 CMC, vol.58, no.3, pp.585-601, 2019

A Privacy-Preserving Algorithm for Clinical Decision-Support Systems Using Random Forest

Alia Alabdulkarim[1], Mznah Al-Rodhaan[2], Yuan Tian*,[3] and Abdullah Al-Dhelaan[2]

Abstract: Clinical decision-support systems are technology-based tools that help healthcare providers enhance the quality of their services to satisfy their patients and earn their trust. These systems are used to improve physicians' diagnostic processes in terms of speed and accuracy. Using data-mining techniques, a clinical decision support system builds a classification model from hospital's dataset for diagnosing new patients using their symptoms. In this work, we propose a privacy-preserving clinical decision-support system that uses a privacy-preserving random forest algorithm to diagnose new symptoms without disclosing patients' information and exposing them to cyber and network attacks. Solving the same problem with a different methodology, the simulation results show that the proposed algorithm outperforms previous work by removing unnecessary attributes and avoiding cryptography algorithms. Moreover, our model is validated against the privacy requirements of the hospitals' datasets and votes, and patients' diagnosed symptoms.

Keywords: Privacy-preserving, clinical decision-support system, random forests.

1 Introduction

Patient health records (PHRs) are considered a vital asset in all health institutions, such as hospitals and clinics. Physicians refer to PHRs during diagnosis and record their findings, such as lab results and X-ray images. Furthermore, when diagnosing a patient, a physician performs a procedure that is called "differential diagnosis", in which he or she lists all possible diseases and eliminates them one-by-one until only one disease is left [Ledley and Lusted (1959)]. To simplify this process, clinical decision-support systems (CDSSs) are used to enhance physicians' diagnostic process [Musen, Middleton and Greenes (2014)].

Since PHRs are now computerized, health institutions can use them to build a classification model for classifying a patient's new symptoms. A CDSS utilizes data-mining techniques or a knowledgebase to aid physicians during diagnosis [Berner and La Lande (2007)]. For further enhancement, hospitals may collaborate to generate a more accurate classification

[1] Information Technology Department, King Saud University, Kingdom of Saudi Arabia.
[2] Computer Science Department, King Saud University, Kingdom of Saudi Arabia.
[3] Nanjing Institute of Technology, China.
* Corresponding Author: Yuan Tian. E-mail: ytian@ksu.edu.sa.

model by combining their datasets. However, such collaboration will raise privacy concerns that are related to patients' records [Alabdulkarim, Al-Rodhaan and Tian (2017); Liang, Lu, Chen et al. (2011); Liu, Lu, Ma et al. (2016); Lu, Lin and Shen (2013)].

The privacy of shared data is a serious issue [Ma, Zhang, Cao et al. (2015); Rong, Ma, Tang et al. (2018); Xiong and Shi (2018)]; thus, privacy preserving algorithms for securing data-mining techniques enable the extraction of hidden patterns from a dataset without actually accessing it [Zhang, Tong, Tang et al. (2005)]. A privacy-Preserving naïve Bayes classifier (PPNBC) model was proposed in Liu et al. [Liu, Lu, Ma et al. (2016)]. The authors utilized a cloud to collect and aggregate encrypted PHRs and train the NBC. The privacy of the resulting model was maintained by a third party.

Various data-mining techniques have been used for diagnosing many types of diseases, including heart diseases [Abdar, Kalhori, Sutikno et al. (2015); Chaurasia and Pal (2013)], diabetes [Rahman and Afroz (2013)], and lung cancer [Krishnaiah, Narsimha and Chandra (2013)]. Comparing the accuracy rates, the study in Abdar et al. [Abdar, Kalhori, Sutikno et al. (2015)] has found that among SVM, KNN, and neural networks, C4.5 was the most accurate. Furthermore, comparing J48 and Bayesian networks, the study in Rahman et al. [Rahman and Afroz (2013)] found that the former has higher performance accuracy. Hence, decision trees are the most accurate classification models for medical applications.

In an ensemble of decision trees, instead of returning one decision class, each tree in the ensemble returns a decision class. Then, the class that receives the majority of votes is returned as the final decision class. The random forest ensemble is constructed by randomly generating multiple subsets of the dataset and from each dataset subset, a single decision tree is built [Wu, Feng, Naehrig et al. (2016)]. Furthermore, the studies in Alickovic et al. [Alickovic and Subasi (2016); Azar and El-Metwally (2013); Özçift (2011)] have shown the positive impact of tree ensembles on the classification performance and results.

Generating decision trees from multiple distributed datasets that are owned by different parties without disclosing their content is called privacy-preserving data-mining, in which privacy is achieved through encryption and secret-sharing schemes.

Motivated by the need for a privacy-preserving CDSS, that would mainly aim to build an accurate classification model to aid physicians while maintaining patients' health records private, in this paper, we propose a privacy-preserving clinical decision-support system (P-PCDSS) that uses random forests to enable multiple parties (hospitals), through a cloud, to collaborate in creating a classification model (random forest) for the CDSS without revealing patients' records. We have proposed in Alabdulkarim et al. [Alabdulkarim, Al-Rodhaan and Tian (2017)] a privacy-preserving model for a healthcare systems. Part of the proposed model addresses the privacy and security challenges faced by clinical decision-support systems. In this paper, PPCDSS model is based on the proposed privacy-preserving random forest (PPRF) algorithm. The study in Liu et al. [Liu, Lu, Ma et al. (2016)] has proposed a different solution to the same problem using PPNBC. The simulation results show that the PPRF algorithm outperforms the PPNBC algorithm in terms of average runtime in preserving hospital and patient privacy by removing unneeded attributes and avoiding the use of cryptography. Furthermore, our model is validated against all datasets and patient privacy

requirements. The main contributions of our work are threefold:

1. We propose a PPCDSS that helps hospitals create a classification model privately for the clinical decision-support system and diagnose new patients' symptoms without disclosing their datasets.

2. In order to protect patients' privacy, we propose a privacy-preserving random forest (PPRF) algorithm that builds a random forest model from distributed datasets without disclosing the datasets.

3. Our proposed model has a shorter average runtime than that of the work in Liu et al. [Liu, Lu, Ma et al. (2016)], which used NBC as a classification model.

The remainder of the paper is organized as follows: The system architecture is in Section 2. Then, a set of notations and preliminaries are presented in Section 3. The proposed PPRF algorithm is described in Section 4, followed by privacy analysis and performance evaluation in Sections 5 and 6, respectively. The current literature and related work are reviewed in Section 7. Finally, the conclusions and future work are presented in Section 8.

2 System architecture

2.1 System model

In our model, we build a random forest (RF) from all participating hospitals' datasets while preserving the patients' privacy. Fig. 1 depicts the layout of all system participants and their relations. The main participants in our PPRF model are the trusted authority (TA), the hospitals, and the cloud.

1. Trusted authority (TA): The TA is responsible for RSA key generation and management during initialization and setup.

2. Hospitals: Each hospital generates a local random forest ensemble and sends it to the cloud. In addition, they vote out trees from the global ensemble.

3. Cloud: The cloud receives the local ensembles from the participating hospitals and securely aggregates them into a single global ensemble.

2.2 Threat model

We define an adversary, $\mathcal{A}$, whose goal is to eavesdrop on the transmission between the hospitals, $\mathcal{H}$, and the cloud, $\mathcal{C}$. $\mathcal{A}$ may acquire the ensemble and/or the encrypted arrays of votes. By intercepting the latter, $\mathcal{A}$ may try to decipher the arrays to obtain hospitals' votes. Whereas, with the former, $\mathcal{A}$ may be able to reconstruct the original training datasets. However, we assume no collusion occurs between the system parties that would result in disclosing patients' personal health records since they are not part of the decision trees. Furthermore, we assume any hospital $\mathcal{H}$ will provide the cloud with legitimate ensembles.

Table 1: Notation

Symbol	Description
RF_h	An RF ensemble of hospital h
D	Dataset
$\|D\|$	Size of dataset D
d_i	A subset dataset from D
t_i	A decision tree that was generated from d_i

2.3 Privacy requirements

PHRs and patients' symptoms are the most important assets and should remain private. The following privacy requirements must be met to guarantee hospital and patient privacy:

1. **Privacy of the hospitals' datasets:** Each hospital records patients' medical histories and diagnoses in their databases, thereby forming a dataset that could be used as a training corpus for the PPRF algorithm. However, due to the sensitive nature of such records, hospitals normally refrain from sharing them. Therefore, their privacy must be ensured and preserved when we design our model.

2. **Privacy of the hospitals' votes:** Hospitals may want to remove a single or a set of trees from the ensemble; however, revealing their votes could expose information from their datasets. Hence, hospitals' votes must remain private.

3. **Privacy of the patients' diagnosed symptoms:** To diagnose a patient, a physician would use the CDSS to retrieve the possible diagnoses based on the patient's symptoms. However, if no privacy guarantees were offered, then no patient would allow his or her symptoms to be fed into the system. Therefore, patients' symptoms must remain private.

3 Notations and preliminaries

3.1 Notations

In this section, we explain the notations that are used throughout the paper. Tab. 1 lists the notations and their meanings.

3.2 Preliminaries

In this section, we present the details of the C4.5 algorithm for generating decision trees and the definition of random forests.

3.2.1 Decision tree algorithm C4.5

Quinlan's C4.5 decision tree algorithm is commonly used in classification problems [Quinlan (2014)]. The main objective of the algorithm is to construct a decision tree from a

dataset of examples and their classes. The algorithm follows the divide-and-conquer model, where at each step it tries to find the best attribute for splitting the dataset. This is done by calculating two values: entropy and information gain. To calculate these values, we set a_j to be one of the possible values of attribute A, where $1 \leq j \leq p$; let n denote the number of classes in dataset D; and select a decision class $c_i \in C$, where C is the set of decision classes and $1 \leq i \leq n$. See Eqs. (1) to (3) for more details. In Eq. (2), the entropy of each attribute in the dataset is calculated. Entropy(a_j) is evaluated using Eq. (1), where D is substituted by a_j.

$$\text{Entropy}(D) = \sum_{i=1}^{n} \left(\frac{-freq(c_i, D)}{|D|} \cdot log_2 \frac{freq(c_i, D)}{|D|} \right) \tag{1}$$

$$\text{Entropy}_A(D) = \sum_{j=1}^{p} \frac{|a_j|}{|D|} \cdot \text{Entropy}(a_j) \tag{2}$$

$$\text{IG}(A) = \text{Entropy}(D) - \text{Entropy}_A(D) \tag{3}$$

The attribute of maximum Information Gain (IG) will be selected as the best attribute for splitting the dataset into p partitions. The process will iterate until there are no more partitions.

3.2.2 Random forest

To construct a random forest, distinct subsets of the dataset are extracted first. Then, a single decision tree is generated from each subset. The generated trees form a random forest ensemble. The classification result is the class returned by the majority of trees [Wu, Feng, Naehrig et al. (2016)].

4 The proposed PPRF algorithm

The PPRF algorithm uses the C4.5 algorithm to generate single decision trees to form the RF ensemble. In this section, we describe the proposed PPRF algorithm by explaining its steps, followed by a description of the complete algorithm.

4.1 Design rationale

Physicians usually collect patient's vital signs and symptoms for diagnosis. The symptom-disease matching process can be time consuming [Ledley and Lusted (1959)]. In our model, we propose a privacy-preserving design for a PPCDSS model to improve physicians' productivity and accuracy. The PPCDSS model is based on our privacy-preserving random forest (PPRF) algorithm. The algorithm repeatedly executes a C4.5 algorithm to generate decision trees from subsets. Then the algorithm aggregates and shares the final ensemble with all hospitals while preserving the privacy of the patients.

An RF is an ensemble of single decision trees, where each is built from a random subset of the dataset. Since our dataset is an aggregation of datasets of all hospitals, we can consider each single dataset as a random horizontal subset. Therefore, each hospital can generate its own random forest and transmit it to the cloud (Fig. 1(a)). There, the ensemble aggregator (Fig. 1(b)) will form the initial global ensemble by removing all redundant trees. The output of the previous step will be sent to the electronic vote manager (EVM) (Fig. 1(c)), which it will send it to all hospitals to review and vote (Fig. 1(d)). Each hospital will review the initial ensemble and vote out any tree that reveals knowledge of its own dataset. Then, the hospital will send its vote back to the cloud (Fig. 1(e)). Each vote can only be seen by the cloud. Thus, the identity of an objecting hospital is hidden from the others. Finally, the EVM will remove the trees that have been voted out from the ensemble and transmit the final global ensemble to all hospitals (Fig. 1(f)). Although no security or privacy techniques were used in this model, privacy preservation was achieved by hiding the original datasets.

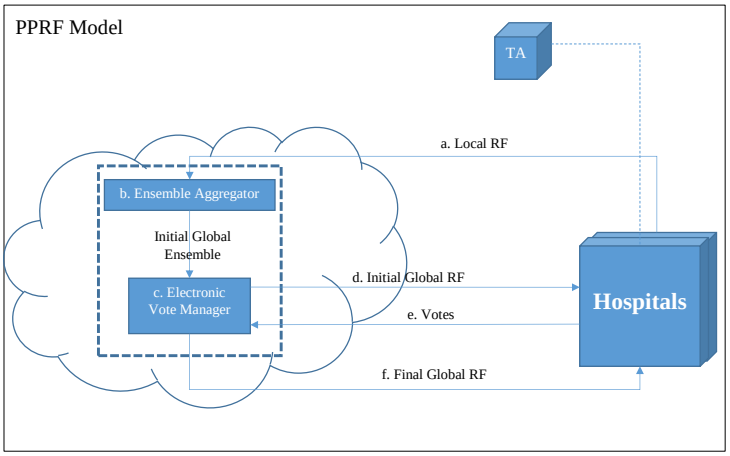


Figure 1: Privacy-preserving random forest (PPRF) model

4.2 Construction

The following steps describe the PPRF algorithm and the roles of the cloud and hospitals:

Step 1. The cloud will prompt each hospital h to prepare its local RF ensemble, which is denoted as RF_h.

Step 2. Each hospital h will prepare its ensemble RF_h by generating decision trees $t_1....t_s$ from various subsets and send it to the cloud.

Step 3. The cloud will receive the ensembles and securely aggregate them into one ensemble after removing redundancies.

Step 4. The EVM will be launched by the cloud to eliminate trees that are unwanted by the hospitals from the ensemble.

Step 5. Each hospital will cast its votes and encrypt them using the cloud's RSA public key.

Step 6. Upon receiving the votes, the cloud will remove voted-out trees to form the final ensemble.

Step 7. The cloud will send the final ensemble to all hospitals.

4.3 Algorithm design

As described earlier, each hospital will generate random subsets of its own dataset. Then, from each subset dataset, a decision tree will be built using the algorithm that is presented in Section 3.2.1. After the cloud securely aggregates the distinct trees, it will run the EVM. The final results will be propagated to all hospitals. This process can be detailed as follows:

1. At each hospital site h, random subsets $(d_1...d_s)$ of the dataset D are generated.

2. For each $d_i \in (d_1....d_s)$, a decision tree t_i is built.

3. All decision trees $t_1....t_s$ are sent to the cloud as random forest ensemble RF_h (Algorithm 1).

Algorithm 1: Build Local Random Forest Ensemble

Input: k = number of generated trees
Output: local RF
LocalRF={}
for $i \leftarrow 0$ ***to*** k **do**
    $d_i = randomSubset(D)$
    $t_i = buildDecisionTree(d_i)$
    if t_i ***is unique*** **then**
        LocalRF.add(t_i)
end
return LocalRF

Algorithm 2: Build Global Random Forest Ensemble

Input: $LocalRF_h$ = local RF ensemble from hospital h
k = number of generated trees
n = number of hospitals
Output: global RF
GlobalRF = { }
for $i \leftarrow 0$ ***to*** n **do**
  for $j \leftarrow 0$ ***to*** k **do**
    $t = LocalRF_i[j]$
    if t ***is unique*** **then**
      GlobalRF.add(t_i)
  end
end
return GlobalRF

Hospital 1:

#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	1	0	0	0	0
2.	0	1	0	0	0	0
3.	0	1	1	0	1	1
4.	0	0	0	0	0	0
5.	0	0	1	1	1	2
6.	1	1	1	1	0	3

Hospital 2:

#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	0	1	1	1	2
2.	0	0	1	0	0	2
3.	0	1	0	0	0	0
4.	0	0	1	0	0	2
5.	0	0	0	0	0	0
6.	0	1	0	0	0	0
7.	1	1	1	1	0	3
8.	0	1	1	0	1	1
9.	1	1	0	1	0	1

Hospital 3:

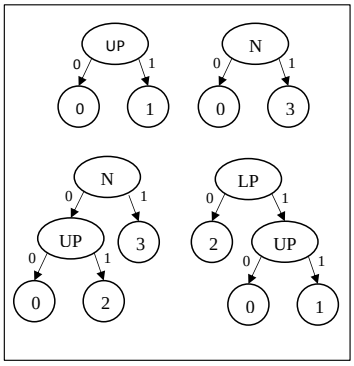
#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	1	0	0	0	0
2.	0	0	1	1	0	2
3.	1	1	1	1	1	3
4.	1	1	1	1	0	3
5.	0	0	0	0	0	0
6.	0	1	1	0	1	1
7.	1	1	0	1	0	1

Figure 2: Hospitals datasets. Acquired from Dheeru et al. [Dheeru and Karra Taniskidou (2003)]

Hospital 1:

#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	1	0	0	0	0
2.	0	1	0	0	0	0
3.	0	1	1	0	1	1
4.	0	0	0	0	0	0
5.	0	0	1	1	1	2
6.	1	1	1	1	0	3

(a)



(b)

Figure 3: RF of Hospital 1: (a) Hospital 1's dataset; (b) Hospital 1's RF model

Algorithm 2 describes how the cloud receives RF_h from each hospital and securely aggregates them, while removing redundancies, into one ensemble (the initial global ensemble). Then, a secure electronic vote will be held to allow hospitals to anonymously vote out any tree without revealing their identities to other hospitals. After agreeing upon the ensemble of trees, each hospital now will acquire the same random forest ensemble (the final global ensemble).
By the end of the training process, each hospital can diagnose new symptoms locally. Since RF returns the disease class that received the majority of the votes, the top 3 disease classes can be returned.

4.4 Descriptive scheme

In this section, we describe an experimental case study of the PPRF model. Consider the datasets of the three hospitals in Fig. 2. We intend to show how the three hospitals will collaborate to build the RF model.

First, each hospital will randomly draw subsets of its dataset and generate decision trees based on these subsets (Figs. 3, 4 and 5) to form its local RF model.

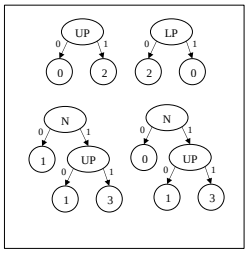
Then, the cloud will receive the three RF models (RF1, RF2, and RF3) and remove redundancies (RF2 and RF3 in Figs. 4 and 5, respectively have a common tree). Therefore, the initial global RF model will be as shown in Fig. 6. Thereafter, the cloud will send the initial model to all hospitals to run the vote. Assuming no hospital objects to any decision tree in the model, the final global model will be as shown in Fig. 6.

Consider the example of Dr. Alice and Bob. Dr. Alice will use the RF model to trace Bob's symptoms (Nausea = 0, Lumbar pain = 0, Urine pushing = 1, Micturition pain = 1, Burning of urethra = 1). The model will return 7 votes for class 2, 3 votes for class 0, 2 votes for class 1, and 0 votes for class 3. Therefore, Dr. Alice can confirm that Bob would most likely be diagnosed with inflammation of the urinary bladder.

Hospital 2:

#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	0	1	1	1	2
2.	0	0	1	0	0	2
3.	0	1	0	0	0	0
4.	0	0	1	0	0	2
5.	0	0	0	0	0	0
6.	0	1	0	0	0	0
7.	1	1	1	1	0	3
8.	0	1	1	0	1	1
9.	1	1	0	1	0	1

(a)



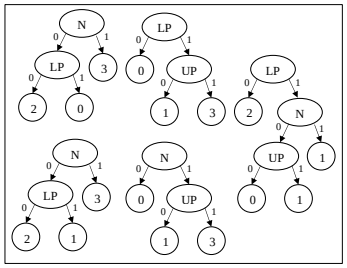
(b)

Figure 4: RF of Hospital 2: (a) Hospital 2's dataset; (b) Hospital 2's RF model

Hospital 3:

#	Nausea	Lumbar pain	Urine pushing	Micturition pains	Burning of urethra	Class
1.	0	1	0	0	0	0
2.	0	0	1	1	0	2
3.	1	1	1	1	1	3
4.	1	1	1	1	0	3
5.	0	0	0	0	0	0
6.	0	1	1	0	1	1
7.	1	1	0	1	0	1

(a)



(b)

Figure 5: RF of Hospital 3: (a) Hospital 3's dataset; (b) Hospital 3's RF model

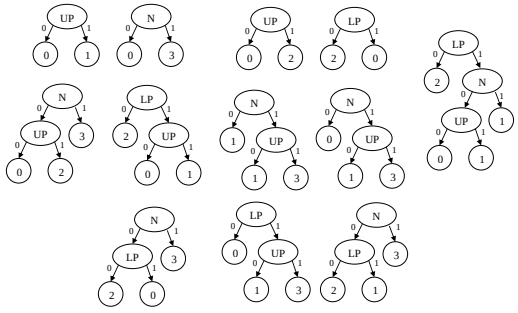


Figure 6: RF Model after securely aggregating the RF model of all hospitals

5 Privacy analysis

In this section we show how our design goals for preserving the privacies of the patients, the hospitals' datasets, the evaluated proportions and the voting process were all satisfied.

5.1 Privacy of hospitals' datasets

The PPRF model aims at building a random forest classification model from distributed datasets without disclosing them. Our model shows how each hospital was responsible for building the trees, and the cloud's main task was to collect the trees and filter them. Therefore, the privacy of the datasets was preserved by not sharing or transferring them over the network.

5.2 Privacy analysis of the EVM

The EVM is responsible for securely collecting participants' votes. The EVM will supply all parties with its public RSA 32-bit key to encrypt their votes before sending them to the cloud. To decrypt the votes, an adversary must obtain the private key; therefore, the resistance of the EVM to attacks is dependent on the secrecy and strength of the private key.

5.3 Privacy of the diagnosed symptoms

The proposed model allows the hospitals to store the classification model at their sites. Thus, physicians may use the system locally and offline for diagnosis. Hence, patients' symptoms will not be exposed to cyber-attacks by transmitting them over the network.

6 Performance evaluation

To simulate the model, we have used a laptop with an Intel Core Processor i7 and 8 GB RAM with a Linux OS (Ubuntu 16.04) and Python as the programming language. We aim at evaluating our PPCDSS in terms of confidentiality and correctness through security analysis. Our goal is to ensure that the implemented security measures do not affect the accuracy of the data-mining techniques that are used.

Dataset. The dataset that we use to generate the decision trees was acquired from [Dheeru and Karra Taniskidou (2003)] with one amendment: We have removed the "Temperature" attribute, because after converting it from a continuous to a discrete attribute (0 = "no fever" and 1 = "fever"), it was found that it always yielded the lowest information gain (IG) value. Hence, we considered five attributes: Nausea (N), Lumbar pain (LP), Urine pushing (UP), Micturition pain (MP), and Burning of the urethra (BU).

Implementation. The dataset was distributed among the hospitals and each hospital generated decision trees from a randomly sampled subset of the dataset. During set-up, the number of generated trees (n) will be decided. After each hospital has generated n distinct decision trees, they will be sent to the cloud for assembly into a single collection or ensemble. Then, the ensemble will be sent back to the hospitals for them to approve it. Each hospital will review the ensemble and fill their votes into an array, of which each element

corresponds to one tree in the ensemble. Before the votes are returned, they will be encrypted using the RSA algorithm to hide them from other hospitals so that no hospital can know the others' votes; the cloud's public key will be used in this case. After receiving the votes, the cloud will exclude every voted-out tree from the ensemble to form the final RF ensemble. The final ensemble will be propagated back to the hospitals; hence, all hospitals will obtain the same RF model.

Performance on datasets. Using datasets (DSs) of various sizes that range from 10 to 120 tuples, we find that the average runtime of PPRF increases as the size of the DS increases (Fig. 7). However, the of increase is not significant; the shortest runtime (0.2 seconds) is observed when the DS size is 10 tuples and the longest runtime (1 second) when the DS size is 120 tuples.

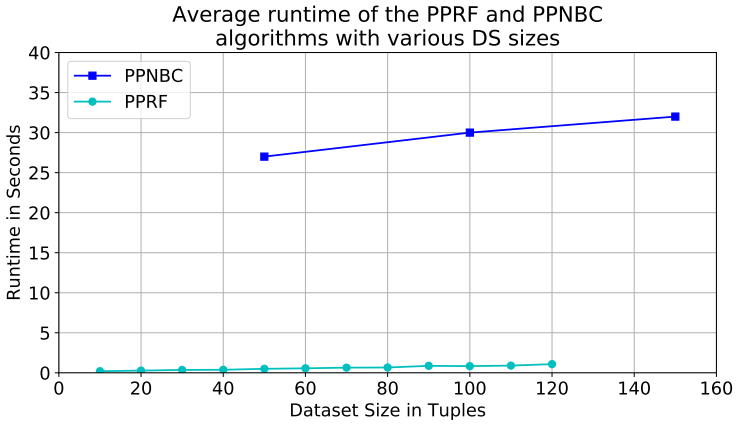


Figure 7: Average runtimes of the PPRF and PPNBC algorithms over various DS sizes. NOT $= 100$

Evaluation on NOT. The simulation code was tested using various numbers of generated trees (NOTs) that ranged from 10 to 100. Fig. 8 depicts the changes in the average runtime when increasing NOT. We find that the the average runtime increases as NOT increases; the longest runtime of 0.1 seconds is observed when NOT = 10 and the slowest runtime of 1.1 seconds when NOT = 100.

Performance of cipher functions. We use RSA in our PPRF algorithm to encrypt the votes, which are stored in an array, element by element. The details of the runtime results show that the encryption and decryption operations take $\approx 0.21\%$ of the average total runtime. The encryption operation took an average of approximately ≈ 0.0002 seconds, whereas the decryption operation took approximately ≈ 0.001 seconds. Fig. 9 and 10 depict the average runtimes for the encryption and decryption operations, respectively. The figures show a slight linear increase in the runtime as the NOT increases; that is because the array size is equal to the size of the global ensemble, and therefore, the growth of the functions is $O(n)$.

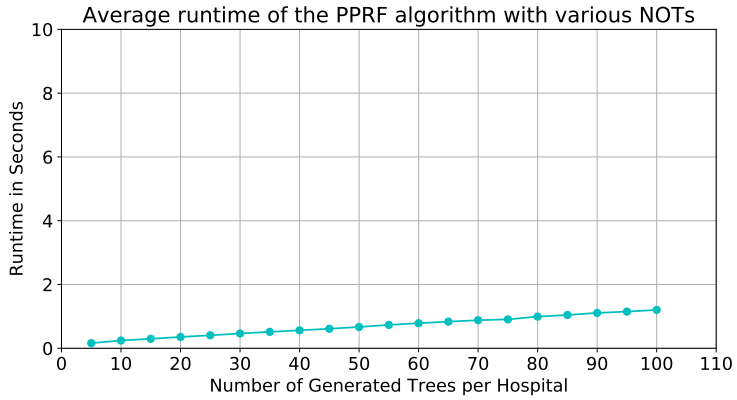


Figure 8: Average runtime of the PPRF algorithm over various NOTs. $|DS| = 120$

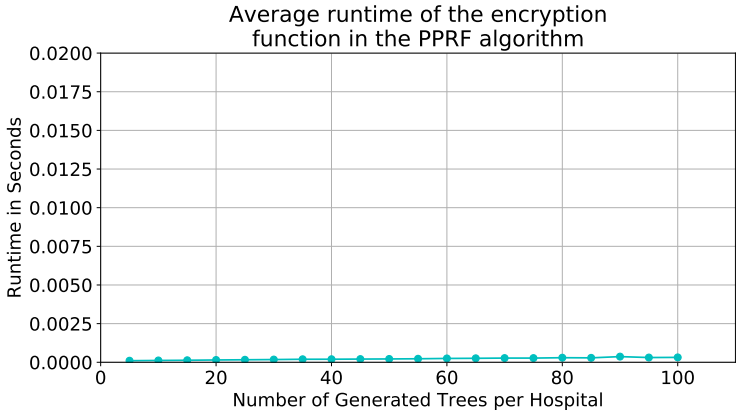


Figure 9: Average runtime of the encryption function. $|DS| = 120$

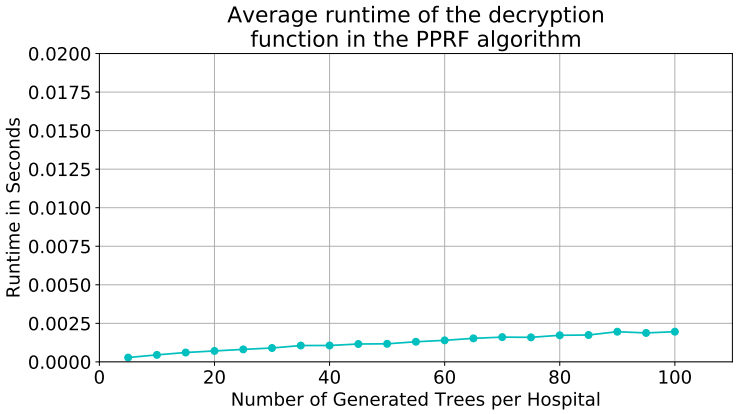


Figure 10: Average runtime of the decryption function. $|DS| = 120$

Evaluation of cipher key size. All previous simulations were conducted using RSA key pairs of 32 bits. Even though the key length should be atleast 2048 bits, we were limited to our choice due to the machine specifications. However, since the encryption and decryption functions are used to encrypt arrays of small integers, the overhead of using larger keys is expected to be negligible. Therefore, it is recommended to use larger key sizes.

To determine how the key size affects the average runtime of the PPRF algorithm, we repeat the simulations with three key sizes: 8 bits, 16 bits, and 24 bits. According to Fig. 11, the key size has no major effect on the average runtime.

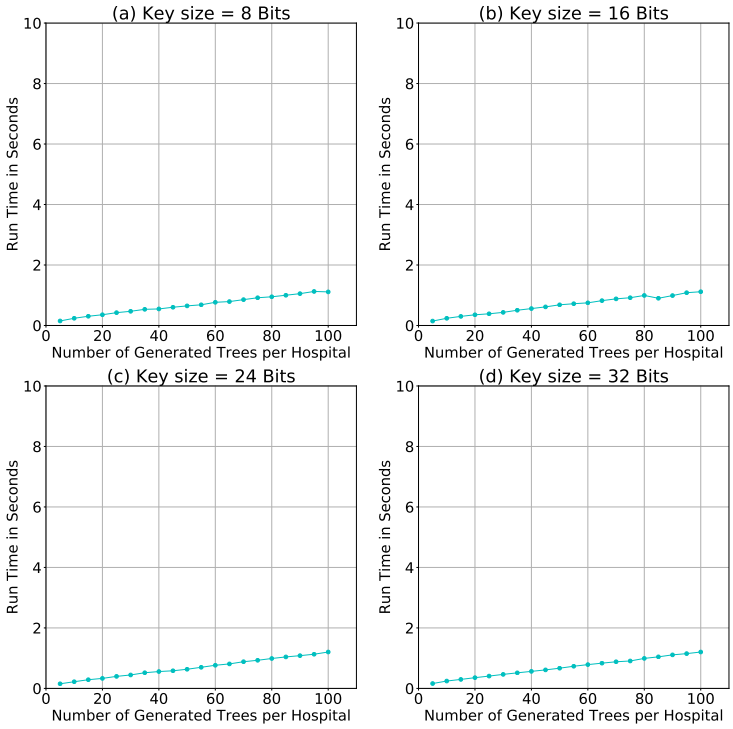


Figure 11: Average runtime of the PPRF algorithm using various key sizes. $|DS| = 120$

Performance comparison. The study in Liu et al. [Liu, Lu, Ma et al. (2016)] proposed a PPNBC classification algorithm that uses Paillier, the secure multiplication protocol [Samanthula, Elmehdwi and Jiang (2015)], and their novel additive homomorphic proxy aggregation (AHPA) protocol for secure aggregation. Furthermore, the authors have used the same dataset in Dheeru et al. [Dheeru and Karra Taniskidou (2003)] with some modifications; they have converted the "Temperature" attribute into 51 attributes, one for each temperature degree. The average runtime of their algorithm when the dataset size is between 50 and 100 tuples ranges between 27 seconds and 30 seconds (Fig. 7). However, our PPRF algorithm performed better, with average runtimes for the same dataset size that range between 0.5 seconds and 0.8 seconds. As a result, our PPRF algorithm has outper-

formed the NBC algorithm in Liu et al. [Liu, Lu, Ma et al. (2016)] in terms of average runtime. Moreover, for diagnosing a new patient using the PPNBC model, patients' symptoms will be encrypted and transmitted to a third party (where the classification model is stored) via the cloud, thereby exposing patients' data to network attacks. Our model bypasses these risks by storing the classification model at the hospitals' sites, which enables the hospitals to diagnose patients offline.

Scalability. As the system grows by increasing the number of participating hospitals, it is expected to have more ensembles to filter. The growth of this function depends on the number of trees per ensemble. However, the array of votes' size is not effected by the system's growth because it is linked to the NOT. Furthermore, the growth of the tree generation process is determined by the NOT, subsets' sizes, and/or number of features. The increase in any of these parameters will increase the growth of the ensemble generation.

7 Related work

Ledley et al. [Ledley and Lusted (1959)] are the pioneers of the CDSS. They demonstrated the efficiency of computers in simplifying complicated diagnostic processes. Later, additional studies, such as Warner [Warner (1961)], proposed new CDSSs. The authors were the first to classify congenital heart diseases using a Bayesian classifier. The study in Schurink et al. [Schurink, Lucas, Hoepelman et al. (2005)] proposed models that addressed issues that intensive care unit physicians commonly face in relation to the diagnosis and treatment of infectious diseases, using naïve Bayes. Addressing privacy issues that are related to CDSS, the authors in Liu et al. [Liu, Lu, Ma et al. (2016)] have proposed a PPNBC algorithm for a PPCDSS to preserve patients' privacy while using the system. They have solved the collusion problem between the cloud and the processing unit through their novel AHPA algorithm.

Random ensembles of decision trees were also used in medical applications. For instance, the study in Alickovic et al. [Alickovic and Subasi (2016)] used RF, CART, and C4.5 to diagnose heart arrhythmia; their RF model performed the best in terms of accuracy. Furthermore, the study in Azar et al. [Azar and El-Metwally (2013)] compared the SDT, boosted decision tree, and decision tree forest and found that the decision tree forest performed the best in terms of accuracy. The study in Özçift et al. [Özçift (2011)] proposed an RF model that yielded 76-90% classification accuracy and compared their model to other studies in literature. As a result, we conclude that random ensembles of decision trees yield a better accuracy than that of single decision trees.

8 Conclusion

Taking the advantage of technological advancements in medical field increases the efficiency and quality of the healthcare services that are provided by health institutions. CDSS is an example of a medical application that improves physicians' performances and patients' experiences. However, requiring these systems to work over the network poses a threat to patients' privacy by exposing their data to network attacks.

In this work, we proposed a privacy-preserving healthcare system, namely, PPCDSS, based

on the proposed PPRF algorithm. With the removal of unnecessary attributes and the avoidance of cryptography methods, simulation results have demonstrated that the PPRF algorithm outperformed the PPNBC algorithm in Liu et al. [Liu, Lu, Ma et al. (2016)] in terms of average runtime. Thus, our proposed PPRF algorithm provides a privacy-preserving environment for transmitting patients' records over the network and building a decision-tree ensemble without disclosing patients' information. As a future work, we intend to generalize our model to accept attributes with more possible values, and also to insure integrity of the model with Message Authentication Code (MAC).

Acknowledgement: This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University.

References

Abdar, M.; Kalhori, S. R. N.; Sutikno, T.; Subroto, I. M. I.; Arji, G. (2015): Comparing performance of data mining algorithms in prediction heart diseases. *International Journal of Electrical and Computer Engineering*, vol. 5, no. 6, pp. 1569-1576.

Alabdulkarim, A.; Al-Rodhaan, M.; Tian, Y. (2017): Privacy-preserving healthcare system for clinical decision-support and emergency call systems. *Communications and Network*, vol. 9, no. 4, pp. 249.

Alickovic, E.; Subasi, A. (2016): Medical decision support system for diagnosis of heart arrhythmia using dwt and random forests classifier. *Journal of medical systems*, vol. 40, no. 4, pp. 108.

Azar, A. T.; El-Metwally, S. M. (2013): Decision tree classifiers for automated medical diagnosis. *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2387-2403.

Berner, E. S.; La Lande, T. J. (2007): Overview of clinical decision support systems. *Clinical Decision Support Systems*, pp. 3-22.

Chaurasia, V.; Pal, S. (2013): Data mining approach to detect heart dieses. *International Journal of Advanced Computer Science and Information Technology*, vol. 2, no. 4, pp. 56-66.

Dheeru, D.; Karra Taniskidou, E. (2003): Uci machine learning repository: acute inflammations data set. https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations.

Krishnaiah, V.; Narsimha, D. G.; Chandra, D. N. S. (2013): Diagnosis of lung cancer prediction system using data mining classification techniques. *International Journal of Computer Science and Information Technologies*, vol. 4, no. 1, pp. 39-45.

Ledley, R. S.; Lusted, L. B. (1959): Reasoning foundations of medical diagnosis; symbolic logic, probability, and value theory aid our understanding of how physicians reason. *Science*, vol. 130, no. 3366, pp. 9-21.

Liang, X.; Lu, R.; Chen, L.; Lin, X.; Shen, X. (2011): Pec: a privacy-preserving emergency call scheme for mobile healthcare social networks. *Communications and Networks, Journal of*, vol. 13, no. 2, pp. 102-112.

Liu, X.; Lu, R.; Ma, J.; Chen, L.; Qin, B. (2016): Privacy-preserving patient-centric clinical decision support system on naive bayesian classification. *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 655-668.

Lu, R.; Lin, X.; Shen, X. (2013): Spoc: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 614-624.

Ma, T.; Zhang, Y.; Cao, J.; Shen, J.; Tang, M.; et al. (2015): Kdvem: a k-degree anonymity with vertex and edge modification algorithm. *Computing*, vol. 97, no. 12, pp. 1165-1184.

Musen, M. A.; Middleton, B.; Greenes, R. A. (2014): Clinical decision-support systems. *Biomedical Informatics*, pp. 643-674.

Özçift, A. (2011): Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis. *Computers in Biology and Medicine*, vol. 41, no. 5, pp. 265-271.

Quinlan, J. R. (2014): *C4. 5: Programs for Machine Learning*. Elsevier.

Rahman, R. M.; Afroz, F. (2013): Comparison of various classification techniques using different data mining tools for diabetes diagnosis. *Journal of Software Engineering and Applications*, vol. 6, no. 3, pp. 85.

Rong, H.; Ma, T.; Tang, M.; Cao, J. (2018): A novel subgraph k^{+} -isomorphism method in social network based on graph similarity detection. *Soft Computing*, vol. 22, no. 8, pp. 2583-2601.

Samanthula, B. K.; Elmehdwi, Y.; Jiang, W. (2015): K-nearest neighbor classification over semantically secure encrypted relational data. *IEEE transactions on Knowledge and data engineering*, vol. 27, no. 5, pp. 1261-1273.

Schurink, C.; Lucas, P.; Hoepelman, I.; Bonten, M. (2005): Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units. *The Lancet Infectious Diseases*, vol. 5, no. 5, pp. 305-312.

Warner, H. R. (1961): A mathematical approach to medical diagnosis: application to congenital heart disease. *JAMA*, vol. 177, no. 3, pp. 177.

Wu, D. J.; Feng, T.; Naehrig, M.; Lauter, K. (2016): Privately evaluating decision trees and random forests. *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 335-355.

Xiong, L.; Shi, Y. (2018): On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing. *Computers, Materials & Continua*, vol. 56, no. 1, pp. 137-149.

Zhang, P.; Tong, Y.; Tang, S.; Yang, D. (2005): Privacy preserving naïve bayes classification. *Advanced Data Mining and Applications*, pp. 744-752.