International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(1): 157-170 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i01.0182

Privacy-Preserving Medical Diagnosis System Using Harris Hawk Optimization in Edge Computing

Malathy N*, Grace Sophia J, Swathi S, Vijaya Subasri K

Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu. India. *Corresponding Author's Email: malathy@mepcoeng.ac.in

Abstract

With the enhancement of artificial intelligence and machine learning, nowadays it is in trend that mobile clients acquiesce their sign for diagnosing medical illnesses. Edge computing methodology is frequently used for medical diagnosis as it reduces the transmission latency and allows users and devices even at remote locations to analyze the data at the edge of the network. Since data-driven machine learning algorithms need to develop an identification system over huge medical data, they may be concerned about the privacy of data leakage during medical diagnosis. To solve this issue in our work a privacy-preserving medical diagnosis system is developed on edge. With our model, we encrypt the user input during the submission of the user input and that will be diagnosed for the disease and the user will be given the result in encrypted form which he could decrypt, to preserve the privacy of the user. The security and experimental analysis of our model explains the efficiency of our proposed system. The gradient boosting (CatBoost) model is redesigned by following the cloud-edge model, which accepts the ciphered model parameters rather than usual data to get rid of the amount of cipher to plain text computation using Triple-DES. In addition, we have optimized our model using the Harris Hawk Optimisation technique. Additionally, our algorithm offers private and prompt diagnosis while maintaining secure diagnosis on the edge. Our security study and investigational assessment depict that our algorithm is effective, efficient and secure.

Keywords: Catboost, Ciphertext, DES, Gradient boosting, Harris hawk optimization.

Introduction

Machine learning in the medical field largely focuses on creating algorithms and methods to assess if a system's behavior in diagnosing diseases is accurate. Machine learning (ML) algorithms are being applied to the diagnosis of illnesses is one illustration of the medical field's benefit from this technology. To detect diseases early and enhance therapies, machine learning technology can help find hidden or complex patterns in diagnostic data. We located several technologies, some of which enhance their accuracy by learning from new data, both in use and in development. One of the areas where mobile devices can have the biggest influence is personalized diagnostics. Diagnosing based on Machine learning has enormous advantages in enhancing the eminence of healthcare services and staying away from costly diagnosis charges when compared to the dearth of experts and the high cost of manual diagnosis. Due to this, both academic and industrial fields have given machine learning-based medical diagnosis a lot of attention. More and more requirements have risen as a result of the development of telemedicine applications in the fields of mobile telemedicine and clinical healthcare (1-4), and mobile telemedicine (4). However, the growth has also been escorted by several issues, including a lack of training data, security flaws, and privacy worries. It is a major problem in medical practice that gathering enough medical data takes a lot of time and money.

The amount of health data stored in a single medical source is usually limited, which makes the development of data-driven machine learning difficult. To create a suitable diagnosis model, it is necessary to exchange the training data that is dispersed throughout numerous medical institutions. With the advent of cloud computing, machine learning on outsourced medical data has been extensively studied because of the advancements in large storage capacity and infinite processing power (5, 6). However, the increased frequency of interaction between mobile users and the cloud leads to undesirable transmission delay and slow request answers (7-9). Patients' lives, health, and medical safety are

(Received 27th October 2023; Accepted 13th January 2024; Published 30th January 2024)

significantly impacted by a delayed diagnosis response, particularly those who have been diagnosed with an acute disease (such as acute heart disease, pneumonia, etc.)., Edge computing is a novel paradigm for computing, has been proposed to solve this problem by using edge nodes that are located close to mobile users to reduce latency and deliver efficient computing services (10-13). The development of machine learning techniques based on edge computing during the past few years (14-16) has been significant in enhancing the effectiveness of diagnosis using edge computing.

Adopting a real-time high-performance edge model and accurate diagnosis of medical issues is crucial to focus on the vulnerability in medical diagnosis. The gradient boosting (CatBoost), the most advanced machine learning model, exhibits exceptional talent in Kaggle contests thanks to its great prediction performance in the distributed scenario. Additionally, thanks to its tree-based structure, CatBoost has improved explainability and simplicity. As a result, many systems have used the CatBoost model to make medical diagnoses (17-19), however, they disregard the essential issue of information security during the training stage. When patients are told they have a private disease (like HIV or Hepatitis B virus), they frequently experience some psychological resistance. It is seen as a factor in the condition's deterioration. Consequently, it is essential to preserve their privacy. In addition to the fact that a lot of sensitive information is contained in medical data, more and more data are prohibited from being transformed into plaintext as a result of privacy policies (such as GDPR (20) and HIPPA (21). Protecting the privacy of medical analysis in the context of edge computing is therefore crucial. (22-25).

Homomorphic encryption (HE) (26) is a potential approach to address privacy issues since it minimizes the likelihood of information leakage while preserving data secrecy. The currently used single or dual-cloud mode cloud computing framework, which has been expanded to include edge computing, is the main support for the privacy-preserving machine learning algorithms based on HE (27). Unfortunately, because the private key is preserved in a single cloud rather than two, the single-cloud model (28) is more likely to cause a privacy leakage than the dual-cloud model. Sensitive data are exposed once the cloud has been compromised. In addition, the dual-cloud model's practical uses are restricted by the high presumption of non-collusion among two partial honest cloud servers (29-31). Additionally, a significant amount of secure computation over encrypted data is required during the machine learning training phase. The first difficult problem arises from the growing amount of outsourced encrypted data, which places a significant computational burden, particularly for edge nodes with limited resources (32). Thus, it is crucial to consider lightweight when utilizing privacy-preserving machine learning in edge computing. Our privacy-preserving CatBoost over encrypted model parameters greatly reduces computational overhead when compared to data-sharing-based privacy-preserving machine learning. In this study, we introduce the Secure Encrypted Machine Learning model in edge computing. Our algorithm primarily uses the following constructions:

- CatBoost on edge: Using model parameters developed over many edge nodes rather than training data, the algorithm builds a CatBoost-based diagnosis model, eliminating the limitations of cumbersome training data storage and ensuring the viability of CatBoost.
- Privacy-preserving training: The algorithm creates HE-based secure computing with a single-cloud model, choosing the best parameters over encrypted model parameters. Only one part of the secret key is kept in the single cloud because it is randomly split into two parts. Thus, the single cloud model can guarantee the dependability of the privacy-preserving training on the resource-constrained edges in addition to offering strong privacy preservation for training the lightweight CatBoost.
- Secure diagnosis on CatBoost at the edge: A mobile user can send an edge encrypted query, and the edge will provide the associated diagnosis results. This is how the algorithm offers secure diagnosis. It is used throughout the procedure to ensure the privacy of the returned diagnosis results for executing the private and prompt diagnosis.

To ensure privacy preservation throughout the training phase, previous jobs on preserving privacy using machine learning (33-35) have been offered, however, these schemes lacked implementation. Since then, more and more plans

to preserve privacy have been put out. To support matrix factorization with encrypted data, A privacy-preserving non-negative matrix factorization technique based on additionHomomorpic Encryption was presented by Fu *et al*. (36). However, since these matrix parameters can be obtained by a third party during the computation process, there is a risk of privacy leakage, which can be acquired by a third party during the computation process, which could result in privacy issues.

Ma et al. suggested a random tree architecture with a Paillier cryptosystem that offered accurate and secure training on encrypted data (29). To build a model without disclosing personal information, Wang *et al.* (31) presented a collaborative neural network approach that protects privacy. A privacy-preserving approach for productively training neural networks was created by Mohassel *et al.* (33). The HE-based processes used in the aforementioned methods (37) are useful for machine learning while preserving privacy. Nevertheless, there is a large overhead associated with the safe calculation that was applied over a sizable volume of encrypted data (31).

To tackle the aforementioned issue, a model-sharing-based privacy-preserving machine learning framework has been created that outsources encrypted model parameters instead of utilizing more local data. In addition to ensuring the training of machine learning, it can shift some computation from being outsourced over ciphertexts to being performed locally over plaintexts, which can increase efficiency and lighten the workload. Yu et al. initially created a methodology based on models that were outsourced from different data owners without revealing local data (37). However, this approach uses random numbers in place of encryption technology, which is very susceptible to inference attacks that result in privacy breaches (38).

After that, Cheng *et al.* (39) suggested an encrypted model parameter-based secure XGBoost. These settings can, however, be accessed and decrypted by a different party. The security of local data may be in danger because the parameters also contain critical information. Li *et al.* (40) established a secure classification service with outsourced encrypted Support Vector Machine (SVM) models; nevertheless, it cannot offer privacy-preserving model training. Based on encrypted models, Aono *et al.* (41) developed a privacy-preserving deep learning system without revealing the participants' local data to a server. This approach significantly sped up the execution of associated secure computation while maintaining accuracy. Unfortunately, Wang *et al.* (42) showed how the single-cloud model will result in privacy leakage from the aforementioned schemes (40, 41). When the cloud is breached, it is simple to leak due to the trained model's privacy.

The dual-cloud approach is used to avoid the single-cloud model's limitations and stop data leakage during computation. Liu *et al.* (30, 31) proved the security and accuracy of secure computation using a dual-cloud server technique. Additionally, Hu et al.'s research (28) demonstrated that the non-colluding dual-cloud model outperformed the single-cloud model in terms of security. Even if one server is compromised, the other server's presence prevents it from leaking the trained model's privacy information. Privacy concerns in edge computing were taken into account in Liu *et al.*'s (27) adaptation of the secure computation based on a dual cloud model to the edge computing environment. Unfortunately, sending encrypted data between two cloud servers is necessary to perform secure computation, which increases computational overhead and communication load. Additionally, each resource-constrained edge node needs to do six modular multiplication operations, two modular addition operations, and five modular exponentiation operations to perform safe computation, which is not practical in an edge computing environment. Despite the suggestion of Zhang *et al.*'s (43) privacy-preserving feature transform on edge with lightweight, robust privacy preservations cannot be guaranteed because the submitted photos were only provided in plaintext. To the extent of our understanding, the edge computing literature does not account for the trade-off between privacy concerns and lightweight (46-49). In addition to efficiency and real-time model training, we develop a lightweight privacy-preserving machine learning system with good privacy preservations on the edge. The authors in (49) proposed efficient task scheduling in fog computing but they didn’t focus on security.

Because we employed the effective method catboost, our research improves prediction

accuracy. Unlike xgboost, which requires preprocessing of categorical data, catboost allows our approach to run more quickly (45). The most pertinent features were chosen from the dataset using the Harris Hawk optimization technique, which improved the feature selection process and may have reduced computing complexity as well as improved the performance and interpretability of the model. When compared to other research that has already been done, our system did well with these (49). Furthermore, compared to RSA, the Ed25519 key generation technique is quick and difficult to break. Ed25519 serves as the authentication key. We used TripleDES to provide secure communication of diagnosis from the edge platform to mobile users and symptoms from mobile users to the edge platform. Because it uses three 56-bit keys to provide a better level of security, the TripleDES algorithm works well. Our algorithm performs better than other algorithms and research works thanks to these adjustments (47-49).

Methodology

This section provides a detailed description of the machine learning technique CatBoost, the Triple DES encryption algorithm, and the Ed25519 key generation algorithm.

Harris Hawks Optimization Algorithm

Input: Existing problems

Output: Optimized dataset

1. Initialize the population
2. In the population- Evaluate the fitness value
3. Identify the best fitness score
4. Divide the population into different clusters and promote cooperation between them.
5. Perform Exploration - by randomly modifying the solutions.
6. Perform Exploitation - by making small modifications to the best solution.
7. Repeat the steps from 2 to 5 until the best solution is found.

Figure 1 represents the workflow of the Harris Hawks Optimization Algorithm. The algorithm uses a set of formulas to update the position of the Hawks using equations 1-4.

1. Position update formula for exploratory hawks:

$$x_{new} = x_{old} + rand() * (x_{best} - 2 * x_{old}) \quad [1]$$

where x_old is the current position of the hawk, x_best is the position of the elite hawk with the best fitness and rand() is a random number between 0 and 1.

2. Position update formula for elite hawks:

$$x_{new} = x_{old} + rand() * (x_A - x_B) \quad [2]$$

where x_old is the current position of the elite hawk, x_A and x_B are the positions of two randomly selected elite hawks, and rand() is a random number between 0 and 1.

3. Calculation of step size:

$$step\_size = l * \exp(-c * iter) \quad [3]$$

where l is the initial step size, c is a constant value, iter is the current iteration number.

4. Updating the position using the step size:

$$x_{new} = x_{old} + step\_size * randn() \quad [4]$$

where randn() is a random number generated from the normal distribution.

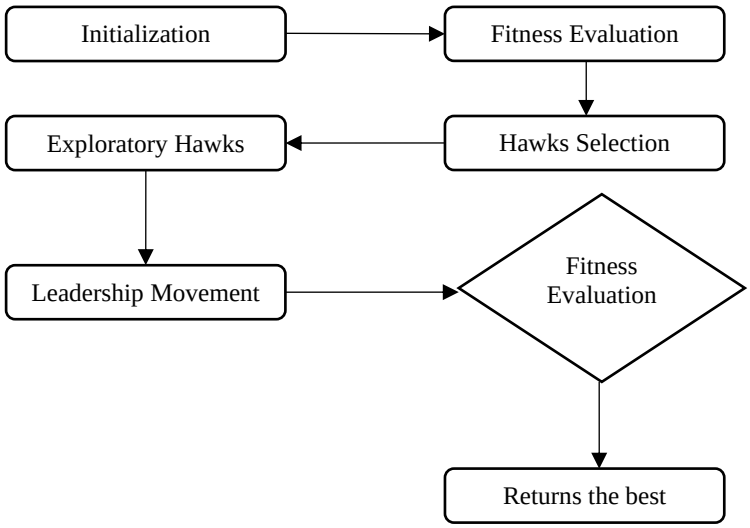

Figure 1: Harris Hawks Algorithm

Taking privacy restrictions into account, the offloading procedure may be optimized using the HHO algorithm. In edge computing applications, the HHO algorithm can help maintain privacy by intelligently choosing the best edge node for data processing while taking privacy needs into account.

CatBoost Algorithm

CatBoost manages category features automatically. Consider a dataset D with n samples. Each sample has a real-valued goal, y, and m sets of features in a vector, x. Given a loss function $L(y_i,F^t)$, gradient boosting adopts an additive version which successively builds a series of approximations Ft in a greedy manner using equations 5-7. The i^{th} expected output value y_i and the t^{th} function F_t that estimates y_i are the two values that make up the loss function. After creating function F_t, we can find another function

$F_t = F^{t-1} + \alpha.h_t$ [5]

where α is a step size and function h_t is a base predictor selected from a family of functions H to minimize the anticipated loss, to enhance our estimations of yi. That is,

$h^t = arg_{h\in H}min\ £L(y,F^{t-1} + h)$ [6]

Taylor approximation or negative gradients are used to approach the minimization in this way:

$h_t = arg_{h\in H}min\ £\left(\frac{\delta L_\gamma}{\delta F^t} - 1^{-h}\right)^2$ [7]

Ed25519 KeyGeneration Algorithm

Ed25519 utilizes tiny private keys (32 or 57 bytes, respectively), tiny public keys (32 or 57 bytes), and tiny signatures (64 or 114 bytes) with a high level of security (128 or 224 bits, respectively).

The EdDSA contains the private key as priv_Key and the public key as pub_Key using equation 8.

pub_Key = priv_Key * G. [8]

The seed, a random integer used to create the private key, is used.The open key Using the EC point multiplication, pub_Key is a point on the elliptic curve: (The private key multiplied by the curve's generating point G) pubKey = privKey * G.

EdDSA Sign (EdDSA_sign)

- Add G to pubKey to calculate pubKey.
- Create a secret integer r = hash(hash(priv_Key) + message) mod q) deterministically. (This is somewhat condensed)
- Multiply it by the curve generator to determine the public key point hidden behind r: R = r * G
- The formula for h is h = hash(R + pub_Key + message) mod q.
- Put s = (r + h * priv_Key) mod q to work.
- R, s, deliver the signature.

EdDSA Verify Sign(EdDSA_verify_sign)

- The formula for h ish = hash(R + pubKey + message) mod q.
- Calculate P1 = s * G, and then calculate P2 as R + h * pub_Key.
- Return P1==P2

The fact that the points P1 and P2 are identical EC points establishes that the points P1, generated by the associated private key, and P2, produced by the corresponding public key, are identical.

Secure Computation using Triple DES

The Triple Data Encryption method (TDEA or Triple DEA), sometimes known as Triple DES (3DES or TDES), is a symmetric-key block cipher that employs the DES cipher method on three occasions for each data block. Without having to create an entirely new block cipher algorithm, Triple DES offers a reasonably easy way to increase the key size of DES to thwart such attacks. Three DES keys, K1, K2, and K3, totaling a combined total of 56 bits (excluding parity bits), make up the "key bundle" used by Triple DES. This is the encryption algorithm using equations 9-10.

$Cipher = E_{K3}(D_{K2}(E_{K1}(plaintext)))$ [9]

The decryption algorithm is:

$Plaintext = D_{K1}(E_{K2}(D_{K3}(ciphertext)))$ [10]

Where E denotes encryption and D denotes decryption.

System Model

The core components of our system concept are the Key Generation Centre (KGC), Cloud Platform (CP), Edge Nodes (ENs), and Mobile Users (MUs) as shown in Figure 2. Assume that the system contains N ENs.It should be noted that a secure channel, such as Secure Socket Layer (SSL) or Transport Layer Security (TLS), is used to synchronize the communication between these entities. Each entity's specific role is illustrated as follows:

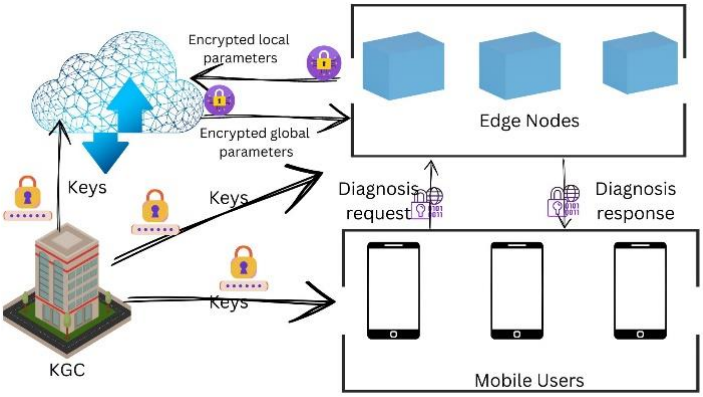

Figure 2: System Model

- Key generation center: Our system's secret shares are sent to other entities for use in safe computing in the future, and the Key generation center is completely trusted to create, manage, and distribute those shares. (Step 1).
- Edge node: An EN is a medical institution with limited storage and processing power that stores a small amount of medical data. An EN is willing to work cooperatively with other ENs to create a global model during the training phase. This model then submits locally optimal model parameters after encryption and offers compute services to the CP to carry out secure computation. (Step 2).
- Cloud platform: The computing and storage capacities of CP are infinite. To build a global model, it first gets the encrypted model parameters from several ENs and then selects the globally optimal model parameters (Step 3).
- Mobile user: An EN in the area can receive an encrypted diagnosis request from a MU (Step 4), and the MU can then get the encrypted diagnosis result from the EN (Step 5). To protect diagnosis privacy, a MU and the EN work together to implement secure computing of the diagnosis phase.

Threat Model

For the hostile perspective, we take into account the potential risks to the system based on the data that system entities (such as CP, ENs, and MUs) have access to.

- Objects making threats: Assume that KGC can be trusted to distribute keys. MUs, ENs, and CP are viewed as sincere but inquisitive organizations that follow rules but try to decipher encrypted content to find out more. In actuality, cooperation between CPs and ENs compromises EN privacy, but cooperation between an EN and a MU also compromises personal data about particular individuals. It is not worthwhile to collaborate with other organizations to accomplish CP, EN, and MU in order to stop the disclosure of personal information. We assume that neither CP nor ENs nor MUs are coordinating with one another.
- Threats from an outside enemy: It is assumed that the data transmitted over the communication channels between the MUs and ENs and the CP and ENs, respectively, can be intercepted by an external attacker. Additionally, a foe may pervert an EN, MU, or CP.

System Framework

Here, we present a thorough explanation of how to build a fundamental CatBoost architecture for the global diagnostic model. Next, to provide a quick and private diagnosis service, we suggest a secured diagnosis on edge.

The proposed system contains 3 stages:

- Key generation: The EdDSA_sign() function is invoked to generate private and public key pairs during signup. Then, the Ed25519 verifysign() function is invoked on the login page to check the identity of the user. The user gets authenticated only if the verifysign() method is true.
- Privacy-preserving CatBoost: Before delivering decision nodes to the CP, ENs locally train and encrypt them to create a global model over N ENs (Step 1). CP then designates the global node as the optimal split of a decision node among the submitted nodes to accomplish global optimization (Step 2). At last, each EN creates neighborhood leaf nodes (Step 3).
- Secure diagnosis on edge: Before sending symptoms to a nearby EN, a MU must encrypt them to create a secure diagnosis service. The

anonymity of the provided symptoms and the provided diagnosis results must be maintained.

Privacy-preserving CatBoost

We suppose that different ENs build the global model collectively without exchanging training data. Many ENs are thought to store data with a non-i.i.d distribution, which means that the global distribution and each individual biased distribution are preserved without losing generality. As a result, the final trained model of each EN retains local distinctions in addition to learning knowledge across all ENs.Specifically, over N ENs, the suggested privacy preserving CatBoost is built. The representation of the kth tree model is $F_k(x)= w_q(x)$ during the training of the kth round. Decision nodes and leaf nodes make up the tree nodes, with a split value included in each decision node. Figure 3 shows the overall workflow

BuildDecisionNode

The construction of the decision node involves two steps by applying equations 11-12:

- Building locally optimal split
- Building globally optimal split

BuildLocalNode: The optimal splitting is chosen for the ith (i € (1,N)) EN while maximizing Gain.

Gain= ½ * gain $-\psi$ [11]

Gain= $\frac{G_L^2}{H_L+\psi} + \frac{G_R^2}{H_R+\psi}$ [12]

BuildGlobalNode

The Cloud Platform will locate the split with the maximum gain to execute the globally optimal split after obtaining the locally optimal split and the evaluation indexes from N Edges. It also returns the global optimum in an encrypted format to each of the edge nodes and the edge nodes in turn decrypt the parameter and use it.

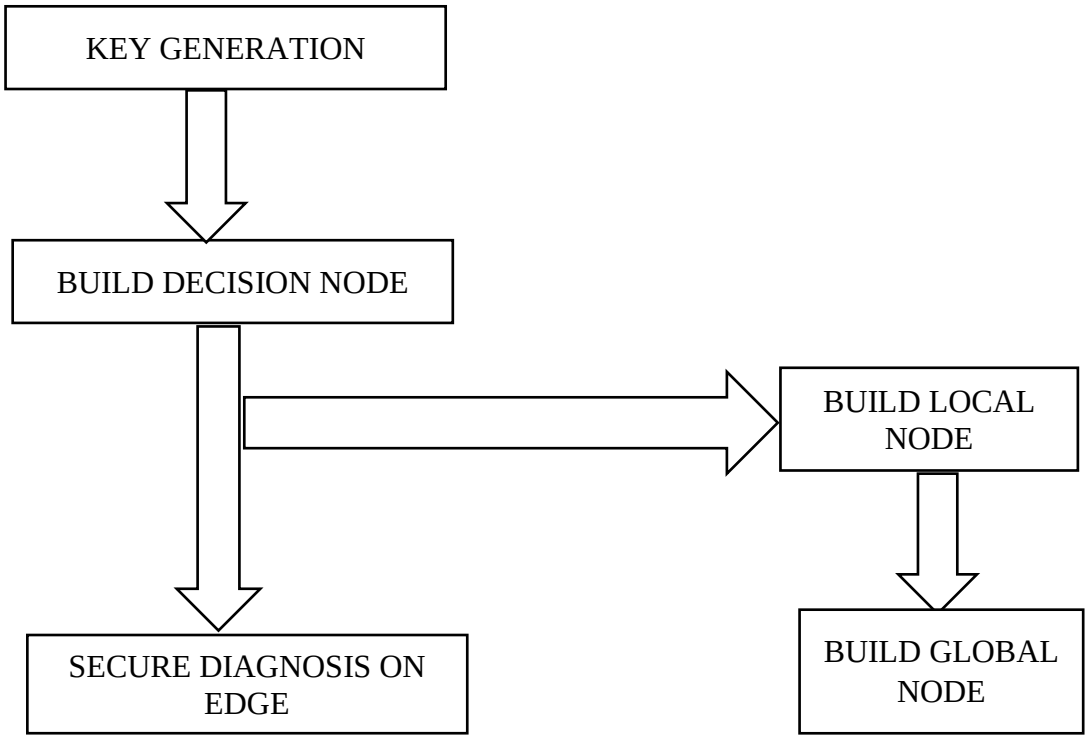

Figure 3: Overall System Architecture

Algorithm 1. Building Decision Node

Input: Training dataset X with n instances and m features, where each instance is represented by a feature vector x_i and corresponding class labels y_i.
Output: Decision tree T represents the local edge node.

1. **Initialization**: Create an empty root node root for the decision tree T.
2. **Stopping Criteria**: If any of the following conditions are met, stop growing the tree and return the current node as a leaf node:
    - All instances in the current node belong to the same class.
    - All features have been used for splitting.
    - The maximum depth of the tree has been reached.
3. **Feature Selection**: Select the best feature f_best from the remaining unused features based on a suitable criterion (e.g., information gain, Gini index, etc.).
4. **Splitting**: Split the current node into child nodes based on the selected feature f_best. Iterate over all possible feature values v of feature f_best and create a child node for each value.
5. **Instance Partitioning**: Partition the instances in the current node into the corresponding child nodes based on the feature value v of feature f_best. Assign an instance to the child node whose feature value matches v.
6. **Recursion**: Recursively apply steps 4-7 to each child node until the stopping criteria are met.
7. **Leaf Node Creation**: If a child node becomes empty after partitioning, create a leaf node and assign the majority class label of the instances in the parent node as its predicted class label.

Return the decision tree T.

Algorithm 2. Building Global Node

Input: Encrypted gains, Encrypted Local decision tree
Output: Globally Optimal decision tree

1. Initialize variables: v_f_b = v_score_b, v_s_b = v_score_b.
2. For n in range 0 to N:
    - **CompareEncIndex**: Compute v_score_a_hash = v_score_hash_b * v_score_a.
    - Update v_f_b using scalar multiplication: v_f_b *= v_score_a_hash.
    - Update v_s_b using scalar multiplication: v_s_b *= v_score_a_hash.
3. Compute v_s_com = v_score_b * v_score_hash_a - v_score_hash_b * v_score_a.
4. If v_score_a_n - v_score_hash_a_n * v_score_a_n< 0, then:
    - Update v_score_a_n with v_score_hash_a_n.
    - Update v_score_hash_a_n with v_score_a_n.
5. Return v_f_b, v_s_b

Algorithm 3. Secure Diagnosis on Encrypted CatBoost

Input: Triple DES Encrypted instance (enc_sym), Triple DES encrypted CatBoost(CB) = {"((" F_K "))" ^"K" ⫽ "k=1" } comprises of K encrypted trees.
Output: Encrypted diagnosis result (enc_ŷ).

1. Set (enc_ŷ) to (0).
2. For 1 ≤ k ≤ K, do the following:
    - Set (node) as the root node of (F_K).
    - While true, do the following:
        - If (node) is a leaf node, then:
            - Obtain label weight (w) from the leaf node.
            - Update (enc_ŷ) by multiplying it with (w): (enc_ŷ) ← (enc_ŷ) × (w).
            - Break the loop.
        - Else, do the following:
            - Obtain the split threshold (s) on f-th feature from (node).

- Obtain the f-th feature value (sym_f) from (enc_sym).
- Apply Triple DES((s), (sym_f)).
- If (s) ≤ (sym_f), then:
  - Set (node) as the left child of (node).
- Else, do the following:
  - Set (node) as the right child of (node).

3. Return (enc_$\hat{y}$) as the encrypted diagnosis result.

The diagnosis result is computed using equation 13.

$$\hat{y} = \sum_{k=1}^{K} F_k(sym) \quad [13]$$

Secure Diagnosis in Edge

We design a secure diagnosis method between the EN and MU while considering the limitations of MUs' limited processing power and the confidentiality of symptoms submitted. To finish, we employ secure lightweight computation. Note that an EN owned by a medical facility maintains the encrypted local diagnosis model for secure diagnosis over encrypted requests and that sensitive data is contained in the parameters of EN's trained model and the information of MU's requests. Therefore, it is essential to guarantee strong privacy preservations without leaking any privacy during the diagnosis process.

Results and Discussions

Attack Analysis

We categorize these into groups based on the descriptions in Section 4.2attacks into the following categories when operating in a hostile environment.

Type-I: Corruption: Assuming that a rival tries to exert pressure and collaborate with CPs, ENs, and MUs to access private information and change secret keys stored in these entities.

There are three attack models used in this kind of attack.

- Text-only cipher attack model: The adversary can look at parameters that are encrypted and try to figure out secret keys.
- Known-sample attack paradigm: In a known-sample attack paradigm, an adversary can obtain certain plaintext parameters and their corresponding ciphertexts, and then try to derive the secret keys.
- Chosen-plaintext: The opponent can encrypt specific plaintexts to extract the corresponding ciphertexts, which can then be used to deduce secret keys.

Type-II: Eavesdropping: In the unlikely event that a hostile party attempts to listen in on the data being sent across the communication channel, he or she will attempt to discover private information about other people and extract these private facts.

Performance Analysis

The experiment setting is described in detail in this part, followed by an examination of the theoretical performance in comparison to other privacy-preserving schemes.

Experimental settings

We use two open datasets to conduct our evaluation as shown in Table 1.

Table 1: Summary of the datasets used in our study

Dataset	Objects	Features
Heart	303	14
Thyroid	3163	25

1. Heart disease dataset: It has two labels, "0" for a patient without heart illness and "1" for a patient with heart disease, totaling 303 instances, 14 traits, and two labels.
2. Thyroid disease dataset: It has two labels, "0" for a patient without thyroid disease and "1" for a patient with thyroid disease. There are 3,163 instances in total, along with 25 features.

The system we propose is written in Java, and a PC tester with 3.30 GHz four-core CPUs and 4 GB of RAM is used to evaluate the trials. We use the cross-validation approach to split the dataset into thirds and use the remainder as the validation set. Catboost is used with the parameters g 14 0, c 14 20, and sampling rate rates as 14 80% over local training data of each EN to train a model over several ENs.The entire procedure for constructing an XGBoost's kth tree involves constructing decision nodes across N ENs using BuildLocalNode and BuildGlobalNode. Each EN constructs the section 3 local decision nodes during the BuildLocalNode phase.

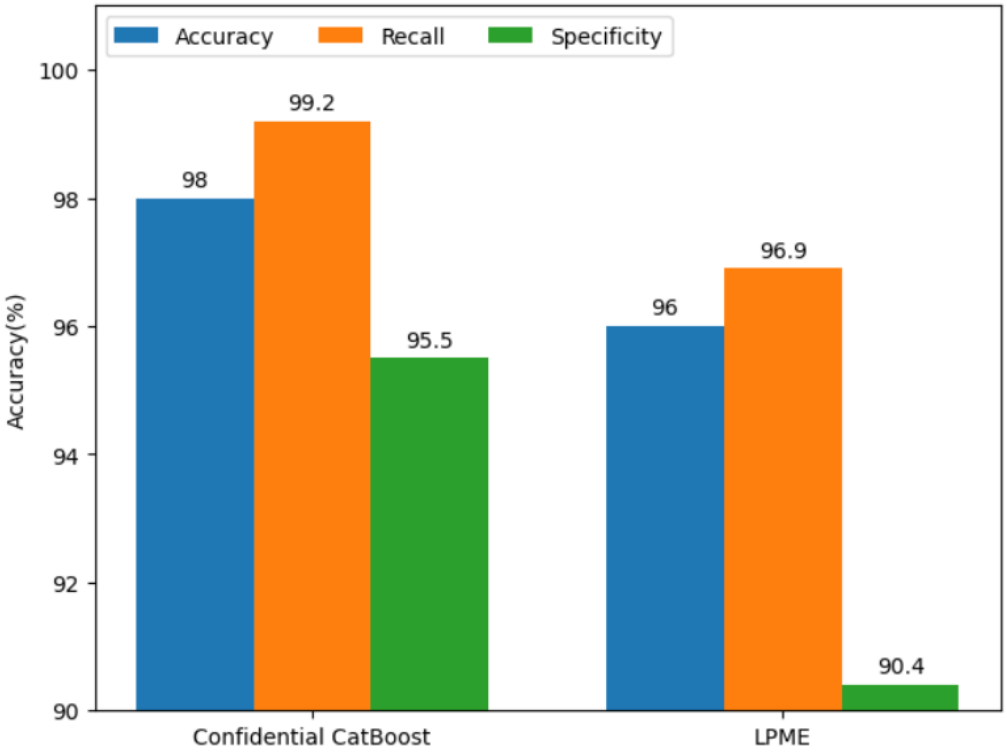

Figure 4: Accuracy of LPME |n|=1024 bits, N=3, K=5, h=3

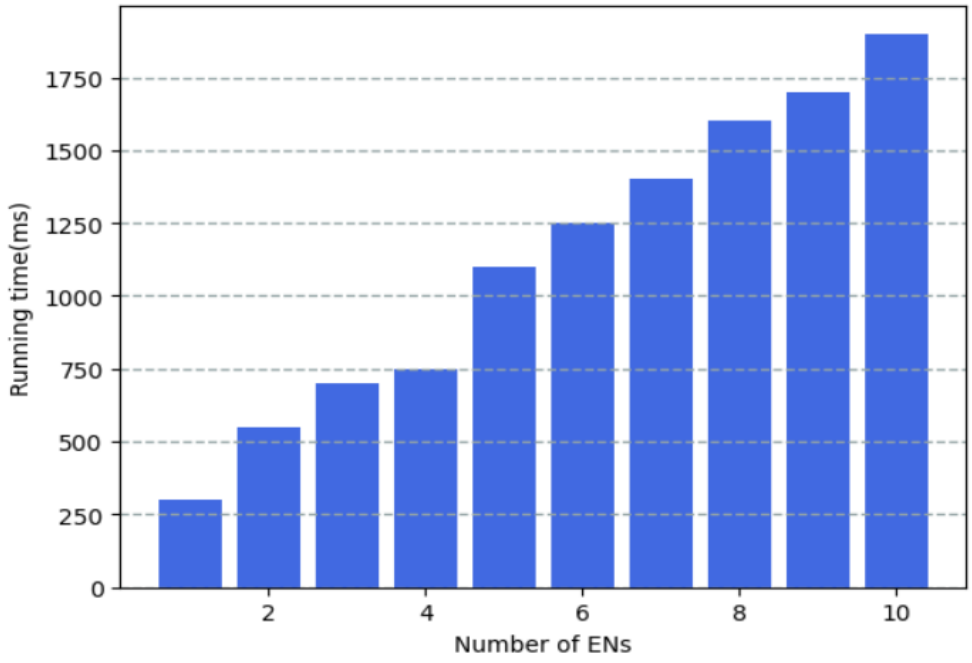

Figure 5: Running Time |n|=1024 bits, N=3, K=5, h=3

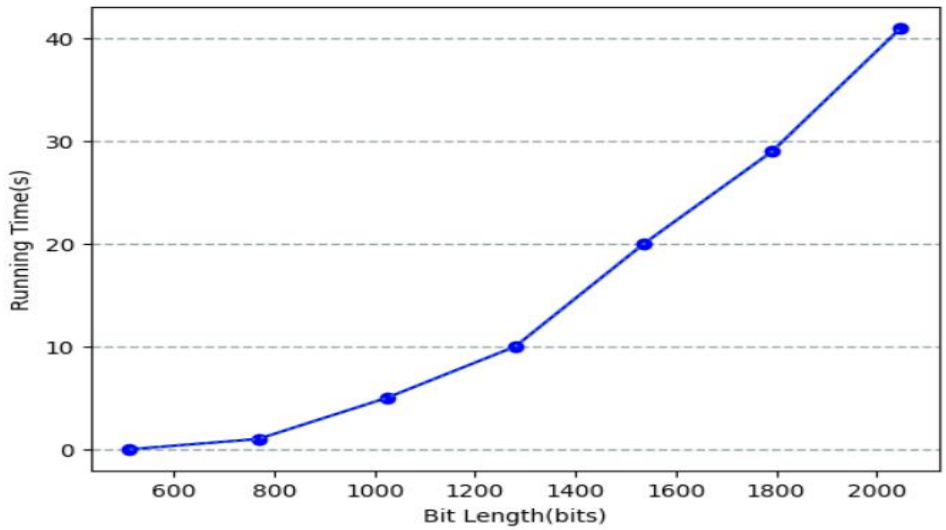

Figure 6: Running Time N=3, K=1, h=3

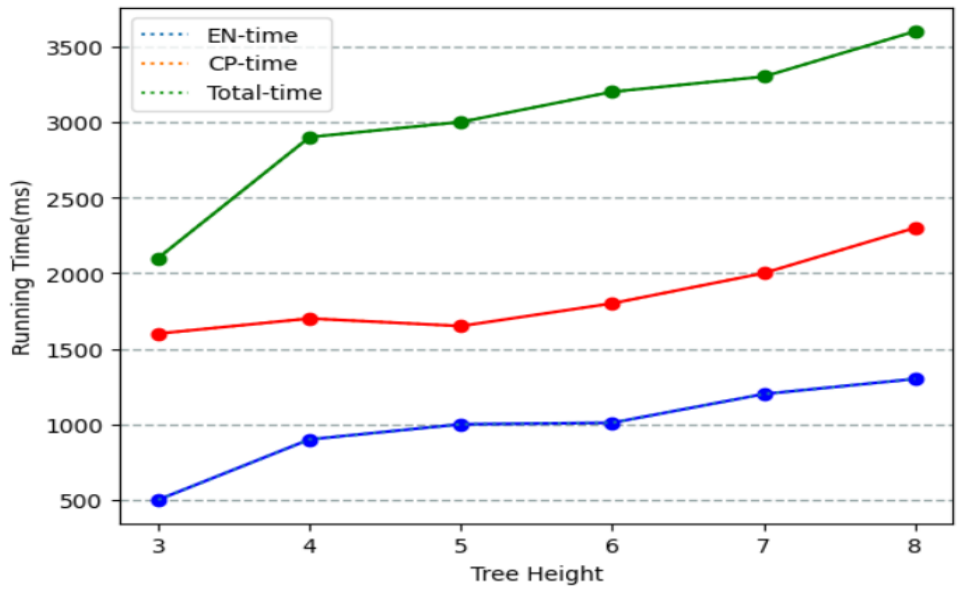

Figure 7: Running TimeN=3, K=1, h=3

Comparative Analysis

To demonstrate the LPME system and (29), we create a thorough comparison analysis of the (29), which, in contrast to our approach, is a distributed learning framework that safeguards privacy and safely learns over encrypted training data. Figure 4 shows the accuracy and figure 5-7 shows the running time for various bit lengths.

- Our system's accuracy increases with the number of trees in the tree; for example, when K is 14, it is 85.4 percent accurate for heart disease and 95.3 percent accurate for thyroid disease, while at K is 1, it is 90.6 percent accurate for heart disease and 97.1 percent accurate for thyroid disease.
- Our system has a small accuracy difference when compared to the original CatBoost, which is implemented over the global dataset, with an accuracy difference of less than 1% between the two datasets. When K= 14 is used, the accuracy is 94.6 percent (a 0.3% improvement over the dataset for heart disease) and 99.1 percent (a 0.1% improvement over the dataset for thyroid disease) when compared to the original CatBoost.

Conclusion

In this study, an edge-based confidential Catboost framework is developed that can both offer secure CatBoost across edge nodes with strong privacy protections and enable real-time, privacy-preserving medical diagnosis. The suggested system's secure computing may build the CatBoost model securely with minimal overhead and effectively deliver medical diagnoses without leaking personal information. The Harris Hawks Optimization reveals the best newly generated dataset required. The overall accuracy value was used to prove that it provides the best output compared to other strategies. The time efficiency was accurate in the proposed model. The model addresses the private network active attacks. The security and efficacy of the system on edge computing were validated by real-world dataset experiments.

Abbreviations

Nil

Acknowledgment

The authors would acknowledge the Mepco Schlenk Engineering College for providing resources to carry out the research work.

Author contributions

All the authors are equally contributed.

Conflict of interest

The authors declare that they have no conflict of interest.

Ethics approval

This work does not need ethical approval.

Funding

This work does not receive any funds to carry out the work.

References

1. Wang X, Ma J, Miao Y, Liu X, and Yang R. Privacy-preserving diverse keyword search and online pre-diagnosis in cloud computing. IEEE Transactions on Services Computing. 2019[cited 16 December 2019];15(2):710-723.
2. Zhang Y, Xu C, Li H, Yang K, Zhou J, and Lin X. HealthDep: An efficient and secure deduplication scheme for cloud-assisted eHealth systems. IEEE Transactions on Industrial Informatics. 2018[cited 2 May 2018];14(9):4101–4112.
3. Fu B, Liu P, Lin J, Deng L, Hu K, and Zheng H. Predicting invasive disease-free survival for early-stage breast cancer patients using follow-up clinical data. IEEE Transactions on Biomedical Engineering. 2018[cited 22 November 2018];66(7): 2053–2064.
4. Galletta A, Carnevale L, Bramanti A, and Fazio M. An innovative methodology for big data visualization for telemedicine. IEEE Transactions on Industrial Informatics. 2018[cited 31 May 2018];15(1): 490–497.
5. Kononenko I. Machine learning for medical diagnosis: History, state of the art and perspective. Artificial Intelligence Medicine. 2001:23(1):89–109.
6. Friedman C P, Wong A K, and Blumenthal D. Achieving a nationwide learning health system. Sci Transl Med. 2010[cited 10 November 2010];2(57):57cm29.
7. Miao Y, Liu X, Choo K K R, Deng R H, Wu H, and Li H. Fair and dynamic data sharing framework in cloud-assisted internet of everything. IEEE Internet Things. 2019[cited 6 May 2019];6(4):7201–7212.
8. Miao Y, Tong Q, Choo K K R, Liu X, Deng R H, and Li H. Secure online/offline data sharing framework for cloud-assisted industrial Internet of Things. IEEE Internet Things. 2019[cited 14 June 2019];6(5):8681–8691.
9. Miao Y, Weng J, Liu X, Choo K K R, Liu Z, and Li H. Enabling verifiable multiple keywords search over encrypted cloud data. Information Sciences.2018; 465:21–37.
10. Dai P, Liu K, Wu X, Xing H, Yu Z, and Lee V C. A learning algorithm for real-time service in vehicular networks with mobile edge computing. IEEE International Conference on Communications. 2019[cited 20-24 May 2019];1–6.
11. Wang F, Cong Z, Feng W, *et al.* Intelligent edge-assisted crowd cast with deep reinforcement learning for personalized QoE. IEEE Conference on Computer Communications. 2019[cited 17 June 2019].
12. Lin CC, Deng DJ, Chih YL, and Chiu HT. Smart manufacturing scheduling with edge computing using multi-class deep Q network. IEEE Transactions on Industrial Informatics. 2019[cited 29 March 2019];15(7): 4276–4284.
13. Aujla GS, Chaudhary R, Kaur K, Garg S, Kumar N, and Ranjan R. SAFE: SDN-assisted framework for edge–cloud interplay in secure healthcare ecosystem. IEEE Transactions on Industrial Informatics. 2018[cited 24 August 2018];15(1):469–480.
14. Sayeed MA, Mohanty SP, Kougianos E, and Zaveri HP. eSeiz: An edge-device for accurate seizure detection for smart healthcare. IEEE Transactions on Consumer Electronics. 2019[cited 30 May 2019];65(3): 379–387.
15. Rahmani AM, Gia TN, Negash B, *et al.* Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Generations Computer Systems. 2018;78:641–658.
16. Ogunleye AA and Qing-Guo W. XGBoost model for chronic kidney disease diagnosis. IEEE Transactions on Computational Biology and Bioinformatics. 2019[cited 17 April 2019];17(6):2131-2140.
17. Nishio M, Nishizawa M, Sugiyama O, *et al.* Computer-aided diagnosis of lung nodule using

gradient tree boosting and Bayesian optimization. PloS One. 2018[cited 19 April 2018];13(4).
18. Rao A R and Clarke D. A fully integrated open-source toolkit for mining healthcare big-data: Architecture and applications. IEEE International Conference on Healthcare Informatics. 2016[cited 8 December 2016]; 255–261.
19. E Union. General data protection regulation. U S D of Health and H Services, Health insurance portability and accountability act 1996.
20. Maryam Farhadi, Hisham M and Haddad M. Static Analysis of HIPPA Security Requirements in Electronic Health Record Applications. IEEE International Conference on Computer Software and Application. 2018[Cited 22 June 2018].
21. Miao Y, Ximeng L, Robert HD, *et al.* Hybrid keyword-field search with efficient key management for industrial Internet of Things. IEEE Transactions on Industrial Informatics. 2018[cite 21 October 2018]; 15(6):3206–3217.
22. Miao Y, Ma J, Liu X, Li X, Liu Z, and Li H. Practical attribute-based multi-keyword search scheme in mobile crowdsourcing. IEEE Internet Things. 2017[cited 4 December 2017];5(4): 3008–3018.
23. Miao Y, Ximeng L, Kim-Kwang RC, *et al.* Privacy-preserving attribute-based keyword search in shared multi-owner setting. IEEE Transactions on Dependable and Secure Computating. 2019[cited 5 February 2019];18(3):1080-1094.
24. Paillier P. Public-key cryptosystems based on composite degree residuosity classes. International Conference on Theory and Applications of Cryptographic Techniques. 1999[cited 15 April 1999].
25. Liu X, Deng R H, Yang Y, Tran H N, and Zhong S. Hybrid privacy-preserving clinical decision support system in fog-computing computing. Future Generation Computer Systems. 2018[cited January 2018];78(2): 825-837.
26. Pratibha Chaudry, Ritu Gupta, Abhilasha Singh. Analysis and Comparison of Various Fully Homomorphic Encryption Techniques. International Conference on Computing, Power and Communication Technologies(GUCON). 2019[cited 27 December 2019].
27. Ma Z, Ma J, Miao Y, and Liu X. Privacy-preserving and high accurate outsourced disease predictor on random forest. Information Sciences. 2019[cited September 2019]; 496:225–241.
28. Liu X, Choo K K R, Deng R H, Lu R, and Weng J. Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans Dependable and Secure Computing. 2016[cited 1 March 2016];15(1): 27–39.
29. Liu X, Deng R H, Choo K K R, and Weng J. An efficient privacy-preserving outsourced calculation toolkit with multiple keys. IEEE Transactions on Information Forensics and Security. 2016[cited 27 May 2016];11(11):2401–2414.
30. Wang Q, Minxin D, Xiuying C, et al. Privacy-preserving collaborative model learning: The case of word vector training. IEEE Transactions on Knowledge and Data Engineering. 2018[cited 26 March 2018]; 30(12): 2381–2393.
31. Mohassel P and Zhang Y. SecureML: A system for scalable privacy-preserving machine learning. IEEE Symposium on Security and Privacy. 2017[cited 26 June 2017].
32. Lindell Y and Pinkas B. Privacy-preserving data mining. 20 th Annual International Cryptology Conference on Advances in Cryptology. 2000[cited July 2000].
33. Sanil A P, Karr A F, Lin X, and Reiter J P. Privacy preserving regression modelling via distributed computation. Proceedings on the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004[cited August 2004]; 677–682.
34. Fu A, Chen Z, Mu Y, Susilo W, Sun Y, and Wu J. Cloud-based outsourcing for enabling privacy-preserving large-scale non-negative matrix factorization. IEEE Transactions on Services Computing. 2019[cited 28 August 2019]; 15(1):266-278.
35. Zainab Hikmat Mahmood, Mahmood Khalel Ibrahem. New Fully Homomorphic Encryption Scheme Based on Multistage Partial Homomorphic Encryption Applied in Cloud Computing. International Conference on Information and Sciences (AiCIS). 2019[cited 14 February 2019].
36. Al-Rubaie M and Chang J M. Privacy-preserving machine learning: Threats and solutions. IEEE Security and Privacy. 2019[cited 29 March 2019];17(2):49-58.
37. Cheng K, Fan T, Jin Y, Liu Y, Chen T, and Yang Q. SecureBoost: A lossless federated learning framework. arXiv: 1901.08755.2019. 2019[cited 25 January 2019].
38. Li T, Huang Z, Li P, Liu Z, and Jia C. Outsourced privacy preserving classification service over encrypted data. Journal of Networks and Computer Applications. 2018[cited 15 March 2018];106: 100–110.
39. Phong L T, Aono Y, Hayashi T, Wang L, and Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Security. 2017[cited 29 December 2017]; 13(5):1333–1345.
40. Wang Z, Song M, Zhang Z, Song Y, Wang Q, and Qi H. Beyond inferring class representatives: User-level privacy leakage from federated learning. Proc IEEE Conf Comput Commun, 2019[cited 17 June]; 2512–2520.
41. Zhang H and Zeng K. Pairwise Markov Chain: A task scheduling strategy for privacy-preserving sift on edge. Proc IEEE Conf Comput Commun, 2019; 1432–1440.
42. Ni J, Zhang K, Lin X, and Shen X S. Securing fog computing for Internet of Things applications: Challenges and solutions. IEEE Commun Surveys Tuts. 2017[cited 12 October 2017];20(1): 601–628.
43. Ma Z, Ma J, Miao Y, *et al.* PMKT: Privacy-preserving multi-party knowledge transfer for financial market forecasting. Future Gener Comput Syst, 2020[cited May 2020];106: 545–558.
44. Chen T and Guestrin C. XGBoost: A scalable tree boosting system. in Proc. ACM SIGKDD Int Conf Knowl Discov Data Mining. 2016[cited 13 August 2016]; 785–794.
45. Pinkas B, Schneider T, Smart N P, and Williams S C. Secure two-party computation is practical. Proc Int

Conf Theory Appl Cryptology Inf Secur. 2009;18: 250–267.
46. Zhuoran Ma, Jiafeng Ma, Yinblin Miao, Ximeng Liu. Lightweight Privacy-Preserving Medical Diagnosis in Edge Computing. IEEE Transactions on Services Computing. 2020[cited 24 June 2020];15(3):1606-1618.
47. Malathy Navaneetha Krishnan, Revathi Thiyagarajan. Multi-objective task scheduling in fog computing using improved gaining sharing knowledge-based algorithm. Concurrency and Computation: Practice and Experience.2022;34:1-22.
48. Malathy Navaneetha Krishnan, Revathi Thiyagarajan. Entropy-based complex proportional assessment for efficient task scheduling in fog computing. Transactions on Emerging Telecommunications Technologies.2023;23:2.
49. Malathy Navaneetha Krishnan, Revathi Thiyagarajan. Opposition-based Improved Memetic Algorithm for Placement of Concurrent IoT Applications in Fog Computing. Transactions on Emerging Telecommunications Technologies.2024.