electronics

MDPI

Article

Privacy-Preserving K-Nearest Neighbors Training over Blockchain-Based Encrypted Health Data

Rakib Ul Haque [1], A S M Touhidul Hasan [2,3,*], Qingshan Jiang [3] and Qiang Qu [3,4]

1 School of Computer Science & Technology, University of Chinese Academy of Sciences, Shijingshan District, Beijing 100049, China; rakibulhaqueraj@mails.ucas.ac.cn
2 Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh
3 Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; qs.jiang@siat.ac.cn (Q.J.); qiang@siat.ac.cn (Q.Q.)
4 Huawei Cloud Blockchain Lab, Huawei, Shenzhen 518055, China
* Correspondence: touhid@uap-bd.edu; Tel.: +880-1819698279

Received: 31 October 2020; Accepted: 23 November 2020; Published: 9 December 2020

Abstract: Numerous works focus on the data privacy issue of the Internet of Things (IoT) when training a supervised Machine Learning (ML) classifier. Most of the existing solutions assume that the classifier's training data can be obtained securely from different IoT data providers. The primary concern is data privacy when training a *K*-Nearest Neighbour (*K*-NN) classifier with IoT data from various entities. This paper proposes secure *K*-NN, which provides a privacy-preserving *K*-NN training over IoT data. It employs Blockchain technology with a partial homomorphic cryptosystem (PHC) known as Paillier in order to protect all participants (i.e., IoT data analyst *C* and IoT data provider *P*) data privacy. When *C* analyzes the IoT data of *P*, both participants' privacy issue arises and requires a trusted third party. To protect each candidate's privacy and remove the dependency on a third-party, we assemble secure building blocks in secure *K*-NN based on Blockchain technology. Firstly, a protected data-sharing platform is developed among various *P*, where encrypted IoT data is registered on a shared ledger. Secondly, the secure polynomial operation (SPO), secure biasing operations (SBO), and secure comparison (SC) are designed using the homomorphic property of Paillier. It shows that secure *K*-NN does not need any trusted third-party at the time of interaction, and rigorous security analysis demonstrates that secure *K*-NN protects sensitive data privacy for each *P* and *C*. The secure *K*-NN achieved 97.84%, 82.33%, and 76.33% precisions on BCWD, HDD, and DD datasets. The performance of secure *K*-NN is precisely similar to the general *K*-NN and outperforms all the previous state of art methods.

Keywords: blockchain; internet of things data; homomorphic cryptosystem; supervised learning; privacy protection

1. Introduction

At present, smart cities include more innumerable superior IoT infrastructures [1] to manage their component efficiently [2]. A tremendous volume of information accumulated from numerous IoT devices stationed in different city areas, such as medical health, agriculture, transportation, and energy transmission [3]. A large volume of reforms prompted by ML technology were proposed to handle the issues emerging from processing obligations of IoT data [4]. *K*-means [5] and *K*-NN [6] are pre-eminent unsupervised and supervised learning models, respectively, that can effectively implement data classification amid all ML models [7]. Therefore, these ML models have been used in various specialties to answer real-world classification dilemmas in IoT-enabled smart health. Considering the synopsis of individual fitness and healthcare records observed by wearable IoT sensors, Refs. [8–10] could be

fed to a *K*-NN classifier for fitness analysis. *K*-NN clustering classifier is also deployed in the field of network interference apprehension to recognize abnormalities from a group of traffic data derived from interactions among IoT devices [11].

The structure of supervised ML classifiers such as *K*-NN or Support Vector Machine (SVM) is denoted as the training stage, which trains a specific classifier to find underlying patterns with a defined output variable. The higher the significance of training samples increased, the more the performance of the ML classifier enhanced [12,13]. An individual such as a network provider or a hospital entity often owns the dataset required for training. It is generally bounded in courses of sample distinctiveness and amount. It is necessary to train the ML model classifiers with an efficient and effective mechanism using a unification of sample sets collected from various entities. Various entities are generally opposed to sharing their datasets for training as there are many safety concerns in terms of ownership, integrity, and data privacy:

1. Most of the training phases manipulate intimate data samples such as medical data reported from clinical wearable IoT devices, resulting in the leakage of private or sensitive and confidential information at the time of training tasks.
2. Latent invaders may cause unauthorized modification of data records by altering or tampering at the time of the data sharing process, resulting in an inaccurate classification of the ML model.
3. The data provider may lose authority, and replication of the shared datasets may occur as datasets are available to the associates.

To secure the data privacy issue of individual data providers, most of the present solutions [14–17] focused on cryptography and differential privacy. Those solutions assumed that the data required for training could be obtained securely from various data providers to classify and analyze. Issues of ownership and data integrity were focused on trivially. However, solutions are invalid due to potential attacks, for most cases, in reality. This paper uses the Blockchain technology to build a reliable data-sharing terrace, which can cover the gap between realistic confinement and typical prediction. In general, a shared filing scheme intended to permit the distribution of tamper-proof records among various individuals is called a Blockchain [18]. Auditing is enabled on Blockchain for immutable records, which confirms the ownership of recorded data.

Some works are focused on privacy-preserving *K*-NN computation, search, query, and classification [19–28]. None of these works used Homomorphic encryption with Blockchain in order to secure sensitive information. To consolidate Blockchain into ML training method is laborious but encouraging. The first difficulty is to outline a suitable training data format convenient for adjusting on Blockchain to secure each data provider's privacy. The second difficulty is to develop the training algorithm that establishes an accurate *K*-NN classifier using the Blockchain's recorded data and secures sensitive information. secure SVM [29] was proposed to address the problems mentioned earlier. In the proposed method, they employed a Blockchain-based, privacy-preserving training algorithm for the SVM using encrypted data of IoT devices from Smart Cities. A public-key cryptosystem is applied by secure SVM to protect the privacy of the data, which are encrypted by the private keys of data providers. However, secure SVM requires too many calculations, comparison, time, and space in order to analyze the health data. In most medical health research, *K*-NN outperformed SVM in terms of performance and time complexity [30–32].

To handle the above challenges, we propose secure *K*-NN, a privacy-preserving *K*-NN schema based on Blockchain and encrypted data of IoT devices. A public-key cryptosystem called Paillier has been employed to shield the IoT data privacy, which is encrypted by the own private key of the respected data providers. Paillier is an additive homomorphic cryptosystem (HC), which is very efficient in terms of time complexity for encryption and decryption than any other algorithm, such as Rabin, RSA, and Goldwasser-Micali [33]. Handling encrypted data could be a problem because of the tremendous amount of intercommunications. However, *K*-NN can handle these circumstances as there is no separate optimization algorithm. *K*-NN has essential procedures such as polynomial

operations and comparison. We design secure building blocks SPO (addition and subtraction), SC, and SBO by using the homomorphic properties of Paillier for secure *K*-NN. With the above building blocks, all iterations of secure *K*-NN do not need any trusted third party at the time of interaction, which significantly lessens the risk of a data breach.

The main contribution of this paper is as follows.

- To establish protected and trustworthy IoT data sharing, Blockchain technology is employed. All the IoT data are encrypted locally by the own private key of the respected data provider. The encrypted data are recorded on a Blockchain by uniquely formatted transactions.
- We designed protected building blocks, such as SPO (addition, subtraction), SBO, and SC using the PHC, i.e., Paillier, and developed a secure *K*-NN training algorithm. There is no requirement for a trusted third-party.
- Rigorous analysis has been done to prove that the secure *K*-NN can protect data privacy at the time of training, achieve similar accuracy as general *K*-NN and outperform all the previous state of the art method.

The rest of the paper is articulated as follows. Related work and preliminaries are discussed in Sections 2 and 3, respectively. The system overview is presented in Section 4. Section 5 summarized the proposed method. Analyzation of confidentiality issues and evaluation of the proposed scheme are explained in Sections 6 and 7, respectively. Finally, in Section 8, the paper is concluded.

2. Related Work

Supervised learning contains two phases: the training phase, where the ML model learns from a given set of labeled specimens, and the classification phase, where labels are the result for a given sample with maximum possibility. Thus, current research on privacy-preserving ML can be divided into two sections, namely privacy-preserving ML training and privacy-preserving ML classification.

2.1. Privacy-Preserving ML Training

In most cases, multiple parties are involved when training an ML model, which results in the privacy issue of the IoT data. The main goal is to protect the data provider's IoT data from being discovered by others at the time of training an ML model. During the last decade, numerous work have been done on this category [15,34–40] and our work focus here.

There are many methods used to secure data privacy in the publishing stage [41–43], but the most common approach is Differential privacy (DP) [15]. It assures the protection of published data by combining vigilantly computed distress to the fundamental data. The DP-based deep learning method was proposed by Abadi [15]. They developed a system to jointly train a neural network with preserving the sensitive information of their datasets. DP-based solutions can achieve more excellent computational performance. These solutions can execute calculations over plain text data. There are some limitations also: Firstly, due to perturbations, the quality and integrity of the training data are reduced significantly. Secondly, each training data's sensitive information is publicly exhibited, so only disruptions are not enough to effectively secure data privacy. The privacy budget parameter is inversely proportional to the model accuracy but directly proportional to data privacy protection.

ML training achieves reliability and provides privacy guarantee on encrypted IoT data using homomorphic encryption, and it allows calculations on ciphertext and preserves the correctness of the data. In order to train various ML models, different protected methods based on HC have been proposed, such as SVM [29,34], Logistic Regression [35,36], Decision Tree [38], and Naive Bayes [37]. Secure protocols [34] have been developed for secure addition and subtraction depending on Paillier, which results in a secure SVM training algorithm. Due to the computational limitation of Paillier, the authors developed an authorization server which worked as a trusted third-party.

PHC method can reach higher data privacy with more efficiency than the Fully homomorphic encryption (FHE) system. The PHC is much more practical than FHE for the addition and

multiplication operations. Complicated calculations can be employed with a trusted third-party [34]. Without that, the model will be inaccurate due to the approximation of complex equations with an individual computational operation [39,40]. On the other hand, the calculations of FHE is costly in terms of time and space complexity. Therefore, existing uses of FHE is prohibitive from the scenario of encryption and forecast. As a result, they are unrealistic in terms of application.

2.2. Privacy-Preserving ML Classification

Usually, two different parties interact in a classification as a service scenario. One holds the data sample, and the other holds the ML model. It is not safe to reveal sensitive data to an unreliable ML model owner for a data owner who wants to know the classification result. On the other hand, the model owner may decline to share the classification result as the asset value is too high for the service provider.

Some existing solutions are [14,17,44–46] for developing an effective solution in order to secure the privacy of both parties. A method was proposed by Wang et al. [45] to classify encrypted images based on multi-layer learning. The authors used a public classifier but considered that the image data should be secure. A privacy-preserving nonlinear SVM method was proposed by Zhu et al. [46] for online-based medical prediagnosis. The authors can protect both individual information of the health record and the SVM model with their design. Rahulamathavan et al. [17] proposed a privacy-preserving SVM data classification system. It can securely classify multiclass datasets. In this schema, the client input data samples are anonymous to the server. At the same time, clients are also unaware of the server-side classifier during the classification process. A group of classification protocols was developed using the HC techniques for employing ML's simple classifiers on encrypted data, such as Naive Bayes, hyperplane decision, and decision trees [14,44].

All the above studies employed standard ML classifiers and developed building blocks to assemble a privacy-preserving classification method. The calculations at the time of training an ML classifier are much complex compared to the classification phase. Those building blocks might be useless due to the complexity of the training algorithm.

2.3. The Novelty of This Paper

Earlier research on secure *K*-NN focused on any specific domain such as data confidentiality, secure query, secure search, secure computation, and secure classification [19–28]. None of them keep track of all the transactions, and most of them considered that the *K*-NN model is already trained. In this study, a partial cryptosystem known as Paillier is employed with Blockchain technology to handle the issues related to ownership, integrity, and data privacy at the time of training *K*-NN classifiers based on the data from various data providers. To be specific, all IoT data from individual data providers is encrypted using Paillier then registered on a distributed ledger. Any data analyst can obtain encrypted data by interacting with the respective data provider and train the *K*-NN classifier. The data analysts can never obtain the plaintext of IoT data on the Blockchain. The secure protocol of operation in *K*-NN is developed to conduct training tasks with encrypted data, i.e., SPO (addition/subtraction), SBO, and SC. A privacy-preserving *K*-NN training algorithm, secure *K*-NN, is proposed based on this building block, and there is no need for a trusted third party. The secure *K*-NN can train *K*-NN classifiers without the loss of accuracy as the training is based on Paillier.

Two well-known security definitions are used as security goals: secure two-party computation [47] and modular sequential composition [48]. The propose method illustrates that the individual data provided is inadequate to acquire any information about other data provider's data. Simultaneously, the model parameters of data analysts are secure from the knowledge of any data providers during the training process.

3. Preliminaries

This section describes all notations, background ideas, and related technologies of this research.

3.1. Notation

A dataset D, which consists of m records, where x_i and y_i is the i-th record in D and l_i is the label of the corresponding x_i and y_i. Define d and (c_{x_i}, c_{y_i}) as two relevant parameters of K-NN. In this paper, we use a PHC named Paillier as the cryptosystem, and let $[[m]]$ represent the encryption of message under Paillier. Notations are summarized in Table 1.

Table 1. Notations.

Signs	Interpretations	Signs	Interpretations
D	dataset	d and (c_{x_i}, c_{y_i})	model parameters
d	distance	(c_{x_i}, c_{y_i})	initial centroid
x_i, y_i	i^{th} record in dataset	t	threshold
$\phi(N)$	euler phi-function	l_i	class label
m	size of the dataset D	$[[m]]$	the encryption of
A	Labeled data's array	-	m under Paillier

3.2. Homomorphic Cryptosystem

Cryptosystems are mainly based on three algorithms: Generation of key (*KeyGen*), data encryption (*Enc*), and data decryption (*Dec*). In public-key cryptosystems, a pair of keys (*PK; SK*) is used, such as for encryption and decryption public key (*PK*) and private key (*SK*) are used respectively. A cryptosystem property, which can map the operations over ciphertext to the corresponding plaintext without being aware of the decryption key, is known as Homomorphic. Definition 1 describes the homomorphic property of the cryptosystem.

Definition 1. (*homomorphic [33]*) *A method of public-key encryption (Gen, Enc, Dec) can be homomorphic only if for all n and all (PK; SK) output by Gen* (1^n)*, it is possible to define groups* $\mathbb{M}, \mathbb{C}$ *(depending on PK only) such that:*

1. $\mathbb{M}$ and $\mathbb{C}$ are the message space and all ciphertexts outcome respectively by Enc_{pk} are elements of $\mathbb{C}$.
2. $Dec_{sk}(o(c_1, c_2)) = \sigma(m_1, m_2)$ is held for any $m_1, m_2 \in \mathbb{M}$, any c_1 output by $Enc_{pk}(m_1)$, and any c_2 output by $Enc_{pk}(m_2)$.

A partial homomorphic cryptosystem known as Paillier is being used in the proposed schema. It is a public key cryptography method with the partial homomorphic property as it allows only two operations, secure addition and subtraction. Let p and q are n-bit primes, $N = pq$. N and $(N, \phi(N))$ (Let, $N > 1$ be an integer. Then Z_N^* is an abelian group under multiplication modulo N. Define $\phi(N) \underline{\underline{def}} \mid Z_N^* \mid$, the order of the group Z_N^*.) are the public key and private key, respectively. $c := [[(1+N)^m r^N mod N^2]]$ is the encryption function in Paillier, where $m \in \mathbb{Z}_N$ and $m := [[\frac{[c^{\phi(N)} mod N^2 - 1]}{N} \times \phi(N)^{-1} mod N]]$ is the decryption function. More details about Paillier is explained in [32].

3.3. K-Nearest Neighbors (K-NN)

K-nearest neighbors (*K*-NN) [6] clustering is a type of supervised ML algorithm, used for classification and predictive regression problems. There is no specific training phase and uses whole training data [49] during classification because of that, it is called a lazy learning algorithm. It does not assume anything about the underlying data. Distance d needs to be calculated in order to find the designated centroid (c_{x_j}, c_{y_j}). There are different methods to find the distance in *K*-NN algorithm, i.e., Euclidean distance d_e (Equation (1)), Manhattan distance d_m (Equation (2)), Cosine

distance d_c (Equation (3)), etc methods. In this study, we will use Manhattan distance d_m. Let, $(x_1, y_1), (x_2, y_2)....., (x_m, y_m) \in D$. Algorithm 1 illustrate the entire process.

$$d_e = \sqrt{(c_{x_j} - x_i)^2 + (c_{y_j} - y_i)^2} \tag{1}$$

$$d_m = |(c_{x_j} - x_i)| + |(c_{y_j} - y_i)| \tag{2}$$

$$d_c = \frac{(c_{x_j} \times x_i) + (c_{y_j} \times y_i)}{\sqrt{x_i^2 + c_{x_j}^2} \times \sqrt{y_i^2 + c_{y_j}^2}} \tag{3}$$

Algorithm 1: Basic *K*-NN

Input: $D = \{(x_1, y_1),...,(x_m, y_m)\}$, threshold t,$A = \{(c_{x_k}, c_{y_k})\}$, [initially $k = 1$]
Output: labeled $D = \{(l_1, x_1, y_1),...,(l_m, x_m, y_m)\}$
while $i = 1$ *to* m **do**
  while $j = 1$ *to* k **do**
    Compute d_{m_j} by Equation (2);
  end
  Identify the minimum d_{m_j};
  Increase k by 1;
  if $d_{m_j} > t$ **then**
    Put (x_i, y_i, l_k) to A;
  end
  else if $d_{m_j} \leq t$ **then**
    Put (x_i, y_i, l_j) to A;
  end
end

3.4. Blockchain System

Blockchain is a public and shared ledger, consists of a list of blocks. It is developed in cryptocurrency systems for registering transactions such as Bitcoin. It ensures secure transactions among untrusted participants. Various Blockchain platforms are: Ethereum, HyperLedger, etc., have been employed in different real-life sectors. According to the access restriction of users in Blockchain, Blockchain platforms are classified into consortium Blockchains, private Blockchains, and public Blockchains.

There are various advantages of Blockchain:

- Decentralized: It is developed on a peer-to-peer network as a shared ledger, and there is no requirement of a trusted third-party.
- Tamper-proof: Consensus protocols are employed by Blockchain, such as Proof-of-Work (PoW). Thus, Data manipulation is impractical.
- Traceability: The rest participants can easily verify the transactions between two parties in a Blockchain system.

Despite having many advantages, Blockchain has the vulnerability of data privacy to skilled attackers. Initially, all transactions are registered as plain texts in blocks, which exposes the transaction's vital information to other participants, and adversaries [50]. Therefore, privacy and security issues must be handled cautiously when using Blockchain in terms of data sharing platform.

4. Problem Description

This segment illustrates the issues of secure *K*-NN training across encrypted IoT data accumulated from various parties, which includes the system design, threat type, and design purposes.

4.1. System Design

A data flow IoT ecosystem is developed and shown in Figure 1, including IoT devices, data providers, the Blockchain platform, and data analysts.

- ZigBee, 3^{rd} generation (3G)/4^{th} generation (4G), and Wireless Fidelity (WiFi) are examples of the wired or wireless network through which IoT devices can sense and transmit valuable information, including medical data, smart cities, etc. In this study, due to the lack of computational capabilities, IoT devices will not participate in the data sharing and analysis processes.
- Data providers gather all the data from IoT devices within their range. All the data comprises sensitive information, so all the data are encrypted using partially homomorphic encryption by the data provider and registered in a Blockchain.
- To gather the encrypted IoT data from all data providers, the Blockchain-based IoT platform serves as a distributed database, where protocols are maintained, and all data are recorded in a shared ledger. The built-in consensus mechanism ensures the sharing of IoT data in a secure and tamper-proof way.
- IoT data analysts intend to get a rooted perspicacity within the data registered in the Blockchain-based platform by using the existing analyzing techniques. Data analysts will obtain encrypted data from corresponding data providers in order to train the *K*-NN classifiers.

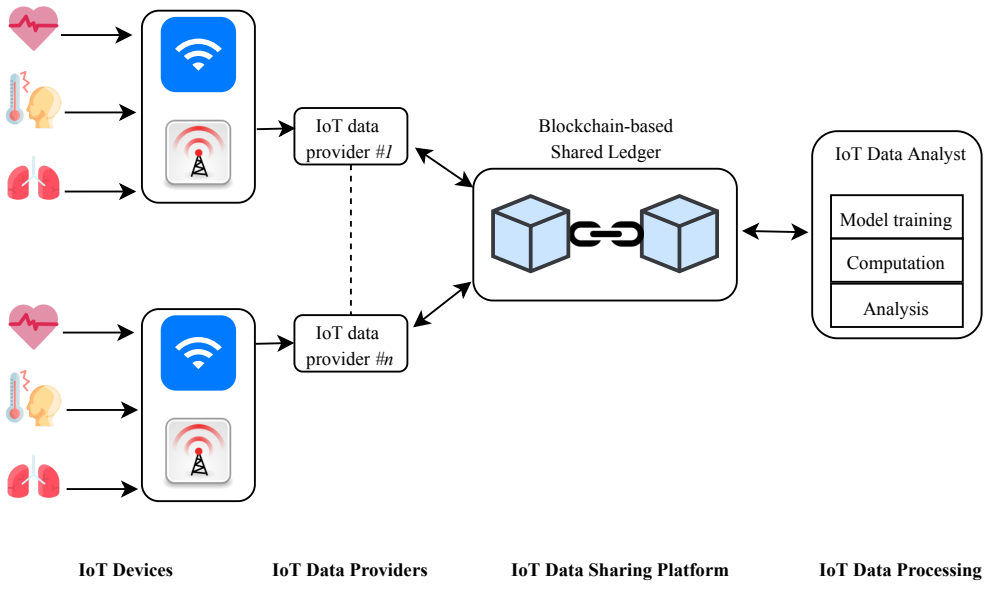


Figure 1. System model of a data-driven IoT ecosystem.

4.2. Threat Type

Various latent threats exist over individual entities and at the time of their interactions, according to the system model description in Figure 1. This study focuses on the threats related to data privacy throughout the interaction between the data provider and data analysts. It is assumed that the data analyst is a curious but honest foe. To be more specific, the data analyst is honest for maintaining the protocol of predesigned ML training and curious regarding the contents of the data. Moreover, the data analyst strives to acquire further knowledge by analyzing the intermediate data at the time of computation on encrypted data.

The subsequent models of threat are considered based on the data analyst's collected vital information with various attack inclinations mostly employed in the literature [51,52]. On the other hand, the data provider may also try to identify the data analyst's model parameter from the intermediate data.

- Recognized Ciphertext Model. The data analyst can merely obtain the encrypted IoT data registered in the Blockchain Platform. The IoT data analysts can record intermediate outputs when training the secure algorithm, such as iteration steps.
- Recognized Background Model. The IoT data analyst expects to know more further details of shared data. However, from the shared ciphertext model, an IoT data analyst may gather more information by using her previous knowledge. To be more specific, the IoT data analyst can conspire with distinct IoT data providers to infer the sensitive information of other participants.

4.3. Design Purposes

Consider more than one IoT data provider and data analyst conspire to steal other participant's privacy. Assume that all the participants as a curious-but-honest foe who executes protocol honestly but has an interest in other's private information. Any number of participants may conspire with each other. The proposed method aims to shield the individual participant's privacy and securely train the *K*-NN classifiers. The security goals are as follows:

- At the time of encountering curious-but-honest foe, the data analyst and individual data provider's data are protected from disclosure.
- At the time of encountering more than one parties conspire with each other, the data analyst and individual data provider's privacy also will be protected from disclosure.

5. The Construction of Secure *K*-NN

This section illustrates the system specifications of the proposed privacy-preserving *K*-NN training method over block-chain-based encrypted IoT data.

5.1. System Overview

For clearness, consider that a data analyst intents to train *K*-NN classifier based on the data gathered from various IoT data providers. Figure 2 illustrates the system overview, where the individual data provider preprocessed IoT data instances, encrypts them locally using their own private key, and register those encrypted data in the Blockchain-based distributed ledger. Present key supervision mechanisms [53–55] can be applied to handle the encryption abilities of data providers. The IoT data analyst can train a *K*-NN classifier by collecting the encrypted data registered in the public ledger and erect a protected algorithm with the building blocks of SPO, SBO, and SC. At the time of the training process, it is essential to interact between IoT data analyst and IoT data provider for reciprocating intermediate outcomes.

However, it is essential to mention that many comparison tasks are necessary to perform the training on an *K*-NN model. To accomplish the comparison task on encrypted data is extremely expensive, costly, and time-consuming. On the other hand, accurate intermediate data cannot be shared because the parameters of *K*-NN algorithm are easy to guess. There is a high possibility that in this situation, data analysts can be successful at guessing the original data. Therefore, to reduce the algorithm's complexity, to make the method more realistic and protected from the privacy breach of both data providers and data analysts, we introduce a SBO. The data provider adds a small amount of bias (δ) to secure the data's privacy at the time of sharing the intermediate data. This small amount of discrimination does not cause any significant change in the classification process.

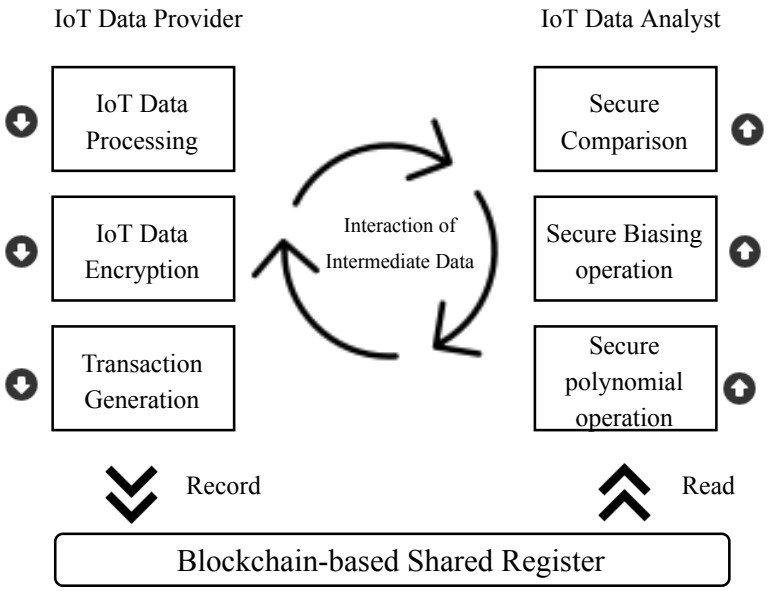


Figure 2. System overview of Secure *K*-NN.

5.2. Encrypted Data Sharing via Blockchain

To aid model training, without the sacrifice of generality, consider that the same training task's data instances have been locally preprocessed and designated with the corresponding feature vectors [16].

A unique transaction arrangement is defined in order to save the encrypted IoT data in the Blockchain. The proposed transaction structure primarily consists of two fields: input and output.

The input terminal comprises:

- The address of the data provider
- The encrypted version of data
- Name of the IoT device from where the data is generated

The corresponding output terminal holds:

- The address of the data analyst
- The encrypted version of data
- Name of the IoT device from where the data is generated

Hash value will be the addresses of the data provider and data analyst, and the encrypted data is determined from the homomorphic encryption, i.e., Paillier. Depending on the consideration that the length of the private key is 128 bytes, the length of the individual encrypted data instance is set to 128 bytes and stored in the Blockchain. The segment length of the IoT device type is 4 bytes.

The node serving as the data provider in the Blockchain network broadcasts it in a P2P system after assembling a new transaction, where the miner nodes will validate the correctness of the operation. A specific miner node can package the transaction in a new block and adding the block to the existing chain using current consensus algorithms, i.e., the PoW mechanism. Multiple transactions can be registered in a single block.

5.3. Building Blocks

Section 4 already specified that the goal is to secure the privacy of various IoT providers and design a privacy-preserving algorithm for training *K*-NN models over multiple private datasets afforded by diverse IoT providers.

5.3.1. *K*-NN

Several methods are available in order to calculate the distance, which is a model parameter of the *K*-NN. In this research, Manhattan distance d_m (Equation (2)) is considered due to its simplicity in calculation. Algorithm 1 illustrate the entire process.

5.3.2. Secure Polynomial Operations (SPO)

In the proposed secure *K*-NN training schema, we develop secure polynomial addition and subtraction to securely train the *K*-NN model using the homomorphic property of Paillier's. Reliable additions, secure subtractions and multiplication can be achieved straightforward. The homomorphic properties in Paillier can be defined as: $[[m_1 + m_2]] = [[m_1]] \times [[m_2]]\ (modN^2)$, and the homomorphic properties in case of subtraction can be described as:$[[m_1 - m_2]] = [[m_1]] \times [[m_2]]^{-1}\ (modN^2)$. $[[m^{-1}]]$ is known as the modular multiplicative inverse, which can calculate $[[m]] \times [[m]]^{-1}\ (modN^2) = 1$ in Paillier. $\phi(N)$ function can compute $[[m]]^{-1}$, $[[m]]^{-1} = [[m]]^{\phi(N)-1}$. Therefore, using ciphertext manipulation, the secure polynomial multiplication can be achieved, as shown in Equation (4).

$$[[am_1 + bm_2]] = [[m_1^a]] \times [[m_2^b]](modN^2) \tag{4}$$

Similarly, the secure polynomial division can be achieved, as shown in Equation (5).

$$[[m_1/a + m_2/b]] = [[m_1^{a^{-1}}]] \times [[m_2^{b^{-1}}]](modN^2) \tag{5}$$

However, this research need only secure polynomial addition and subtraction. Thus, the secure polynomial addition and subtraction are statistically indistinguishable, as Paillier is statistically indistinguishable [33].

5.3.3. Secure Biasing Operations (SBO)

In line no. 6 of Algorithm 2, data provider *P* calculate distance $[[d_{m_j}]]$ using SPO. Next step is to send the encrypted distance $[[d_{m_j}]]$ to the data analyst *C*. If the data provider *P* sent the encrypted distance $[[d_{m_j}]]$ to the data analyst *C* then *C* will decrypt it and try to guess the private data of the data provider. There is a high possibility of success as the data analyst *C* initiated the clustering point. Therefore, to protect the privacy of the data provider *P*'s data at the time of training the *K*-NN algorithm, we introduce the secure biasing operation (SBO). *P* will add a small amount of bias δ using SPO before sending the data to *C* in order to protect the data privacy. The bias δ will be unknown to *C*.

Algorithm 2: Secure Comparison

1 ***P's* Input:** $D = \{m_1, m_2\}$, Bias δ, $[-3 \leq \delta \leq -1, \delta \neq 0, 1 \leq \delta \leq 3]$
2 ***C's* Input:** Public key PK, Private key SK
3 ***P's* Output:** $flag$
4 P computes $[[m_1 + \delta]]$ and $[[m_2 + \delta]]$ by SPO;
5 P send $[[m_1 + \delta]]$ and $[[m_2 + \delta]]$ to C;
6 C decrypts and compares $[[m_1 + \delta]]$ and $[[m_2 + \delta]]$;
7 **if** $([[m_1 + \delta]]) \geq ([[m_2 + \delta]])$ **then**
8 | C send flag 0 to P;
9 **end**
10 **else**
11 | C send flag 1 to P;
12 **end**

In this study, the range of bias will depend on the coefficient of variation (CV), where $CV = \frac{Standard Deviation}{Mean} = \sigma/\bar{x}$. Standard Deviation and Mean is computed using, $\sigma = \sqrt{\frac{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}{n-1}}$ and $\bar{x} = \sum x_i/n$ respectively. Here n is the total number of data and x_i stands for each data of the data-set. CV value greater than or equal to one, means that the data is scattered. CV value less than one, means that the is not scattered. Therefore, at the time of the experiment, various values for bias were chosen. The range was within $[1 \leq \delta \leq 5]$, when $CV < 1$ and $[-5 \leq \delta \leq -1]$, when $CV >= 1$. We found that bias range $[1 \leq \delta \leq 3]$ and $[1 \leq \delta \leq 3]$, gives proper classification results, when CV>=1 and CV<1, respectively. Therefore, If the coefficient of variation is greater than or equal to 1, $CV \geq 1$, than set the range of δ to $[-3 \leq \delta \leq -1]$ and On the other hand, if the coefficient of variation is less than 1, $CV < 1$, than set the range of δ to $[1 \leq \delta \leq 3]$. Note that the value of δ will never be equal to ($\delta == 0$).

5.3.4. Secure Comparison (SC)

The secure comparison in the proposed method is illustrated as the comparison between two encrypted numbers $[[m_1]]$ and $[[m_2]]$. For participants A and B engage in the secure comparison algorithm, neither parties can obtain real information. Our secure comparison algorithm protocol is exhibited in Algorithm 2, and the security proof is described in Section 6.

Proposition 1. *(Security of Secure Comparison Algorithm). Algorithm 2 is secure in the curious-but-honest model.*

To develop SBO, we use the secure polynomial operation (mainly addition and subtraction) based on the homomorphic property of Paillier's. A small amount of bias δ (where $[-3 \leq \delta \leq -1, \delta \neq 0$, $1 \leq \delta \leq 3]$) is encrypted and added with d_{m_j} by data provider P using SPO (addition/subtraction). However, the data analyst C will never be able to extract the exact value of the bias δ value, or of the range, and this small amount of bias does not affect the classification task, which is shown in the performance evaluation Section 7. SBO ensures the privacy of the data provider P. If δ is positive, the definition will be:

$$[[m_1]] \times [[\delta]](modN^2) = [[m_1 + \delta]]$$

Again, when δ is negative, the definition will be:

$$[[m_1]] \times [[\delta]]^{-1}(modN^2) = [[m_1 - \delta]]$$

5.4. Training Algorithm of Secure K-NN

For protected optimum design parameters, we outline a privacy-preserving K-NN training algorithm. Assume there is a single IoT data analyst C and n number of data providers P. Algorithm 3 specifies the training algorithm for secure K-NN. In Algorithm 3, the K-NN model parameters and sensitive data of IoT data providers are confidential. At the time of facing any collusion or curious-but-honest adversaries, individual members cannot infer any vital information of another member from the algorithm's execution process's intermediate outcomes. Section 6 illustrates the security proofs for Algorithm 3.

Proposition 2. *(Security of Privacy-Preserving K-NN Training Algorithm). Algorithm 3 is secure in the curious-but-honest model.*

Algorithm 3: Secure *K*-NN Training Algorithm

***P's* Input:** $D = \{(x_1, y_1),...,(x_m, y_m)\}$, Bias δ, $[-3 \leq \delta \leq -1, \delta \neq 0, 1 \leq \delta \leq 3]$
***C's* Input:** threshold t, a pair of keys (PK_c, SK_c), A = $\{(c_{x_k}, c_{y_k})\}$, [initially $k = 1$]
***C's* Output:** labeled $D = \{(l_1, x_1, y_1),...,(l_m, x_m, y_m)\}$
C initializes (c_{x_1}, c_{y_1});
while $u = 1$ *to* n **do**
  C sends $[[c_{x_1}]]$ and $[[c_{y_1}]]$ to P_u;
  while $i = 1$ *to* m **do**
    while $j = 1$ *to* k **do**
      P_u computes $[[d_{m_j}]]$ by SPO and SC and $[[d_{m_j} + \delta]]$ by SBO ;
    end
    P_u send $[[d_{m_j} + \delta]]^u$ to C;
    C decrypts $[[d_{m_j} + \delta]]^u$ and identify the minimum $(d_{m_j} + \delta)^u$;
    C increase k by 1;
    if $(d_{m_j} + \delta) > t$ **then**
      C Put (x_i, y_i, l_k) to A;
    end
    else if $(d_{m_j} + \delta) \leq t$ **then**
      C Put (x_i, y_i, l_j) to A;
    end
  end
end

6. Security Analysis

The security analysis manifests in this section under the identified ciphertext model and the public background model. Two security definitions were followed: secure two-party computation [47] and modular sequential composition [48]. A satisfied, protected two-party calculation protocol is safe in the face of curious-but-honest foes, and modular sequential composition implements a way to develop secret protocols in a modular way. Security proof of the proposed algorithm is described based on these two definitions.

6.1. Background of Security Proof

The notation in the article [14] was followed: Let, F is computed by a protocol π and $F = (F_A, F_B)$ be a polynomial function; Input of A's and B's are a and b respectively using π, desire to calculate $F(a, b)$; A's view is the tuple $view_A^\pi(\lambda, a, b) = \lambda; a; m_1, m_2, ..., m_n$ where $m_1, m_2, ..., m_n$ are the message received by A at the time of execution. B's view is defined in a similar manner. $output_A^\pi(a, b)$ and $output_B^\pi(a, b)$ are the outputs of A and B respectively. π's global outcome is $output^\pi(a, b) = (output_A^\pi(a, b), output_B^\pi(a, b))$.

Definition 2. *(Secure Two-Party Computation [47]). If for all possible inputs (a, b) and simulators S_A and S_B holds the following properties ($\approx$ denotes computational indistinguishability against probabilistic polynomial-time adversaries with the negligible advantage in the security parameter λ.), only then a protocol π privately computes f with statistical security:*

$$\{S_A, f_2(a, b)\} \approx \{view_A^\pi(a, b), output^\pi(a, b)\}$$

$$\{f_1(a, b), S_B\} \approx \{output^\pi(a, b), view_B^\pi(a, b)\}$$

The sequential modular composition's fundamental idea is that: protocol π is run to call an ideal functionality F by n participants, e.g., to calculate F privately, A and B send their inputs to a trusted third-party and receive the result. If the secure two-party computation is satisfied by the protocol π, and the same functionality as F can be achieved by protocol ρ privately, then the protocol of ρ in π can replace the ideal protocol for the functionality F; Then, the latest protocol π^{ρ} is protected and safe under the curious-but-honest model [14,48].

Theorem 1. *(Modular Sequential Composition [37]). Let the two-party probabilistic polynomial time functionalities be $F_1, F_2, ..., F_n$ and $F_1, F_2, ..., F_n$ is calculated by the protocols $\rho_1, \rho_2, ..., \rho_n$ in the presence of curious-but-honest adversaries. Let, two-party probabilistic polynomial time functionality be G and G is securely computed in the $F_1, F_2, ..., F_n$ by a protocol π - hybrid model in the presence of curious-but-honest adversaries. Then, $\pi^{\rho_1,\rho_2,...,\rho_n}$ securely calculate G in the presence of curious-but-honest adversaries.*

6.2. Security Proof for Secure Comparison

Two entities are involved in Algorithm 2: P and C. The function is

$$F : F([[m_1 + \delta]]_C, [[m_2 + \delta]]_C, PK_C, SK_C,) = (\phi, (m_1 \geq m_2))$$

Proof of Proposition 1. The view of P is

$$view_P^{\pi} = ([[m_1 + \delta]]_C, [[m_2 + \delta]]_C, PK_C)$$

Hence, the simulator:

$$S_P^{\pi}((m_1, m_2); F(m_1, m_2)) = view_P^{\pi}([[m_1]]_C, [[m_2]]_C, [[\delta]]_C, PK_C)$$

where $[[m_1]]_C$ and $[[m_2]]_C$ are encrypted by PK_C and the confidentiality of $[[m_1]]_C$ and $[[m_2]]_C$ are equivalent to the Paillier cryptosystem. Therefore, P cannot infer the value directly.

The view of C is

$$view_C^{\pi} = (([[m_1 + \delta]]), ([[m_2 + \delta]]), PK_C, SK_C)$$

Then, S_C^{π} runs as follows:

$$F((m_1 + \delta), (m_2 + \delta)) \quad = view_C^{\pi}((m_1 + \delta), (m_2 + \delta), PK_C, SK_C)$$

The bias δ is unknown to C, for that reason C would never be able to get the real m_1 and m_2 from $(m_1 + \delta)$ and $(m_2 + \delta)$. After comparison bewteen $(m_1 + \delta)$ and $(m_2 + \delta)$, C will return a flag with a value 0, if $(m_1 + \delta) \geq (m_2 + \delta)$, other wise value of flag will be 1. C is honest in following the method's protocols. Therefore, C would never infer the value directly. □

6.3. Security Proof for Secure K-NN Training Algorithm

IoT data providers P and an IoT data analyst C, are the roles involved in Algorithm 2. Individual IoT data providers function in the same manner. Every one of the data providers will meet the security requirements if we can prove that one of them meets the security requirements. The function is:

$$F : F(D_{P_u}, PK_C, SK_C) = (\phi, (\{(l_1, x_1, y_1), ..., (l_m, x_m, y_m)\})).$$

Proof of Proposition 2. Individual IoT data provider's P view is

$$view_P^{\pi} = (D_{P_u}, ([[c_{x_k}]], [[c_{y_k}]]), PK_C)$$

where $[[c_{x_k}]]$ and $[[c_{y_k}]]$ are encrypted by PK_C, the confidentiality of $[[c_{x_k}]]$ and $[[c_{y_k}]]$ will be equivalent to the cryptosystem Paillier. So none of the IoT data providers can infer the value directly.

The view of C is

$$view_C^{\pi} = ((d_{m_j} + \delta), (c_{x_k}, c_{y_k}), PK_C, SK_C)$$

Now, the confidentiality of $(d_{m_j} + \delta)$ needs to be discussed, i.e., whether the IoT data analyst can predict the private data of individuals IoT data providers from the value. Clearly, the value is no-solution for the unknown $x_i and y_i$. The IoT data analyst may try to calculate unknown $x_i and y_i$ using the known distance $(d_{m_j} + \delta)$ and centroid (c_{x_k}, c_{y_k}). It is not possible for IoT data analysts to identify the point of IoT data providers because the distance is added with bias value and the data analyst has no idea about the bias value or its range. Even with the brute force cracking, it is not possible to get the real value of dataset D. Consider that individual IoT data provider consists of a small dataset, which is $2-$dimensional, 100 instances, and each dimension is 32 bits (Typically, 4 bytes (32-bit) memory space is occupied by single-precision floating-point). Under this situation, the probability of IoT data analyst successful guessing is $\frac{1}{2^{(n \times 6400)}}$, which is a negligible success probability [33]. □

We obtain the security of Algorithm 3 using modular sequential composition, as SPO, SBO, and SC are used in Algorithm 3, so it is secure in the curious-but-honest model.

7. Performance Evaluation

In this section, the performance of secure K-NN is evaluated based on efficiency and accuracy through extensive analysis using real-world dataset. Firstly, experiment settings are described, and the effectiveness and efficiency are demonstrated by the experimental results.

7.1. Experiment Setup

This segment discusses the testbed, dataset, and all other tasks for data preprocessing and experimental environment.

7.1.1. Testbed

In the proposed model, individual IoT data providers gather all the data in their domain from the IoT devices and perform data encryption. The experiments are executed on MacBook Pro equipped with an Intel Core i5 processor (2.5 GHz), memory (4 GB 1600 MHz DDR3), serving as IoT data providers and IoT data analysts simultaneously. The SPO, SBO, SC, and the Secure K-NN are implemented in the Platform: Google's Colaboratory; Language: Python 3; Browser: Google Chrome.

7.1.2. Dataset

This study uses three real-world datasets, namely Breast Cancer Wisconsin Data Set (BCWD), Heart Disease Data Set (HDD), and Diabetes Data Set (DD) [56,57]. These datasets are publicly available from the UCI machine learning repository. The features of BCWD resembled a digitalized image of a breast mass and described characteristics of the cell nuclei present in the image. Each of the data instances is labeled as benign or malignant. The HDD and DD contain 13 and 9 numeric attributes, respectively. Instances are classified based on the types of heart diseases and diabetes symptoms. Table 2 represents the statistics of the dataset. We run 10-fold cross-validation to avoid overfitting or contingent results, and average results are recorded. 80% of the data has been selected for the model training and the remaining 20% for testing.

Table 2. Statistics of Datasets.

Datasets	Instances Number	Attributes Number	Discrete Attributes	Numerical Attributes
BCWD	699	9	0	9
HDD	303	13	13	0
DD	768	9	0	9

7.1.3. Float Format Conversion

The general *K*-NN training algorithms perform on both integer and floating-point numbers based on the data set. However, all the operations of cryptosystems are done on integers. Therefore for safety purposes, we should perform format conversion into an integer representation. Let, D is a binary floating-point number, which is represented as, $D = (-1)^s \times M \times 2^E$, according to the global standard IEEE 754 [where the sign bit is s, significant number is M and exponential bit E]. A data analyst may perform this format conversion during the implementation of secure *K*-NN based on the dataset type.

7.1.4. Key Length Setting

The security of the public key cryptosystem is closely associated with the length, and a compact key may cause vulnerable encryption. A long key reduces the homomorphic operation's efficiency, and a too-short key may cause the overflow of plaintext space during the homomorphic operations (i.e., the secure polynomial operation and secure biasing operation) on the ciphertext. Therefore, it is crucial to consider the length of the key in order to bypass the possibility of overflow. The Key of Paillier cryptosystem N is set to 1024−bit in secure *K*-NN.

7.2. Evaluation Parameters

We use three commonly used criteria for evaluating ML classifiers (Accuracy (6), Precision (7) , Recall (8)).

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{6}$$

$$Precision = \frac{t_p}{t_p + f_p} \tag{7}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{8}$$

where t_p is the number of relevant (the positive class) that are labeled precisely, f_p is the numbers of irrelevant (the negative class) that are labeled correctly, f_n is the numbers of relevant that are mislabeled and t_n is the number of irrelevant that are mislabeled in the test outcomes.

The general *K*-NN was implemented using raw Python language in order to demonstrate that secure *K*-NN does not lessen the accuracy upon preserving the individual IoT data provider's privacy and securely training the classifier. The main focus is to train the classifier securely. For that reason, the train parameters are not adjusted, and default parameters are used. The results of precision and recall are summarized in Table 3.

Table 3. Summary of performance.

Parameter	Model	Datasets		
		BCWD	**HDD**	**DD**
Accuracy	SVM	96.60%	81.00%	77.00%
	SecureSVM	95.25%	80.89%	76.67.00%
	K-NN (t = 8)	96.96%	83.50%	79.00%
	Secure *K*-NN (t= 8)	97.80%	82.33%	78.00%
Precision	SVM	96.16%	81.79%	75.00%
	SecureSVM	96.02%	81.25%	74.80%
	K-NN (t = 8)	96.54%	83.85%	77.00%
	Secure *K*-NN (t = 8)	96.26%	82.30%	76.00%
Recall	SVM	96.48%	80.38%	71.00%
	SecureSVM	95.65%	79.65%	70.91%
	K-NN (t = 8)	96.85%	83.85%	75.90%
	Secure *K*-NN (t = 8)	96.67%	82.66%	75.1%

The performance of Secure *K*-NN is almost similar to standard *K*-NN and has better performance than SVM [29]. However, the data provider must be careful about the bias value because a larger bias may reduce the classifier's performance. The proposed design shows better robustness on both (discrete attributes and numerical attributes) datasets.

7.3. Efficiency

7.3.1. Building Blocks Evaluation

Table 4 illustrates the running time of the SPO with encrypted datasets on Algorithm 3. Table 4 also gives the time consumption of IoT data providers *P* and data analysis *C*, the total time consumption.

Table 4. Performance of the building blocks in Secure *K*-NN against Secure SVM.

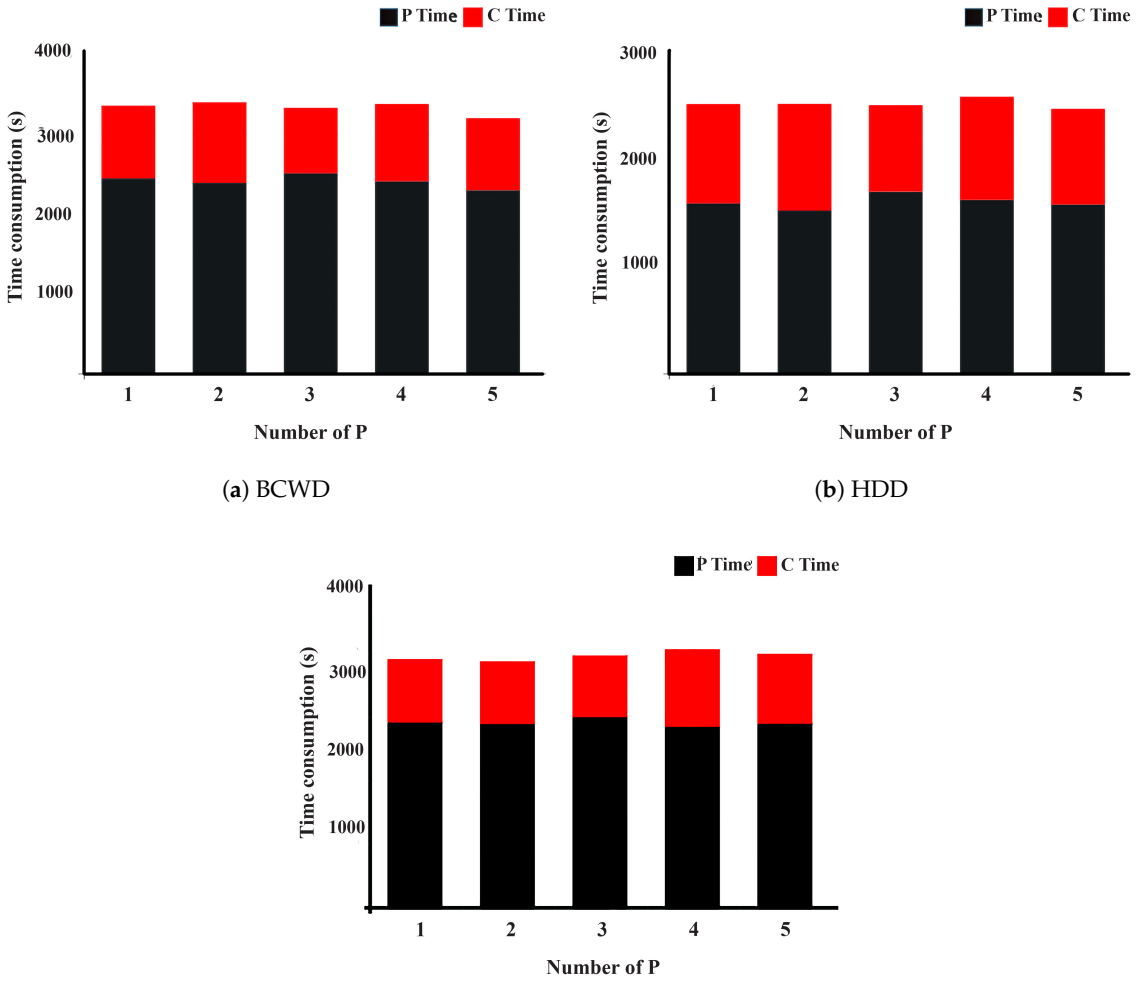
Dataset	Time	Secure SVM	Secure *K*-NN
BCWD	Total	3674 s	3357.2 s
	P	2789 s	2534 s
	C	1066 s	860 s
	SPO	3462 s	3113 s
HDD	Total	2735 s	2534 s
	P	1761 s	1520 s
	C	924 s	765 s
	SPO	2333 s	1922 s
DD	Total	3959 s	3709 s
	P	3199 s	2920 s
	C	1045 s	995 s
	SPO	3773 s	3527 s

According to the performance results in Table 4, secure *K*-NN spend less than an hour with encrypted dataset BCWD, HDD, and DD at the time of training, which is an acceptable time consumption as a stand-alone algorithm. It is essential to mention that the general *K*-NN is comparatively slower, so it is better not to train a *K*-NN algorithm with a larger dataset at a time. We recommend that the reader convert the larger dataset into small portions and train the secure *K*-NN. In our implementation, we used multi-threading in Python to control the run time of a larger dataset.

In this experiment, various *P* is simulated linearly. Therefore the *P* time shown in Table 4 is the accumulation of time consumed by different *P*. In a real-world application, various *P* can run their algorithms parallelly so that the time consumption of *P* and the total time consumption can be reduced. We believe Algorithm 3 to be useful for real-world sensitive applications. Confronting various datasets such as BCWD, DD (numerical attributes), and HDD (discrete attributes), secure *K*-NN manifests satisfying robustness in time consumption.

7.3.2. Scalability Evaluation

Secure *K*-NN believes that various IoT data providers are engaging and contributing data. We distribute the dataset into various identical sections to mimic the situation of different IoT data providers. We observe the fluctuations in time consumption to evaluate the scalability of the proposed scheme when various data IoT providers strive for the calculation. Cases are simulated during the number of IoT data providers rises from 1 to 5. The outcomes are represented in Figure 3. The *X*-axis represents the number of IoT data providers associated with the calculation, and the *Y*-axis represents the time consumption.



(**a**) BCWD (**b**) HDD

(**c**) DD

Figure 3. Time consumption of secure *K*-NN with different numbers of data providers *P*.

Theoretically, The time consumption of secure *K*-NN is proportional to the amount of data and number of iterations in the comparison portion. If the total amount of data and data quality are fixed, the rise in the number of *P* will not affect the time consumption, and the time consumption of *P* or *C* remains the same at different numbers of *P*. There is a small vibration in the total time consumption when the number of *P* rises from 1 to 5 because the program's run time gets disturbed as other host processes are used for the simulation.

8. Conclusions

This paper introduced a novel privacy-preserving *K*-NN training method called secure *K*-NN, which can handle data privacy and data integrity concerns. It employs Blockchain technology to train the algorithm in a multi-party scenario where the IoT data is received from various data providers. A partial homomorphic cryptosystem known as Paillier is applied to assemble an effective and reliable method. Efficiency and security of secure *K*-NN are demonstrated in this study. The proposed method achieves almost similar accuracy to general *K*-NN and outperforms the earlier state of the art. Future work includes developing a versatile structure that allows assembling a broad range of privacy-preserving ML training algorithms on a multi-party scenario with encrypted datasets.

Author Contributions: Data curation, R.U.H.; Funding acquisition, Q.J. and Q.Q.; Investigation, R.U.H.; Methodology, R.U.H. and A.S.M.T.H.; Project administration, A.S.M.T.H.; Software, R.U.H.; Supervision, A.S.M.T.H.; Validation, A.S.M.T.H., Q.J. and Q.Q.; Writing—original draft, R.U.H.; Writing—review & editing, R.U.H., A.S.M.T.H., Q.J. and Q.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is supported by Key-Area Research and Development Program of Guangdong Province under Grant No. 2019B010137002, and the National Key Research and Development Program of China under Grant No. 2020YFA0909100, the National Natural Science Foundation of China under Grants No. 61902385 and 61762062.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hasan, A.S.M.T.; Qu, Q.; Li, C.; Chen, L.; Jiang, Q. An Effective Privacy Architecture to Preserve User Trajectories in Reward-Based LBS Applications. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 53. [CrossRef]
2. Vongsingthong, S.; Smanchat, S. Internet of Things: A review of applications & technologies. *Suranaree J. Sci. Technol.* **2014**, *1*, 359–374.
3. Zhang, Y.; Yu, R.; Nekovee, M.; Liu, Y.; Xie, S.; Gjessing, S. Cognitive machine-to-machine communications: Visions and potentials for the smart grid. *IEEE Netw.* **2012**, *26*, 6–13. [CrossRef]
4. Provost, F.; Kohavi, R. On applied research in machine learning. *Mach. Learn. Boston* **1998**, *30*, 127–132. [CrossRef]
5. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]
6. Soucy, P.; Mineau, G.W. A simple *K*-NN algorithm for text categorization. In Proceedings of the 2001 IEEE International Conference on Data Mining IEEE, San Jose, CA, USA, 29 November–2 December 2001.
7. Barlow, H.B. Unsupervised learning. *Neural Comput.* **1989** , *1*, 295–311. [CrossRef]
8. Anliker, U.; Ward, J.A.; Lukowicz, P.; Troster, G.; Dolveck, F.; Baer, M.; Keita, F.; Schenker, E.B.; Catarsi, F.; Coluccini, L.; et al. AMON: A wearable multiparameter medical monitoring and alert system. *IEEE Trans. Inf. Technol. Biomed.* **2004**, *8*, 415–427. [CrossRef]
9. Baig, M.M.; Gholamhosseini, H. Smart health monitoring systems: An overview of design and modeling. *J. Med. Syst.* **2013**, *37*, 1–14. [CrossRef]
10. Lee, H.; Choi, T.K.; Lee, Y.B.; Cho, H.R.; Ghaffari, R.; Wang, L.; Choi, H.J.; Chung, T.D.; Lu, N.; Hyeon, T.; et al. A graphene-based electrochemical device with thermoresponsive microneedles for diabetes monitoring and therapy. *Nat. Nanotechnol.* **2016**, *11*, 556–572. [CrossRef]
11. Shen, M.; Wei, M.; Zhu, L.; Wang, M. Classification of encrypted traffic with second-order markov chains and application attribute bigrams. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1830–1843. [CrossRef]
12. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 843–852.
13. Shen, M.; Ma, B.; Zhu, L.; Mijumbi, R.; Du, X.; Hu, J. Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 940–953. [CrossRef]
14. Bost, R.; Popa, R.A.; Tu, S.; Goldwasser, S. Machine learning classification over encrypted data. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 23–26 February 2014.
15. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2016; pp. 308–318.
16. Wang, Q.; Hu, S.; Du, M.; Wang, J.; Ren, K. Learning privately: Privacy-preserving canonical correlation analysis for cross-media retrieval. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
17. Rahulamathavan, Y.; Phan, R.C.W.; Veluru, S.; Cumanan, K.; Rajarajan, M. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. *IEEE Trans. Dependable Secure Comput.* **2014**, *11*, 467–479. [CrossRef]

18. Li, H.; Zhu, L.; Shen, M.; Gao, F.; Tao, X.; Liu, S. Blockchain-based data preservation system for medical data. *J. Med. Syst.* **2018**, *42*, 141. [CrossRef]
19. Qi, Y.; Atallah, M.J. Efficient privacy-preserving k-nearest neighbor search. In Proceedings of the 28th International Conference on Distributed Computing Systems, Beijing, China, 17–20 June 2008; pp. 311–31.
20. Zhan, J.Z.; Chang, L.; Matwin, S. Privacy preserving k-nearest neighbor classification. *IJ Netw. Secur.* **2005**, *1*, 46–51.
21. Ni, W.; Gu, M.; Chen, X. Location privacy-preserving k nearest neighbor query under user's preference. *Knowl. Based Syst.* **2016**, *103*, 19–27. [CrossRef]
22. Rong, H.; Wang, H. M.; Liu, J.; Xian, M. Privacy-preserving k-nearest neighbor computation in multiple cloud environments. *IEEE Access* **2016**, *4*, 9589–9603. [CrossRef]
23. Songhori, E.M.; Hussain, S.U.; Sadeghi, A.R.; Koushanfar, F. Compacting privacy-preserving k-nearest neighbor search using logic synthesis. In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.
24. Wu, W.; Parampalli, U.; Liu, J.; Xian, M. Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments. *World Wide Web* **2019**, *22*, 101–123. [CrossRef]
25. Park, J.; Lee, D.H. Privacy preserving k-nearest neighbor for medical diagnosis in e-health cloud. *J. Healthc. Eng.* **2018**, *2018*. [CrossRef]
26. Yang, S.; Tang, S.; Zhang, X. Privacy-preserving k nearest neighbor query with authentication on road networks. *J. Parallel Distrib. Comput.* **2019**, *134*, 25–36. [CrossRef]
27. Xiong, L.; Chitti, S.; Liu, L. K nearest neighbor classification across multiple private databases. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, November 2006; New York, NY, USA; pp. 840–841. [CrossRef]
28. Zhang, F.; Zhao, G.; Xing, T. Privacy-preserving distributed k-nearest neighbor mining on horizontally partitioned multi-party data. In *International Conference on Advanced Data Mining and Applications*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 755–762.
29. Shen, M.; Tang, X.; Zhu, L.; Du, X.; Guizani, M. Privacy-Preserving Support Vector Machine Training Over Blockchain-Based Encrypted IoT Data in Smart Cities. *IEEE Internet Things J.* **2019**, *6*, 7702–7712. [CrossRef]
30. Huang, M.; Han, H.; Wang, H.; Li, L.; Zhang, Y.; Bhatti, U.A. A Clinical Decision Support Framework for Heterogeneous Data Sources. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1824–1833. [CrossRef] [PubMed]
31. Can, Y.S.; Chalabianloo, N.; Ekiz, D.; Ersoy, C. Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study. *Sensors* **2019**, *19*, 1849. [CrossRef] [PubMed]
32. Yin, H.; Jha, N.K. A health decision support system for disease diagnosis based on wearable medical sensors and machine learning ensembles. *IEEE Trans. Multi-Scale Comput. Syst.* **2017**, *3*, 228–241. [CrossRef]
33. Katz, J.; Lindell, Y. Introduction to modern cryptography. In *CRC Cryptography and Network Security Series*; CRC Press: Boca Raton, Florida, FL, USA, 2014.
34. FG-Serrano, F.-J.; N-Vzquez, A.; A-Martn, A. Training Support Vector Machines with privacy-protected data. *Pattern Recognit.* **2017** , *72*, 93–107. [CrossRef]
35. Cock, M.; Dowsley, R.; Nascimento, A.C.A.; Newman, S.C. Fast, privacy preserving linear regression over distributed datasets based on pre-distributed data. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, AISec '15*; ACM: New York, NY, USA, 2015; pp. 3–14.
36. Graepel, T.; Lauter, K.; Naehrig, M. Ml confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology—ICISC 2012*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 1–21.
37. Liu, X.; Lu, R.; Ma, J.; Chen, L.; Qin, B. Privacy-preserving patientcentric clinical decision support system on naive bayesian classification. *IEEE J. Biomed. Health Inform.* **2016**, *20*, 655–668. [CrossRef]
38. Vaidya, J.; Shafiq, B.; Fan, W.; Mehmood, D.; Lorenzi, D. A random decision tree framework for privacy-preserving data mining. *IEEE Trans. Dependable Secure Comput.* **2014**, *11*, 399–411. [CrossRef]
39. Aono, Y.; Hayashi, T.; Phong, L.T.; Wang, L. Privacy-preserving logistic regression with distributed data sources via homomorphic encryption. *IEICE Trans. Inf. Syst.* **2016**, *99*, 2079–2089. [CrossRef]
40. Aono, Y.; Hayashi, T.; P, L.T.; Wang, L. Scalable and secure logistic regression via homomorphic encryption. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, CODASPY '16*; ACM: New York, NY, USA, 2016; pp. 142–144.

41. Hasan, A.S.M.T.; Jiang, Q.; Chen, H.; Wang, S. A New Approach to Privacy-Preserving Multiple Independent Data Publishing. *Appl. Sci.* **2018**, *8*, 783. [CrossRef]
42. Hasan, A.S.M.T.; Jiang, Q.; Li, C. An Effective Grouping Method for Privacy-Preserving Bike Sharing Data Publishing. *Future Internet* **2017**, *9*, 65. [CrossRef]
43. Hasan, A.S.M.T.; Jiang, Q.; Luo, J.; Li, C.; Chen, L. An effective value swapping method for privacy preserving data publishing. *Secur. Comm. Netw.* **2016**, *9*, 3219–3228. [CrossRef]
44. De Cock, M.; Dowsley, R.; Horst, C.; Katti, R.; Nascimento, A.; Poon, W.; Truex, S. Efficient and Private Scoring of Decision Trees, Support Vector Machines and Logistic Regression Models based on PreComputation. *IEEE Trans. Dependable Secure Comput.* **2017**, *16*, 217–230. [CrossRef]
45. Wang, W.; Vong, C.; Yang, Y.; Wong, P. Encrypted image classification based on multilayer extreme learning machine. *Multidimens. Syst. Signal Process.* **2017**, *28*, 851–865. [CrossRef]
46. Zhu, H.; Liu, X.; Lu, R.; Li, H. Efficient and privacy-preserving online medical prediagnosis framework using nonlinear svm. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 838–850. [CrossRef] [PubMed]
47. Goldreich, O. *Foundations of Cryptography: Volume 2, Basic Applications*; Cambridge University Press: Cambridge, UK, 2009.
48. Canetti, R. Security and composition of multiparty cryptographic protocols. *J. Cryptol.* **2000**, *13*, 143–202. [CrossRef]
49. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
50. Gao, F.; Zhu, L.; Shen, M.; Sharif, K.; Wan, Z.; Ren, K. A Blockchain-based privacy-preserving payment mechanism for vehicleto-grid networks. *IEEE Netw.* **2018**, *32*, 184–192. [CrossRef]
51. Shen, M.; Ma, B.; Zhu, L.; Du, X.; Xu, K. Secure phrase search for intelligent processing of encrypted data in cloud-based iot. *IEEE Internet Things J.* **2018**, *6*, 1998–2008. [CrossRef]
52. Zhu, L.; Tang, X.; Shen, M.; Du, X.; Guizani, M. Privacy-preserving ddos attack detection using cross-domain traffic in software defined networks. *IEEE J. Selec. Areas Commun.* **2018**, *36*, 628–643. [CrossRef]
53. Du, X.; Guizani, M.; Xiao, Y.; Chen, H. A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 1223– 1229. [CrossRef]
54. Xiao, Y.; Rayi, V.K.; Sun, B.; Du, X.; Hu, F.; Galloway, M. A survey of key management schemes in wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2314–2341. [CrossRef]
55. Du, X.; Xiao, Y.; Guizani, M.; Chen, H.H. An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw.* **2007**, *5*, 24–34. [CrossRef]
56. Dheeru, D.; Karra, T.E. *UCI Mach Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2017.
57. Detrano, R.; Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Schmid, J.; Sandhu, S.; Guppy, K.H.; Lee, S.; Froelicher, V. International application of a new probability algorithm for the diagnosis of coronary artery disease. *Am. J. Cardiol.* **1989**, *64*, 304–310. [CrossRef]