Journal of Science and Technology
ISSN: 2456-5660 Volume 9, Issue 1 (JAN -2024)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp1-10*

EFFICIENT PRIVACY PRESERVING MEDICAL DIAGNOSIS ON EDGE COMPUTING PLATFORMS

Dr. Jayrajan[1], T. Chandrika[2], T. Sai Shritha [2], Y. Sangeetha [2]

[1]Professor, [2]UG Student, [1,2]Department of Computer Science Engineering
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India

To Cite this Article

Dr. Jayrajan, T. Chandrika, T. Sai Shritha , Y. Sangeetha**, "EFFICIENT PRIVACY PRESERVING MEDICAL DIAGNOSIS ON EDGE COMPUTING PLATFORMS "** *Journal of Science and Technology, Vol. 09, Issue 01 - JAN 2024, pp01-10*

Article Info

Received: 25-12-2023 Revised: 05 -01-2024 Accepted: 15-01-2024 Published: 25-01-2024

ABSTRACT

The advent of edge computing has revolutionized various industries, and healthcare is no exception. Edge computing involves processing data closer to its source, enabling real-time analytics and decision-making. In the healthcare sector, the integration of edge computing offers advantages such as reduced latency and enhanced privacy protection. The conventional system for medical diagnosis often involves the centralization of sensitive patient data in cloud-based platforms. While this facilitates data storage and processing, it raises concerns about data privacy and security. The transmission of medical data to centralized servers introduces latency, potentially hindering real-time decision-making, a critical aspect in healthcare. Furthermore, the conventional system may not be optimized for resource-constrained environments, leading to inefficiencies in computation and increased processing times. Data breaches in centralized systems pose a significant risk to patient privacy, and the potential consequences, such as identity theft and discrimination, underscore the need for a more secure and privacy-preserving approach to medical diagnosis.The proposed system introduces a paradigm shift in medical diagnosis by leveraging edge computing and the Extra Tree Classifier algorithm. Extra Tree Classifier is chosen for its efficiency and accuracy in handling medical data. The key focus of the proposed system is to ensure privacy throughout the diagnosis process. This involves developing machine learning models that can make accurate predictions while preserving the confidentiality of patient records. The system operates efficiently within the resource-constrained environment of edge computing platforms, addressing the drawbacks of the conventional system. Real-time processing is prioritized to cater to healthcare conditions that require immediate attention. The proposed system not only provides timely results but also maintains a high level of accuracy and reliability, instilling trust in healthcare providers and ensuring that patients receive optimal care. This research aligns with the principles of patient-centric care, allowing patients to have more control over their data, share it securely with healthcare providers, and receive real-time decision support while preserving the privacy and integrity of their medical information.

Keywords: Edge Computing platforms, Medical Diagnosis, Privacy Preserving.

1. INTRODUCTION

A lightweight privacy-preserving security protocol for heart disease prediction in an edge computing environment is a crucial innovation that addresses both the healthcare and security challenges of modern digital ecosystems. In this context, edge computing refers to the decentralized processing of data at or near the data source, which is particularly relevant in healthcare settings to ensure real-time processing and minimize latency.The protocol is designed to efficiently predict heart disease risk factors and diagnose potential cardiac issues in individuals while maintaining the utmost privacy and security of sensitive medical data. It leverages edge computing capabilities to process medical data at the source, such as wearable devices or sensors, reducing the need for data transmission to centralized servers and minimizing the risk of data breaches during transit.Privacy preservation is achieved through advanced cryptographic techniques, including homomorphic encryption and secure multiparty computation. These methods allow computations to be performed on encrypted data without revealing the raw medical information, ensuring that the privacy of patients is upheld while still providing accurate predictions and diagnoses.

Additionally, the lightweight nature of the protocol ensures that it can be efficiently implemented on resource-constrained edge devices, which is essential for practical deployment in healthcare settings. This protocol also includes mechanisms for secure authentication and access control to ensure that only authorized healthcare professionals can access the decrypted data for diagnosis and treatment.

2. LITERATURE SURVEY

The Internet of Things (IoT)-based applications and services include sensor networks, healthcare systems, transportation, smart industry, communication systems, smart cities, and manufacturing [1]. The Industrial Internet of Things (IIoT) has been proposed to dramatically enhance qualities of traditional industries, break regional limitations to achieve remote monitoring, perform autonomous production, and provide real-time information to users [2,3,4]. The Internet of Thing (IoT) will deliver about 85% of all IoT devices in healthcare by 2025 [1]. According to Tractia, an intelligent organization, annual earnings in this sector using blockchain technologies would reach USD 9 billion by 2025 [2]. IoT devices are widely used in healthcare to give real-time services to patients and physicians [3]. IoMT-based medical device applications include medical institutions and businesses. However, as the number of internet-connected medical devices (IoMT) increases, greater volumes and inconsistency of data will be generated. With centralized cloud-based characteristics, handling significant data traffic in IoT (IoMT) has now become a severe problem and reason for concern [4]. As a result, patient safety and confidentiality concerns have grown while data collection, data ownership, location privacy, etc., will be at risk. By copying data and changing the identification of healthcare equipment, intruders and hackers can easily target the 5G-enabled IoMT network. IoMT-Cloud currently has a single point of failure, malicious attacks, and privacy leaks, as shown in Figure 1. To ensure network security and secure PHR transmission, data transfer between IoMT and Cloud requires trust, device identification, and user authentication (UA). With the traditional Central Cloud service, however, due to the round-the-clock networking of nodes in this IoT network, it is vulnerable to various security issues, such as message tampering, eavesdropping, and denial-of-service attacks [5]. In the industrial industry, this raises major security issues as the misuse of data can result in the incorrect diagnosis and can cause life-threatening scenarios for the patients under observation [6,7].

The edge computing based IoMT is currently a popular topic. Previous research missed important security issues such as: 1. Healthcare IoMT devices send data to cloud servers that are frequently unencrypted and open to manipulation and attack. As a result, sensitive patient information will likely be accessible. This issue leads to security vulnerabilities. 2. To our knowledge, the need to identify

IoMT medical devices, which leads to the verification and authentication of health data, is considered very important and sensitive, and it can be accomplished quickly using a blockchain in the FC-IoMT system. Moreover, servers at the network's edge should perform more detailed authentication and verification. BAKMP-IoMT, the new IoMT key agreement technique for blockchain-accessible authentication, was designed by [8]. It is also obtained theoretically from the algorithm's top time complexity and the number of patients.

Researcher in a study [9] explored various design research topics on readers' 5G-enabled tactile internet edge computing. In the same way [10] thoroughly examined 5G-assisted smart health of 14 care solutions in the IoT. R. Researchers in a study [11] proposed a multi-cloud cascade architecture, a low-overhead native testing framework, and a medical data storage backup method. This is also something that is examined by researchers [12] proposed a smart authentication (SSA) system to improve patient–physician data security and privacy preservation systems.

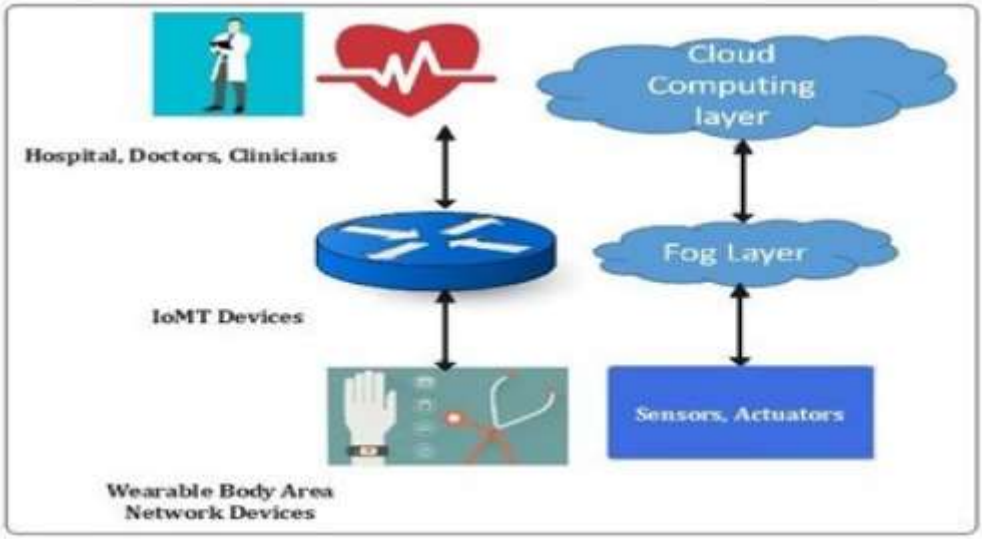


Figure 1. Application of edge computing.

3. PROPOSED SYSTEM

The proposed research work aims to revolutionize mobile-based health symptom prediction and disease diagnosis by addressing the critical issues of response time and patient data privacy. It leverages Edge Nodes to expedite response times and employs Light Weight Homomorphic Encryption to protect patient data throughout the process. This innovative approach, combining LPME and Extra Tree Classifier (ETC), ensures that healthcare applications can deliver timely and accurate disease predictions while upholding the highest standards of patient data security and privacy. Figure 4.1 shows the proposed system model.

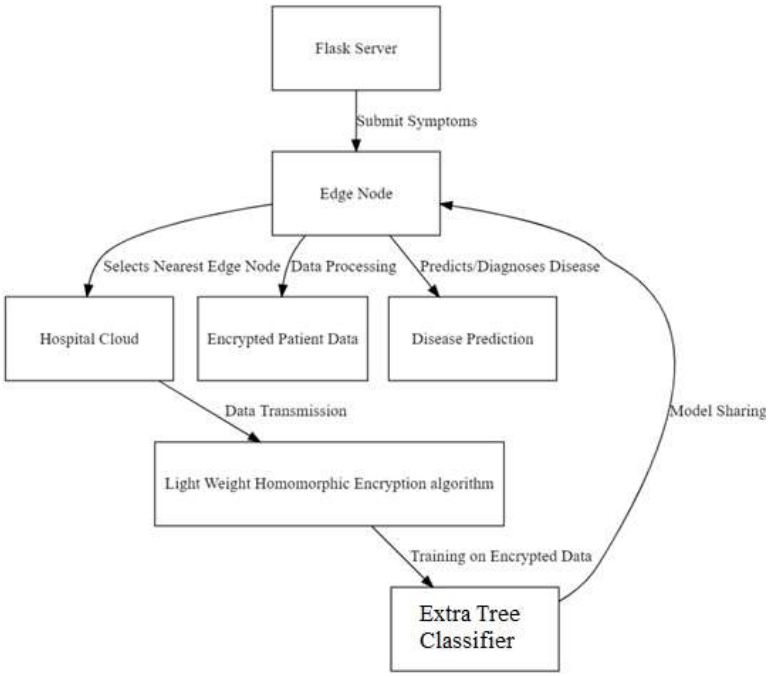


Figure.2: Proposed system model

The proposed research work addresses several key challenges in the context of mobile-based health symptom prediction and disease diagnosis while emphasizing patient privacy and minimizing response times. The research can be summarized as follows:

Step 1: Minimizing Response Time with Edge Computing:

- The research identifies a critical issue: the need for rapid response when a user submits health symptoms for disease prediction via a mobile application.
- To address this problem, the research introduces the concept of Edge Nodes. These Edge Nodes are strategically distributed and connected to the hospital cloud.
- When a user submits a request, the mobile app selects the nearest and available Edge Node for data processing, reducing response times significantly. This minimizes delays in diagnosing critical conditions like heart disease.

Step 2: Privacy-Preserving Data Processing:

- One major concern in using machine learning for disease prediction is the need for training data, which, if exposed, could breach patient privacy.
- To protect patient data, the research employs a Lightweight Homomorphic Encryption algorithm. This encryption method allows for computations to be performed directly on encrypted data without the need for decryption, maintaining data privacy.
- The training process for the disease prediction algorithm (E) is conducted on this encrypted dataset, ensuring that the patient's health information remains confidential.
- The trained model parameters are then shared among multiple Edge Nodes, allowing them to perform disease prediction on new patient data without ever accessing the unencrypted data.

Step 3: Privacy-Preserving Medical Diagnosis:

- The core of the proposed work lies in the Lightweight Privacy Preserving Medical Diagnosis (LPME) framework, which combines the use of homomorphic encryption and machine learning.
- LPME encrypts both patient training and test data, ensuring that sensitive health information remains secure during the entire process.
- The ETC classifier, trained on this encrypted data, can predict or diagnose diseases without the need for decryption, preserving patient privacy.

Light Weight Homomorphic Encryption algorithm

Lightweight Homomorphic Encryption (LWHE) is a cryptographic technique that plays a crucial role in preserving data privacy while allowing computations to be performed on encrypted data without the need for decryption. Unlike traditional homomorphic encryption schemes, LWHE is designed with a focus on efficiency and computational simplicity, making it particularly suitable for resource-constrained environments, such as mobile devices and edge computing platforms.

The key characteristic of LWHE is its ability to enable mathematical operations directly on ciphertexts, preserving the confidentiality of data throughout the computation process. This is achieved through specific algebraic properties and optimizations that reduce the computational overhead associated with homomorphic encryption. LWHE is typically applied to numerical data, making it well-suited for operations like addition and multiplication.

One of the main advantages of LWHE is its minimal computational burden when compared to more complex homomorphic encryption schemes. This efficiency is achieved by trading off some of the security guarantees provided by stronger encryption methods, such as Fully Homomorphic Encryption (FHE). LWHE allows for reasonably fast computations while still maintaining a reasonable level of security for many practical use cases.

LWHE has found applications in various domains, including secure data outsourcing, privacy-preserving machine learning, and secure cloud computing. For instance, in healthcare, LWHE can be used to perform computations on encrypted patient data, enabling medical research and disease prediction while ensuring that sensitive health information remains confidential.

Data Preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Step 1: Data Description: Generates summary statistics for all columns in the dataset. This step provides statistics such as count, mean, standard deviation, minimum, and maximum values for both numeric and categorical columns, giving an overview of the dataset's characteristics.

Step 2: Missing Value Analysis: Visualizes missing values using a matrix plot and displays the total count of missing values for each column. These steps help identify and visualize missing data in the dataset, which is essential for data cleaning and imputation.

Step 3: Data Types: Inspects the data types of columns and converts object-type columns to the category data type. Converting object-type columns to the category data type can reduce memory usage and improve analysis efficiency.

Step 4: Data Preprocessing: Preprocesses the dataset by extracting and transforming date components, removing '$' from ticker values, and converting data types. These steps prepare the data for analysis by making it more suitable for visualization and modeling.

Step 5: Feature Engineering: Removes the 'date' column from the dataset. Feature engineering involves selecting, transforming, or removing features to prepare the dataset for modeling.

Step 6: Final Data Check: Checks for missing values and inspects unique values in specific columns. These final checks ensure that the data is in a suitable state for further analysis and modeling.

Dataset Splitting

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

Extra Tree Classifier Model

Decision Tree Classifier Extremely Randomized Trees Classifier(Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees. To perform feature selection using the above forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of split (Gini Index if the Gini Index is used in the construction of the forest) is computed. This value is called the Gini Importance of the feature. To perform feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and the user selects the top k features according to his/her choice.

Operational Procedure for Extra Trees Classifier

Step 1. Ensemble Learning Concept: Ensemble learning involves combining the predictions of multiple models to improve overall performance. Extra Trees employs an ensemble of decision trees, each trained on a different subset of the data.

Step 2. Decision Trees in Extra Trees Standard Decision Trees: Extra Trees uses decision trees as base learners. The algorithm introduces extra randomness during tree construction, differing from traditional decision trees.

Step 3. Random Feature Selection: At each split in a tree, a random subset of features is considered for splitting. This adds diversity among the trees and increases robustness against overfitting.

Step 4. Bootstrapped Sampling: Extra Trees builds each tree using bootstrapped samples, i.e., random samples with replacement from the training data. This ensures diversity among the trees since each tree sees a different subset of the data.

Step 5. Decision Aggregation: For classification, predictions from all trees are combined through a voting mechanism. For regression tasks, the average of the predictions is considered.

Step 6. Hyperparameters Tuning: Number of Trees (n_estimators): The quantity of trees in the ensemble, impacting both training time and model performance. Maximum Depth: Limits the depth of each tree, influencing the model's capacity to capture complex patterns. Minimum Samples Split: The minimum number of samples required to split an internal node.

Step 7. Handling Overfitting: The inherent randomness in feature selection and sample bootstrapping aids in reducing overfitting. Aggregating predictions from multiple trees enhances generalization.

Step 8. Training Phase: Independent Training: Each tree in the ensemble is trained independently on a different subset of the data. Parallelization: Training trees can be parallelized, making Extra Trees efficient for large datasets.

Step 9. Prediction Phase: Combining Outputs: During prediction, outputs from all trees are combined, and the majority vote determines the final class (for classification tasks).

Step 10. Performance Evaluation: Test Set: Evaluate the model on a separate test set to assess its ability to generalize. Common metrics include accuracy, precision, recall, F1-score for classification, and mean squared error for regression.

Step 12. Fine-Tuning and Optimization: Grid Search: Use techniques like grid search to find the optimal hyperparameters for the specific dataset. Cross-Validation: Employ cross-validation to ensure robust model performance across different data subsets.

4. RESULT AND DISCUSSION

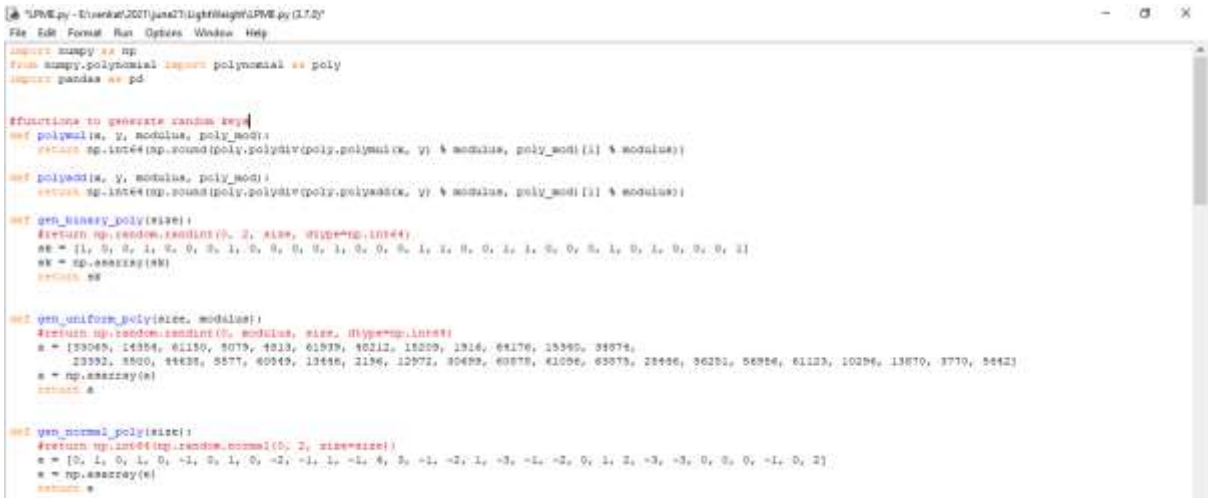
Now-a-days from mobile user can do anything and they can enter their health symptoms to mobile and mobile will send request to hospital cloud and this cloud will apply machine learning algorithm on user data to predict disease and while using such application many drawbacks can be encounter.

- User request has to process faster and if cloud taking too much time for response then this will be severe problem when patient has to diagnose with heart disease. In propose paper this problem is solving by using Edge Node where mobile will choose nearest and free available Edge Node to process request so response time can be minimize..

- To predict disease using machine learning algorithms we need to train the algorithm with existing dataset and if this dataset exposed then patient privacy will be leak. To overcome from this problem author is encrypted patient data for training using Light Weight Homomorphic algorithm and this algorithm not require any decryption process and we can directly operate on encrypted data without performing decryption

In propose work author providing privacy to patient data by applying Light Weight Privacy Preserving Medical Diagnosis (LPME) and ETC classifier. LPME will encrypt patient train and test data and ETC will predict disease from that encrypted data.

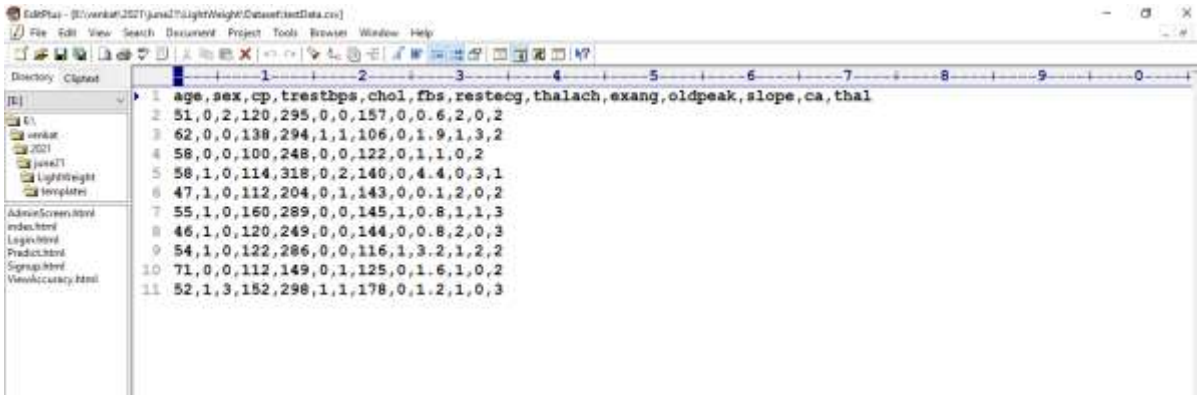
Below code screen showing steps to generate keys, data encryption and decryption process of LPME technique



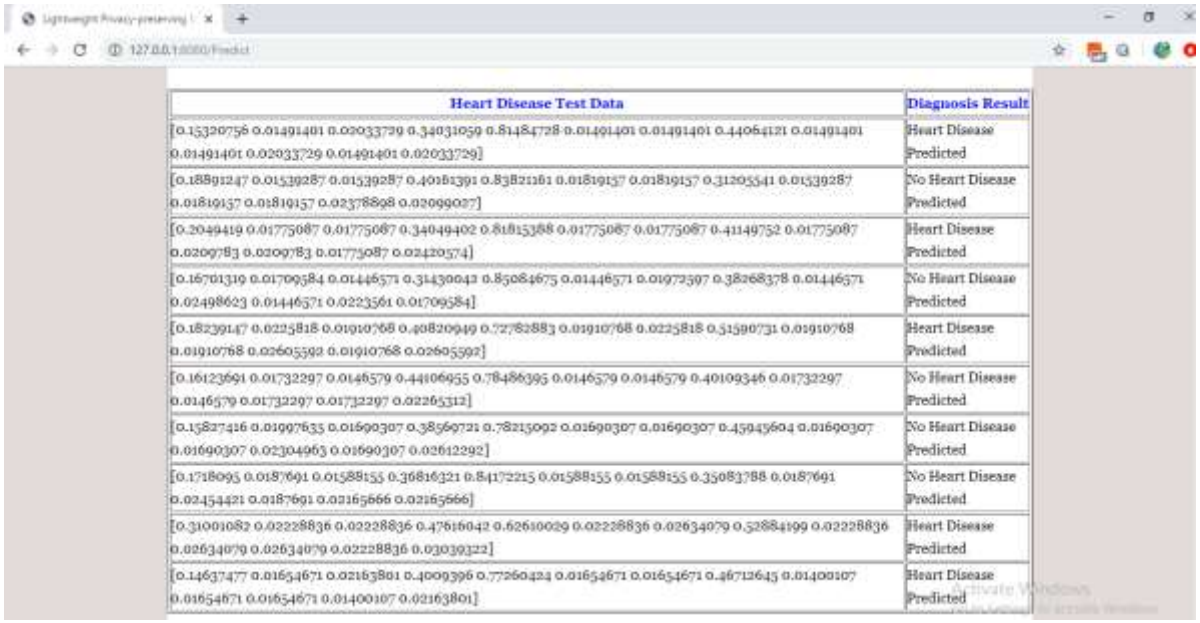
```
File Edit Format Run Options Window Help
import numpy as np
from numpy.polynomial import polynomial as poly
import pandas as pd


#functions to generate random keys
def polymul(x, y, modulus, poly_mod):
    return np.int64(np.round(poly.polydiv(poly.polymul(x, y) % modulus, poly_mod)[1] % modulus))

def polyadd(x, y, modulus, poly_mod):
    return np.int64(np.round(poly.polydiv(poly.polyadd(x, y) % modulus, poly_mod)[1] % modulus))

def gen_binary_poly(size):
    #return np.random.randint(0, 2, size, dtype=np.int64)
    sk = [1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1]
    sk = np.asarray(sk)
    return sk


def gen_uniform_poly(size, modulus):
    #return np.random.randint(0, modulus, size, dtype=np.int64)
    a = [59065, 14354, 61150, 5079, 4913, 61939, 40212, 15209, 1914, 64176, 15360, 36874,
         23392, 5500, 44438, 5577, 40549, 13446, 2156, 12972, 30499, 60878, 41096, 63075, 28446, 56251, 56856, 61123, 10296, 13870, 3770, 5642]
    a = np.asarray(a)
    return a


def gen_normal_poly(size):
    #return np.int64(np.random.normal(0, 2, size=size))
    e = [0, 1, 0, 1, 0, -1, 0, 1, 0, -2, -1, 1, -1, 4, 0, -1, -2, 1, -3, -1, -2, 0, 1, 2, -3, -3, 0, 0, 0, -1, 0, 2]
    e = np.asarray(e)
    return e
```

```
File Edit Format Run Options Window Help
    e = np.asarray(e)
    return e

#function to generate random key
def keygen(size, modulus, poly_mod):
    sk = gen_binary_poly(size)
    temp = sk.tolist()
    print(temp)
    a = gen_uniform_poly(size,modulus)
    temp1 = a.tolist()
    print(temp1)
    e = gen_normal_poly(size)
    temp2 = e.tolist()
    print(temp2)
    b = polyadd(polymul(-a, sk, modulus, poly_mod), -e, modulus, poly_mod)
    return (b, a), sk

#function to encrypt data
def encrypt(pk, size, q, t, poly_mod, pt):
    m = np.array([pt] + [0] * (size - 1), dtype=np.int64) % t
    delta = q // t
    scaled_m = delta * m  % q
    e1 = gen_normal_poly(size)
    e2 = gen_normal_poly(size)
    u = gen_binary_poly(size)
    ct0 = polyadd(polyadd(polymul(pk[0], u, q, poly_mod), e1, q, poly_mod), scaled_m, q, poly_mod)
    ct1 = polyadd(polymul(pk[1], u, q, poly_mod), e2, q, poly_mod)
    return [ct0, ct1]

#function to decrypt data with privacy
def decrypt(sk, size, q, t, poly_mod, ct):
    scaled_pt = polyadd(polymul(ct[1], sk, q, poly_mod), ct[0], q, poly_mod)
    decrypted_poly = np.round(scaled_pt * t / q) % t
    return int(decrypted_poly[0])
```

In above two screens read red colour comments to know about LPME encryption using Homomorphic algorithm.



Above is the test data which ETC will encrypt and perform prediction of disease and if you want u can add new records to above test data and this testData.csv file is available inside 'Dataset' folder



Heart Disease Test Data	Diagnosis Result
[0.15320756 0.01491401 0.02033729 0.34031059 0.81484728 0.01491401 0.01491401 0.44064121 0.01491401 0.01491401 0.02033729 0.01491401 0.02033729]	Heart Disease Predicted
[0.18891247 0.01539287 0.01539287 0.40161391 0.83821161 0.01819157 0.01819157 0.31205541 0.01539287 0.01819157 0.01819157 0.02378898 0.02099027]	No Heart Disease Predicted
[0.2049419 0.01775087 0.01775087 0.34049402 0.81815388 0.01775087 0.01775087 0.41149752 0.01775087 0.0209783 0.0209783 0.01775087 0.02420574]	Heart Disease Predicted
[0.16701319 0.01709584 0.01446571 0.31430043 0.85084675 0.01446571 0.01972597 0.38268378 0.01446571 0.02498623 0.01446571 0.0223561 0.01709584]	No Heart Disease Predicted
[0.18239147 0.0225818 0.01910768 0.40820949 0.72782883 0.01910768 0.0225818 0.51590731 0.01910768 0.01910768 0.02605592 0.01910768 0.02605592]	Heart Disease Predicted
[0.16123691 0.01732297 0.0146579 0.44106955 0.78486395 0.0146579 0.0146579 0.40109346 0.01732297 0.0146579 0.01732297 0.01732297 0.02265312]	No Heart Disease Predicted
[0.15827416 0.01997635 0.01690307 0.38569721 0.78215092 0.01690307 0.01690307 0.45945604 0.01690307 0.01690307 0.02304963 0.01690307 0.02612292]	No Heart Disease Predicted
[0.1718095 0.0187691 0.01588155 0.36816321 0.84172215 0.01588155 0.01588155 0.35083788 0.0187691 0.02454421 0.0187691 0.02165666 0.02165666]	No Heart Disease Predicted
[0.31001082 0.02228836 0.02228836 0.47616042 0.62610029 0.02228836 0.02634079 0.52884199 0.02228836 0.02634079 0.02634079 0.02228836 0.03039322]	Heart Disease Predicted
[0.14637477 0.01654671 0.02163801 0.4009396 0.77260424 0.01654671 0.01654671 0.46712645 0.01400107 0.01654671 0.01654671 0.01400107 0.02163801]	Heart Disease Predicted

In above screen in first column you can see then encrypted test data and in second column you can see prediction result as 'No Heart Disease Detected' or 'Heart Disease Detected.

5. CONCLUSION AND FUTURE SCOPE

The proposed research work presents a comprehensive solution to the challenges associated with mobile-based health symptom prediction and disease diagnosis, with a strong emphasis on patient data privacy and response time optimization. By introducing Edge Nodes into the healthcare ecosystem, the research effectively reduces response times, ensuring that patients receive timely diagnoses, particularly critical in cases like heart disease. The use of Light Weight Homomorphic Encryption safeguards patient data during machine learning model training and prediction, eliminating the risk of data exposure. The LPME framework, which combines this encryption technique with the ETC classifier, offers a robust approach to disease prediction while preserving patient privacy. Overall, this research paves the way for secure, efficient, and privacy-conscious healthcare applications that can revolutionize the delivery of medical services in the digital age.

REFERENCES

[1]. Shah, A.A.; Piro, G.; Grieco, L.A.; Boggia, G. A qualitative cross-comparison of emerging technologies for software-defined systems. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019; pp. 138–145.

[2]. Ali, A.; Mehboob, M. Comparative analysis of selected routing protocols for wlan based wireless sensor networks (wsns). In Proceedings of the 2nd International Multi-Disciplinary Conference, Gujrat, Pakistan, 19–20 December 2016; Volume 19, p. 20.

[3]. Shah, A.; Piro, G.; Grieco, L.A.; Boggia, G. A review of forwarding strategies in transport software-defined networks. In Proceedings of the 2020 22nd International Conference on Transparent Optical Networks (ICTON), Bari, Italy, 19–23 July 2020; pp. 1–4.

[4]. Bruce, R.R.; Cunard, J.P.; Director, M.D. From Telecommunications to Electronic Services: A Global Spectrum of Definitions, Boundary Lines, and Structures; Butterworth-Heinemann: Oxford, UK, 2014.

[5]. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and smart contracts for insurance: Is the technology mature enough? Future Internet 2018, 10, 20.

[6]. Jia, B.; Zhou, T.; Li, W.; Liu, Z.; Zhang, J. A Blockchain-Based Location Privacy Protection Incentive Mechanism in Crowd Sensing Networks. Sensors 2018, 18, 3894.

[7]. Biswas, K.; Muthukkumarasamy, V. Securing smart cities using blockchain technology. In Proceedings of the 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, Australia, 12–14 December 2016; pp. 1392–1393.

[8]. Fernández-Caramés, T.M.; Froiz-Míguez, I.; Blanco-Novoa, O.; Fraga-Lamas, P. Enabling the Internet of Mobile Crowdsourcing Health Things: A Mobile Fog Computing, Blockchain and IoT Based Continuous Glucose Monitoring System for Diabetes Mellitus Research and Care. Sensors 2019, 19, 3319.

[9]. Ali, A.; Naveed, M.; Mehboob, M.; Irshad, H.; Anwar, P. An interference aware multi-channel mac protocol for wasn. In Proceedings of the 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), Karachi, Pakistan, 5–7 April 2017; pp. 1–9.

[10]. Beebeejaun, A. VAT on foreign digital services in Mauritius; a comparative study with South Africa. Int. J. Law Manag. 2020, 63, 239–250.

[11]. Aziz Shah, A.; Piro, G.; Grieco, L.A.; Boggia, G. A quantitative cross-comparison of container networking technologies for virtualized service infrastructures in local computing environments. Trans. Emerg. Telecommun. Technol. 2021, 32, e4234.

[12]. Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A.; Choo, K.-K.R. Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks. IEEE Trans. Netw. Sci. Eng. 2019, 8, 1120–1123.