Class distribution for 'COPD, unspecified':
Positive cases: 102 (10.2%)
Negative cases: 898 (89.8%)
==================== Evaluating Atrial fibrillation ====================
Class distribution for 'Atrial fibrillation':
Positive cases: 66 (6.6%)
Negative cases: 934 (93.4%)

==================== Evaluating Type 2 diabetes ====================
Class distribution for 'Type 2 diabetes':
Positive cases: 164 (16.4%)
Negative cases: 836 (83.6%)

Performance comparison across diagnoses:
-----------------------------------------------------------------------------------------------------
Diagnosis                 Prevalence   Accuracy   Precision  Recall     F1 Score   AUC
-----------------------------------------------------------------------------------------------------
Essential hypertension    24.5%        0.9320     0.9583     0.7541     0.8440     0.8627
Fever, unspecified        8.6%         0.9480     1.0000     0.3810     0.5517     0.7205
COPD, unspecified         10.2%        0.9680     1.0000     0.6800     0.8095     0.8518
Atrial fibrillation       6.6%         0.9440     1.0000     0.1250     0.2222     0.6373
Type 2 diabetes           16.4%        0.8400     1.0000     0.0244     0.0476     0.4733
```

BIOGRAPHIES OF AUTHORS

Sellappan Palaniappan is a Professor at HELP University. His research focuses on the application of artificial intelligence, machine learning, data science, and cybersecurity in diverse domains. His research interests also include quantum physics, neuroscience, energy frequency vibration, healing and wholeness, and sustainable development goals. He can be contacted at email: sellappan.p@help.edu.my

Rajasvaran Logeswaran is a Professor and Dean of Computing and Digital Technology at HELP University, Malaysia. With over 25 years of academic experience, his research interests are in medical image processing, artificial intelligence, data science and cybersecurity, with over 180 publications in books, peer-reviewed journals and international conference proceedings. He actively serves as a speaker at many international conferences, as well as volunteers as a judge in STEM and innovation competitions for schools at the local and international levels. He may be contacted at email: logeswaran.nr@help.edu.my.

	**Kasthuri Subaramaniam** is a Senior Lecturer at the Department of Decision Science, Faculty of Business and Economics, University of Malaya. Her research interests include human-computer interaction, human personality types, augmented reality, artificial intelligence and cybersecurity. She actively serves as a reviewer for refereed journals. She may be contacted at email: s_kasthuri@um.edu.my.
	**Oras Baker** is an Associate Professor and Head of Masters in Cyber Security and Cyber Security Management at University of Ravensbourne London, UK. With 25 years of distinguished experience spanning academia, research, and industry, he specialises in Artificial Intelligence, Software Engineering, Cyber Security, Data Mining, and Machine Learning. He can be contacted at email: O.Alhassani@rave.ac.uk.
	**Bui Ngoc Dung** is a Senior Lecturer in Information Technology at the University of Transport and Communications, Vietnam. He holds a Ph.D. in Informatics from Malaysia University of Science and Technology and has completed postdoctoral research at TU Wien, Austria. His research focuses on machine learning, computer vision, biomedical applications, and structural health monitoring. Dr. Dung also serves as a reviewer, technical committee member, and keynote speaker at international conferences.