AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2) (2026)
www.ajmimc.com

A Privacy-Centric Edge Intelligence Paradigm for Instantaneous Clinical Insight Generation via Secure Data Transformation

S. Krishna Reddy[1], Solapure Dhanusha[1], Mende Poojitha[1], Kothwal Nithish Ram[1], Vutukuri Varun Kumar[1]

[1]Department of Computer Science and Engineering, [1]Sree Dattha Institute of Engineering and Science, Nagarjuna Sagar Road, Sheriguda, Ibrahimpatnam, Rangareddy Dist, 501510, India.

Abstract

The rapid integration of digital technologies in healthcare has intensified concerns surrounding data security and patient privacy, especially in predictive diagnostic systems. Conventional machine learning approaches typically rely on direct access to raw medical data, making them vulnerable to data breaches, unauthorized usage, and privacy compromises. Additionally, many existing solutions lack comprehensive authentication protocols and fail to safeguard sensitive information during both training and inference stages. These limitations highlight the necessity for intelligent frameworks that can deliver high predictive performance while maintaining strict data confidentiality and secure system access. To overcome these issues, this study introduces a secure and privacy-aware healthcare prediction framework that combines machine learning with encryption and authentication mechanisms. The system is implemented using the Flask web framework and integrates an email-based One-Time Password (OTP) verification process via Simple Mail Transfer Protocol (SMTP) to ensure secure user authentication. To protect sensitive medical data, a Lightweight Privacy-Preserving Machine Learning Encryption (LPME) approach is employed, which transforms input features using polynomial-based cryptographic techniques before they are utilized for model training. The framework is evaluated on healthcare datasets related to heart disease and hypothyroidism. Prior to model development, the data undergoes preprocessing steps such as normalization, categorical encoding, and class imbalance handling using the Synthetic Minority Oversampling Technique (SMOTE). For predictive modelling, Extreme Gradient Boosting (XGB) is adopted as the primary classifier due to its effectiveness in modelling complex relationships, while Gaussian Naive Bayes (GNB) is used as a baseline for comparison. The system's performance is assessed using standard evaluation metrics including accuracy, precision, recall, and F1-score.

Keywords:

1. Introduction

The advancement of healthcare infrastructure plays a crucial role in the socio-economic development of any nation, especially in the face of rising incidences of infectious and chronic diseases that continue to impact global populations [1]. In recent years, the emergence of Internet of Things (IoT)-based healthcare systems, commonly referred to as smart healthcare, has transformed patient monitoring by enabling continuous, real-time observation without requiring frequent hospital visits [2]. These systems, supported by wearable sensors and remote monitoring devices, have significantly improved accessibility and efficiency in healthcare delivery. However, the widespread adoption of such technologies has also introduced serious concerns related to data security, including potential breaches of confidentiality and integrity of sensitive patient information [3]. In addition to security challenges, conventional cloud-centric healthcare analytics often suffer from high latency, which can hinder timely medical intervention in critical situations. The delay caused by transmitting patient data to distant cloud servers and receiving analytical feedback can be detrimental in emergency care scenarios where rapid decision-making is essential [4]. To address both privacy risks and latency limitations, this study proposes an Edge-Enabled Diagnostic Framework integrated with Homomorphic Encryption (HE).

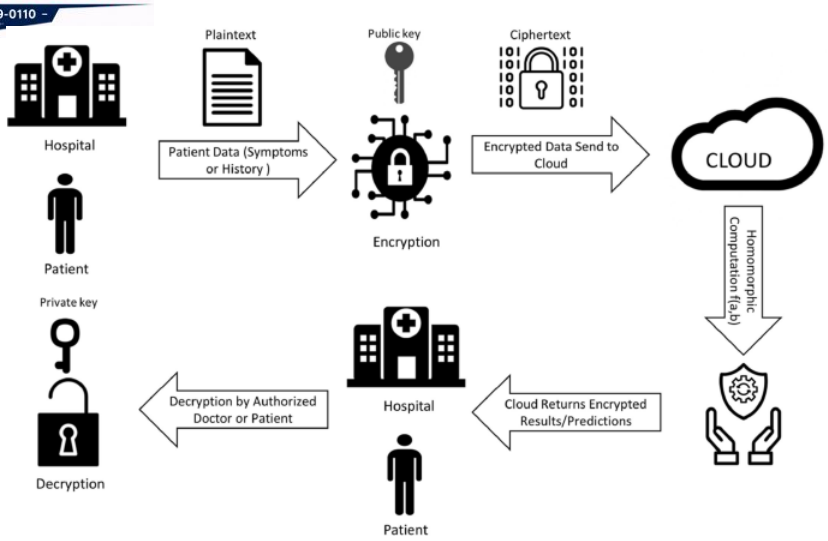


Figure. 1: Edge-enabled diagnostic engine.

By shifting computational processes closer to the data source through edge computing, the system significantly reduces response time and enhances real-time decision-making capabilities [5]. Simultaneously, homomorphic encryption enables secure processing by allowing computations to be performed directly on encrypted data, ensuring that sensitive health information remains protected even during analysis. This approach eliminates the need for decryption at intermediate stages, thereby strengthening privacy preservation, as shown in figure 1. The proposed system is designed to support real-time prediction of critical health conditions using data collected from remote patient monitoring devices. It also aims to meet regulatory requirements such as HIPAA compliance while maintaining low-latency diagnostics. The central objective of this work is to explore how secure, encrypted data processing can be effectively combined with edge computing to deliver reliable and privacy-aware healthcare solutions for real-time medical intervention [6].

2. Literature Survey

Moshawrab et al. [7] presented the state-of-the-art smart wearables that can detect and monitor cardiovascular diseases (CVDs), the leading cause of death worldwide. They focus on the types, features, and applications of smart wearables, such as smartwatches, smart bands, smart rings, and smart glasses, that can measure various physiological signals, such as electrocardiogram (ECG), photoplethysmogram (PPG), blood pressure (BP), and heart rate (HR) signals. Robinson et al. [8] developed a system integrating smart IoT devices to allow access to patient data via the internet using cloud computing. By facilitating the acquisition and accumulation of data needed to monitor patient health, physicians were better able to solve their health problems in timely fashion.

Islam et al. [9] proposed a four-step architecture for e-Health systems comprised of devices, data aggregation and processing, data storage and data analysis. They described applications where the system would be useful, related technologies necessary to make it work and attendant benefits and limitations. Turjman et al. [10] surveyed the integration of healthcare systems with blockchain for addressing health data security, integrity, ownership, privacy, and access control. Begli et al. designed a secure IDS (SVM-IDS) for remote healthcare systems.

Dwivedi et al. [11] developed a peer-to-peer (P2P) strategy for linking distant medical sensors and equipment through the Internet. They came up with the notion of a better blockchain foundation for IoT devices. In a decentralized context, this suggested approach by these authors provides higher security for a healthcare system. Begli et al. [12] designed a secure IDS (SVM-IDS) for remote healthcare systems designed a Deep Neural Network (DNN)-based IDS in an IoMT network. The underlying framework used Principal Component Analysis (PCA) and the Grey-wolf optimization technique to perform feature extraction.

He et al. [13] designed a Deep Learning-based Intrusion Detection System that used a Stacked AE technique to secure healthcare systems. In addition, various ML techniques such as Support Vector Machine (SVM), GNB (NB), XGB, and k-NN were used for comparison. Rahmani et al. [14] proposed the concept of Fog computing in a healthcare IoT system as a smart e-Health system to implement a distributed intermediary layer of intelligence between sensor nodes and the Cloud. They capitalized on the ubiquity of existing healthcare systems to alleviate the burden on sensors, networks, and remote healthcare centers, thereby solving mobility, energy efficiency, scalability, and reliability issues.

3. Proposed Methodology

The proposed methodology establishes a structured analytical framework for secure healthcare data analysis using advanced machine learning techniques integrated with privacy-preserving mechanisms. The analytical pipeline begins with dataset acquisition and organization, followed by data preprocessing and transformation. Sensitive medical data is protected through an encryption-based mechanism before being utilized for model training, ensuring confidentiality throughout the analytical process. The transformed data is then normalized and prepared for feature extraction, enabling effective learning by classification models. Multiple machine learning algorithms are applied to identify patterns related to medical conditions, allowing accurate prediction outcomes. A user interaction layer facilitates data handling, model execution, and prediction tasks, as illustrated in figure 2. A lightweight storage mechanism manages user authentication and trained models, while a server component enables remote data processing and prediction. Continuous evaluation and retraining further enhance analytical performance and adaptability to new healthcare data.

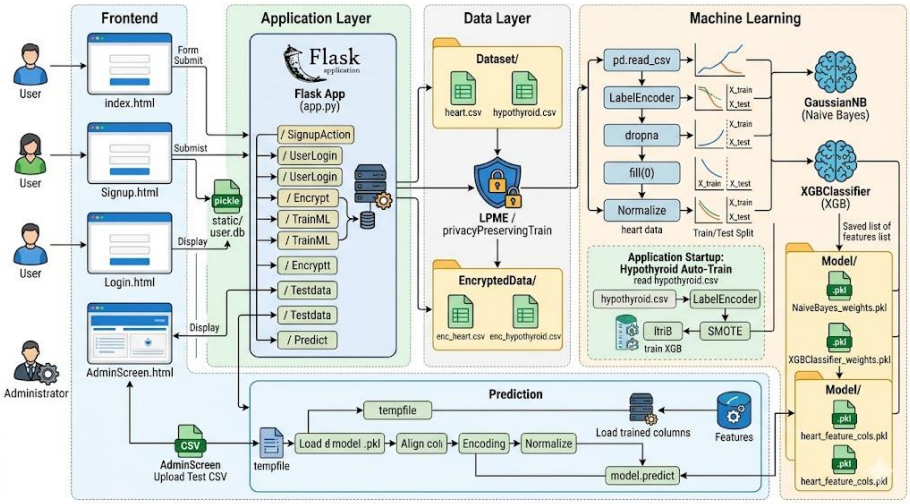


Figure. 2: Proposed system architecture

User Interface (Client Application)

- The system provides a user-friendly interface that enables interaction through a web-based environment.
- It allows operations such as user registration, login, dataset selection, encryption execution, model training, and prediction.
- Users can upload test datasets from local storage and initiate prediction tasks.
- All user inputs are captured through the interface and forwarded to backend modules for processing.

Flask Application Server

- The Flask server acts as the central control unit of the system.
- It manages communication between the user interface, data processing modules, and machine learning components.

- The server handles requests related to dataset encryption, model training, and prediction workflows.
- It enables remote access, allowing users to interact with the analytical framework from different locations.

User Database (Authentication Storage)

- A lightweight storage mechanism is used to maintain user authentication details.
- It stores information such as usernames, passwords, and user-specific details.
- The database supports user registration and login verification processes.
- It ensures quick access and efficient handling of authentication-related operations.

Healthcare Dataset (Input Data Source)

- The dataset serves as the primary input for the analytical framework.
- It includes structured medical data related to specific health conditions.
- The data contains various attributes representing patient information and diagnostic indicators.
- This dataset is used for both training machine learning models and performing prediction tasks.

Data Encryption and Privacy Preservation Module

- A dedicated privacy-preserving mechanism is applied to protect sensitive healthcare data.
- The module transforms raw datasets into encrypted formats before model training.
- This ensures that machine learning models operate on secured data without exposing original information.
- The approach enhances data confidentiality while maintaining analytical utility.

Data Preprocessing and Normalization

- The input data undergoes preprocessing steps such as handling missing values and feature alignment.
- Numerical features are normalized to ensure consistent scaling across all attributes.
- Categorical features are encoded into numerical representations for compatibility with machine learning models.

Machine Learning Classification Models

- The processed data is analysed using multiple classification algorithms to detect medical conditions:
  - GNB: Utilizes probabilistic modelling for efficient classification.
  - XGB Classifier: Employs gradient boosting for high-performance predictive analysis.
- Each model is trained and evaluated independently to compare performance.
- The trained models are stored for future prediction tasks.

Prediction and Output Generation

- The system processes new input data through trained models to generate predictions.
- It identifies the likelihood of specific medical conditions based on input features.
- Results are displayed in a structured format, including row-wise predictions and interpretation messages.
- The output enables users to understand classification outcomes effectively.

Remote Prediction Workflow

- The architecture supports remote data submission and prediction through a client–server model.
- Users can upload datasets from different systems to the server for analysis.
- The server processes the data using trained models and returns prediction results.
- This enables scalable and accessible healthcare analytics.

Model Evaluation and Performance Analysis

- The system evaluates model performance using metrics such as accuracy, precision, recall, and F1-score.
- Confusion matrices are used to analyse classification effectiveness.

- Comparative analysis of multiple models helps in selecting the best-performing approach.
- Continuous evaluation ensures reliability and robustness of predictions.

Adaptive Learning and Retraining

- The framework supports retraining of models when new data becomes available.
- This allows the system to adapt to changing data patterns and improve accuracy over time.
- Updated models are stored and reused for future predictions.
- The continuous learning process enhances long-term system effectiveness.

3.1 LPME

The LPME technique is a custom-designed encryption approach intended to safeguard sensitive medical information while still allowing meaningful machine learning operations on transformed data. Traditional encryption methods protect confidentiality but make it impossible to directly use encrypted data for analytics or model training. LPME addresses this challenge by converting raw numerical values into structured encrypted polynomial coefficients that preserve essential mathematical relationships required for downstream machine learning. The workflow of LPME is illustrated in Figure 3. This enables the system to protect patient confidentiality without fully sacrificing the utility of the dataset. LPME operates on the principle of polynomial-based public key encryption, inspired by lattice and Learning-With-Errors (LWE) cryptographic constructions. It embeds each plaintext feature value into a polynomial representation and introduces controlled randomness and noise using secret and public keys.

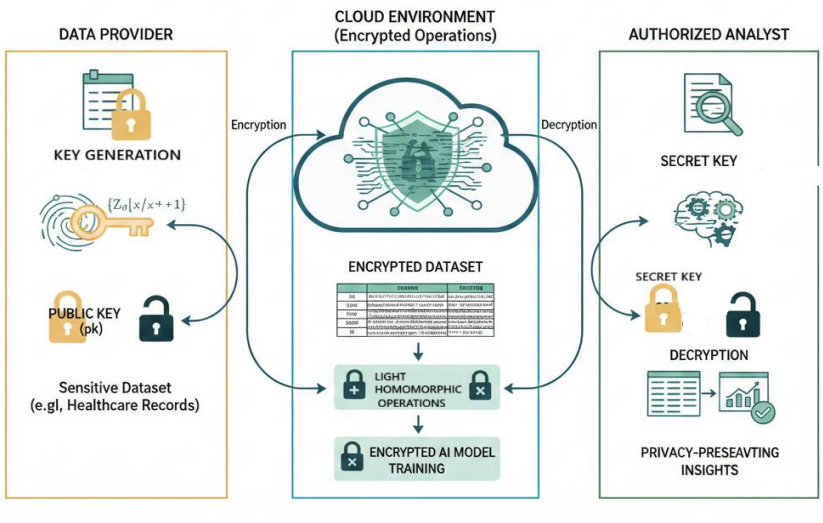


Figure. 3: LPME workflow

This ensures that even if an attacker gains access to the encrypted dataset, the original values cannot be recovered without the secret key. The encryption process produces ciphertext values that appear random but still maintain certain algebraic properties, allowing limited plaintext operations such as scaled addition or multiplication when necessary. In the context of this research, LPME is applied to all numerical features in medical datasets such as heart disease and hypothyroid data. Each feature value is transformed independently, ensuring strong protection for every record.

Dataset Loading: The workflow begins by loading the original medical dataset from a CSV file. The system reads each row and extracts all numerical features that require encryption. This ensures the raw data is properly structured before any transformation. Missing values in the dataset are replaced to maintain consistency during encryption. The entire dataset is then converted to a NumPy array for efficient processing.

Normalization of Feature Values: Each feature value is passed through the normalize value() function, which converts all inputs into integer form. Categorical symbols like ‘T’, ‘F’, ‘?’, or empty values are

mapped to numeric representations. This guarantees the encryption process receives clean, valid integers. Normalization also resolves inconsistent data formatting commonly found in medical datasets.

Key Generation: Before encryption begins, the system generates a public key and a secret key using polynomial-based key generation. The public key consists of two polynomials derived from uniform and noise distributions, while the secret key is a binary polynomial. These keys form the mathematical foundation for secure polynomial encryption. The keys ensure that encrypted values cannot be reversed without the corresponding secret key.

Plaintext Polynomial Encoding: Each normalized integer value is encoded into a polynomial format, where the first coefficient represents the actual message and the remaining coefficients are zeros. This conversion embeds the plaintext into a structure compatible with polynomial cryptography. Encoding as a polynomial also enables homomorphic-style operations. This step prepares the plaintext for integration with randomness and noise.

Message Scaling Using Modulus: The encoded polynomial message is scaled by computing delta = q/t, which maps the plaintext space into the ciphertext modulus space. Scaling ensures message components remain distinguishable after encryption noise is added. This safeguards the message from overflow or loss during polynomial operations. It also standardizes message representation across all features.

Randomness and Noise Generation: Three types of noise components are generated: a binary vector u and two error polynomials e1 and e2. These random elements strengthen encryption security by adding unpredictability to ciphertexts. Even identical plaintext values produce different encrypted outputs due to this randomness. Noise also prevents adversaries from performing statistical attacks.

Polynomial Ciphertext Computation: The encryption function computes ciphertexts (ct0, ct1) using polynomial multiplication and addition with the public key. ct0 combines the message, noise, and part of the public key, while ct1 retains the second half of the key structure. The result is a secure encrypted representation of the original value. These ciphertexts follow the principles of lattice/LWE-style encryption.

Encrypted Value Extraction and Storage: From each ciphertext, the system extracts the first coefficient from ct0, which becomes the encrypted value stored in the dataset. This coefficient is a large integer that hides the plaintext while preserving its analytic structure. The encrypted value is then written into a new CSV file. The process is repeated for all features in each dataset row.

Final Encrypted Dataset Output: After all rows and features are processed, the system exports a fully encrypted dataset ready for privacy-preserving machine learning. The output file contains only encrypted numerical values, protecting sensitive patient information. This ensures downstream ML models operate on data that never exposes raw medical attributes. The encrypted dataset maintains compatibility with normalization and training workflows.

4. Implementation description

This project demonstrates a complete end-to-end machine learning pipeline integrated with privacy-preserving encryption and a Flask-based interactive web application. The system combines secure data handling, preprocessing, SMOTE balancing, multiple machine learning classifiers, and real-time prediction through a user-friendly interface. The implementation focuses strongly on encrypted data processing, model training, evaluation, and prediction execution. Below is the detailed explanation of key components and their workflow:

1. Library Imports and System Setup

- Imports essential Python libraries required for dataset handling (pandas, NumPy), data normalization, and preprocessing (Label Encoder, normalize).
- Includes machine learning packages such as scikit-learn (Gaussian NB, train_test_split), XGB for advanced gradient boosting, and imb learn for SMOTE-based data balancing.

- Utilizes Flask to build a web interface for user interactions such as login, dataset upload, encryption, training, and prediction.
- Loads custom privacy-preserving encryption modules (privacy Preserving Train, privacy Preserving Test) from the LPME framework.
- Creates folders for saving encrypted datasets, trained models, and feature lists using joblib and pickle.

2. Dataset Loading and Initial Processing

- Loads raw medical datasets (heart disease and Hypothyroid datasets) from CSV files.
- Handles initial cleaning, such as filling missing values, removing duplicates, and dropping unnecessary columns.
- Uses Label Encoding to convert categorical attributes into numeric format for ML compatibility.
- Normalizes numerical features to maintain uniform scale before training.
- Reads user-uploaded test files using Flask's secure file handler and temporarily stores them for processing.

3. Data Encryption Using LPME

- Implements a privacy-preserving encryption function that transforms raw datasets into encrypted numerical data while preserving machine learning usability.
- Converts selected datasets (heart, hypothyroid) into encrypted versions and stores them under Encrypted Data.
- Displays a comparison of plaintext and encrypted dataset samples via HTML tables for user verification.
- Ensures no sensitive personal or medical information is exposed during training or prediction.

4. Data Preprocessing Pipeline

- Performs structured preprocessing steps such as:
  - Handling missing values across all columns.
  - Removing duplicate records to avoid redundancy.
  - Label encoding categorical variables.
  - Normalizing numerical features for consistent range.
- For hypothyroid data, applies SMOTE to balance minority classes and prevent biased model training.
- Extracts and stores feature lists from the training dataset to ensure test data alignment during inference.

5. Machine Learning Model Training

- Trains multiple classifiers including:
  - **Gaussian GNB** — a probabilistic model for disease classification.
  - **XGB Classifier** — a high-performance gradient boosting algorithm.
- Splits encrypted datasets into training and testing sets.
- Evaluates each model using metrics such as accuracy, precision, recall, F1-score, and specificity.
- Saves trained model weights and their corresponding feature columns for future predictions.
- Provides a comparison table showing the performance of all trained models on encrypted data.

6. Model Evaluation and Metrics Analysis

- Calculates essential performance metrics including:
  - Accuracy
  - Precision
  - Recall
  - Specificity

    - F1-score
- Generates classification reports for deeper performance insight.
- Displays results inside Flask's Admin Panel for easy visualization.
- Allows users to compare models and select the best-performing one for prediction.

7. Prediction on New Test Data

- Users upload a new medical test file through the Flask interface.
- The system preprocesses the test data using the same pipeline applied to training sets:
    - Handling missing features
    - Adding absent columns
    - Normalizing values
    - Encoding categorical values
- Loads the appropriate pre-trained model and feature list based on the selected dataset.
- Generates predictions for each record and maps predicted class IDs back to disease labels.
- Displays detailed row-wise prediction messages (e.g., "Heart Disease Detected", "Thyroid Not Detected") in the Admin Panel.

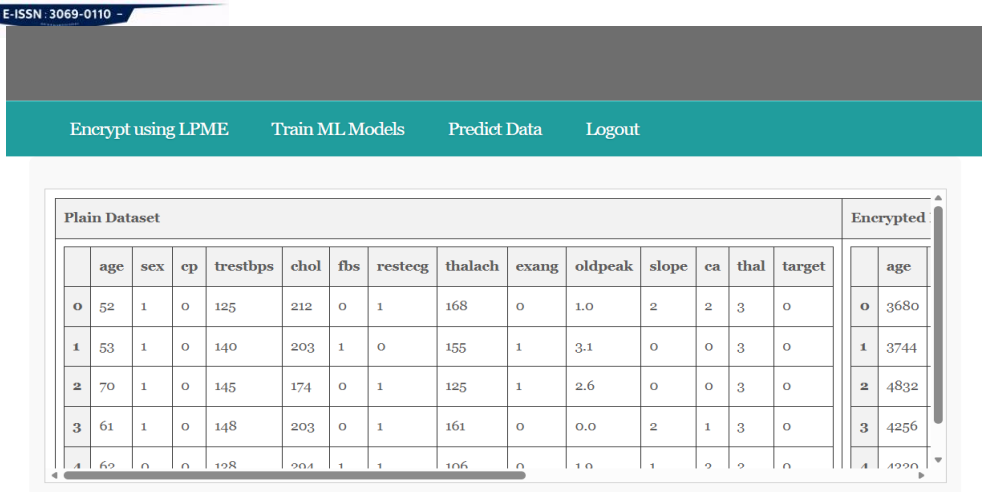
8. Web Interface Integration with Flask

- Implements routes for signup, login, dataset selection, encryption, training, and prediction.
- Uses HTML templates to render dataset tables, results, model metrics, and user notifications.
- Manages user sessions, navigation, input validation, and error handling.
- Ensures smooth interaction between UI and backend operations using Flask request–response lifecycle.

4. 1 Results description

The results of the study demonstrate the effectiveness of integrating privacy-preserving techniques with machine learning models for secure healthcare data analysis. The experimental evaluation was conducted on encrypted datasets to ensure that data confidentiality was maintained throughout the process. Multiple classification algorithms were assessed based on performance metrics such as accuracy, recall, precision, and F1-score. Among the evaluated models, the gradient boosting–based approach showed superior predictive capability, while probabilistic methods provided stable baseline performance. The use of data normalization and imbalance handling techniques contributed to improved model generalization and reliability. Confusion matrix analysis further validated the consistency of classification outcomes across different classes. The results also confirm that encryption does not significantly degrade model performance, highlighting the feasibility of secure learning.

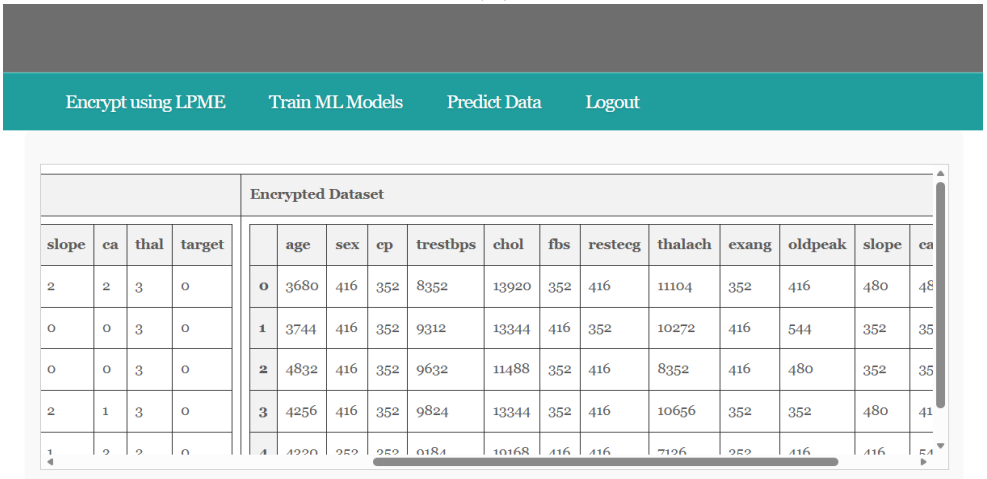
Figure 4 depicts the comparative visualization of the plain heart disease dataset and its encrypted counterpart. Part (A) illustrates the original medical dataset containing clinical attributes used for diagnostic analysis, while Part (B) shows the transformed encrypted representation of the same data. This comparison highlights how sensitive medical information is secured without losing structural integrity required for analysis. The figure emphasizes the effectiveness of encryption in protecting data confidentiality. It also demonstrates the system's capability to operate on encrypted medical datasets.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0

	age
0	3680
1	3744
2	4832
3	4256

(A)



slope	ca	thal	target
2	2	3	0
0	0	3	0
0	0	3	0
2	1	3	0

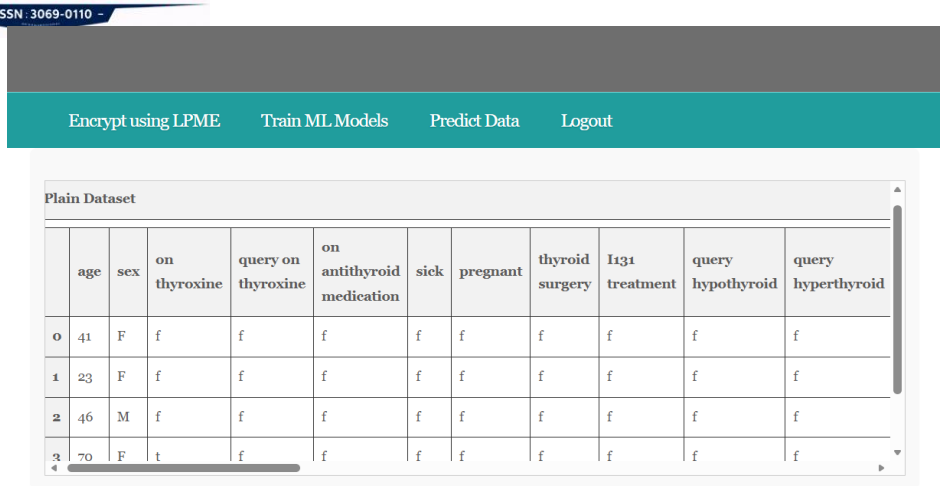
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	3680	416	352	8352	13920	352	416	11104	352	416	480	48
1	3744	416	352	9312	13344	416	352	10272	416	544	352	35
2	4832	416	352	9632	11488	352	416	8352	416	480	352	35
3	4256	416	352	9824	13344	352	416	10656	352	352	480	41

(B)

Figure. 4: (A) Plain and (B) Encrypted Heart Dataset Display Screen

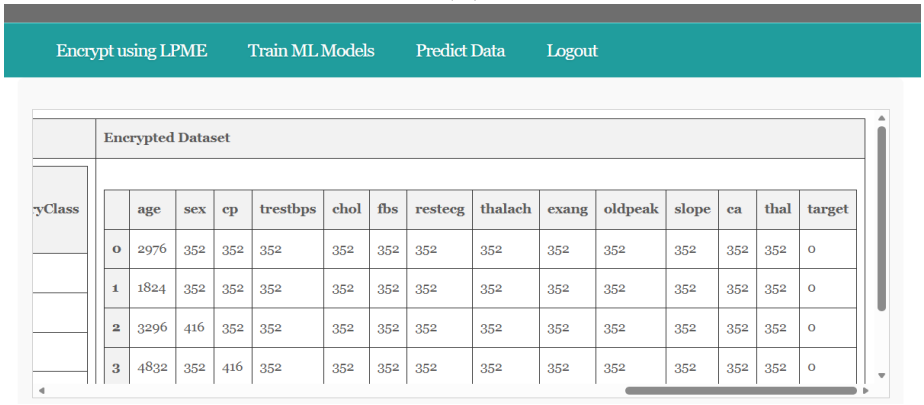
Figure 5 depicts the output visualization of the hypothyroid dataset in both plain and encrypted formats. Part (A) represents the original hypothyroid medical records containing diagnostic features, while Part (B) illustrates the encrypted dataset generated for secure processing. This figure highlights the system's ability to handle multiple medical datasets using a unified encryption and analytics framework. The transformation preserves analytical usability while ensuring data security. It reinforces the applicability of the proposed system across diverse medical diagnostic scenarios.

Figure 6 depicts the prediction result display screen generated after processing the uploaded test data. The figure shows the extracted feature values used by the trained machine learning model for medical condition assessment. It represents the system's ability to interpret patient-specific input attributes and produce diagnostic outcomes. This screen demonstrates real-time inference capability within the edge-enabled architecture. It confirms the integration of secure data handling and intelligent prediction in the proposed system.



	age	sex	on thyroxine	query on thyroxine	on antithyroid medication	sick	pregnant	thyroid surgery	I131 treatment	query hypothyroid	query hyperthyroid
0	41	F	f	f	f	f	f	f	f	f	f
1	23	F	f	f	f	f	f	f	f	f	f
2	46	M	f	f	f	f	f	f	f	f	f
3	70	F	t	f	f	f	f	f	f	f	f

(A)



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	2976	352	352	352	352	352	352	352	352	352	352	352	352	0
1	1824	352	352	352	352	352	352	352	352	352	352	352	352	0
2	3296	416	352	352	352	352	352	352	352	352	352	352	352	0
3	4832	352	416	352	352	352	352	352	352	352	352	352	352	0

(B)

Figure. 5: (A) Plain and (B) Encrypted Hypothyroid Dataset Output Screen

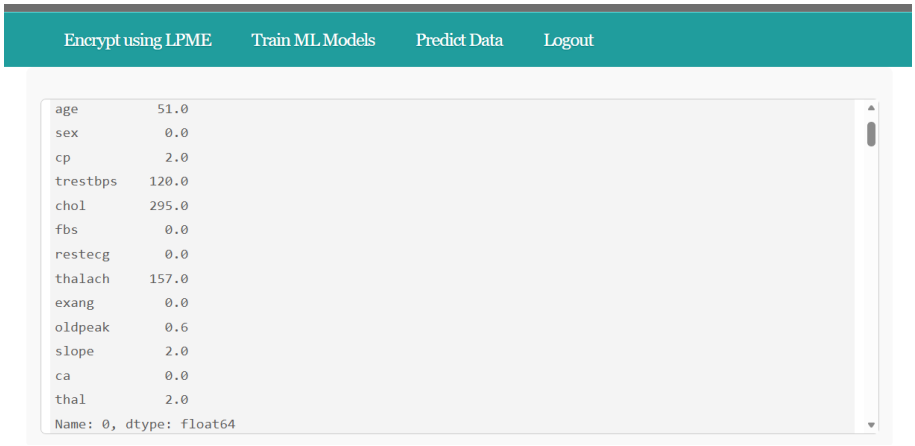


Figure. 6: Prediction Result Display Screen

The comparison table 1 presents the performance of two machine learning algorithms GNB and XGB evaluated on the encrypted heart disease dataset. The metrics considered include Accuracy, Recall, and Specificity, which collectively measure how effectively each model distinguishes between heart disease and non–heart disease cases. GNB performs reasonably well with an accuracy of 81.95%, demonstrating its capability to classify most instances correctly. However, the recall value of 81.77% indicates that some positive cases are still misclassified. XGB, on the other hand, significantly outperforms GNB by achieving a perfect score of 100% across all three metrics. This indicates that XGB correctly identifies all positive and negative cases without any misclassifications. The results clearly show that XGB is the most robust and reliable model for this dataset, making it the preferred choice for accurate heart disease prediction within the proposed system.

Table 1: Comparison of algorithm performance

Algorithm	Accuracy (%)	Recall (%)	Specificity (%)
GNB	81.95	81.77	85.98
XGB	100.00	100.00	100.00

5. Conclusion

The study demonstrates that integrating privacy-preserving mechanisms with machine learning techniques offers a reliable and secure approach for healthcare data analysis. By safeguarding sensitive medical information through encryption while enabling accurate prediction of conditions such as heart disease and thyroid disorders, it effectively addresses key challenges related to data security and patient confidentiality. The incorporation of advanced learning models, along with systematic preprocessing steps like normalization, encoding, and class imbalance handling using SMOTE, enhances the robustness and consistency of predictions. The evaluation outcomes indicate that the framework maintains high predictive performance even when operating on protected data, confirming that privacy measures do not significantly compromise analytical accuracy. This highlights the practicality of deploying secure machine learning solutions in sensitive domains. Furthermore, the approach supports early diagnosis and informed clinical decision-making, contributing to more efficient and dependable healthcare practices. The study presents a comprehensive and scalable solution that successfully balances data privacy with intelligent analytics, making it suitable for real-world medical applications where both security and performance are essential.

References

[1] Rancea, A.; Anghel, I.; Cioara, T. Edge Computing in Healthcare: Innovations, Opportunities, and Challenges. Future Internet 2024, 16, 329. https://doi.org/10.3390/fi16090329

[2] Rauniyar, A.; Hagos, D.H.; Jha, D.; Håkegård, J.E.; Bagci, U.; Rawat, D.B.; Vlassov, V. Federated Learning for Medical Applications: A Taxonomy, Current Trends, Challenges, and Future Research Directions. IEEE Internet Things J. 2024, 11, 7374–7398.

[3] Rahman MA, Shahrior MF, Iqbal K, Abushaiba AA. Enabling Intelligent Industrial Automation: A Review of Machine Learning Applications with Digital Twin and Edge AI Integration. Automation. 2025; 6(3):37. https://doi.org/10.3390/automation6030037

[4] Mehrabi, A.; Yari, K.; Van Driel, W.D.; Poelma, R.H. AI-Driven Digital Twin for Health Monitoring of Wide Band Gap Power Semiconductors. In Proceedings of the 2024 IEEE 10th Electronics System-Integration Technology Conference (ESTC), Berlin, Germany, 11–13 September 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8.

[5] Ma, Q.; Niu, J.; Ouyang, Z.; Li, M.; Ren, T.; Li, Q. Edge Computing-Based 3D Pose Estimation and Calibration for Robot Arms. In Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 22–24 August 2020; pp. 246–251.

[6] Schmitt, J.; Bönig, J.; Borggräfe, T.; Beitinger, G.; Deuse, J. Predictive Model-Based Quality Inspection Using Machine Learning and Edge Cloud Computing. Adv. Eng. Inform. 2020, 45, 101101.

[7] Ali, A.; Pasha, M.F.; Ali, J.; Fang, O.H.; Masud, M.; Jurcut, A.D.; Alzain, M.A. Deep Learning Based Homomorphic Secure Search-Able Encryption for Keyword Search in Blockchain Healthcare System: A Novel Approach to Cryptography. Sensors 2022, 22, 528. https://doi.org/10.3390/s22020528

[8] Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy-preserving healthcare blockchain for IoT. Sensors 2019, 19, 326.

[9] Hang, L.; Kim, D.-H. Design and implementation of an integrated iot blockchain platform for sensing data integrity. Sensors 2019, 19, 2228.

[10] Jia, X.; Hu, N.; Su, S.; Yin, S.; Zhao, Y.; Cheng, X.; Zhang, C. IRBA: An identity-based cross-domain authentication scheme for the internet of things. J. Electron. 2020, 9, 634.

[11] Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Gener. Comput. Syst. 2018, 78, 641–658.

[12] Kim, D.; Yang, H.; Chung, M.; Cho, S.; Kim, H.; Kim, M.; Kim, K.; Kim, E. Squeezed Convolutional Variational AutoEncoder for Unsupervised Anomaly Detection in Edge Device Industrial Internet of Things. In Proceedings of the 2018 International Conference in Information and Communication Technologies (ICICT), DeKalb, IL, USA, 23–25 March 2018; pp. 67–71.

[13] Robinson, Y.H.; Presskila, X.A.; Lawrence, T.S. Utilization of Internet of Things in Health Care Information System. In Internet of Things and Big Data Applications. Intelligent Systems Reference Library; Balas, V., Solanki, V., Kumar, R., Eds.; Springer: Cham, Switzerland, 2020; Volume 180, pp. 35–46.

[14] Islam, M.S.; Humaira, F.; Nur, F.N. Healthcare Applications in IoT. Global. J. Med Res. B Pharma Drug Discov. Toxicol. Med. 2020, 20, 1–3.