International Journal of E-Health and Medical Communications
Volume 12 • Issue 6 • November-December 2021

An Ensemble Random Forest Algorithm for Privacy Preserving Distributed Medical Data Mining

Musavir Hassan, University of Kashmir, India
Muheet Ahmed Butt, University of Kashmir, India
Majid Zaman, University of Kashmir, India

ABSTRACT

A voluminous amount of data is generated because of the inexorably widespread proliferation of electronic data maintained using the electronic health records (EHRs). Medical health facilities have great potential to discern patterns from this data and utilize them in diagnosing specific diseases or predicting the outbreak of an epidemic. This discerning of patterns might reveal sensitive information about individuals, and this information is vulnerable to misuse. This is, however, a challenging task to share such sensitive data as it compromises the privacy of patients. In this paper, a random forest-based distributed data mining approach is proposed. Performance of the proposed model is evaluated using accuracy, f-measure, and κappa statistics analyses. Experimental results reveal that the proposed model is efficient and scalable enough in both performance and accuracy within the imbalanced data and also in maintaining the privacy by sharing only useful healthcare knowledge in the form of local models without revealing and sharing sensitive data.

KEYWORDS

Decentralized Data Mining, F-Measure, Healthcare, Horizontally Partitioned Data, Privacy Preserving, Random Forest

1. INTRODUCTION

The age of big data has empowered several relations to gather extensive volumes of information. In many real world applications data required for crucial data mining tasks is distributed among several parties. To find useful patterns from the data and discover knowledge that can't be mined from the data of single party, these parties must share data. It is unfeasible to centralize the data from participating parties due to huge communication costs, computation costs, central storage requirements, security and most importantly privacy concerns. To overcome the drawbacks of centralized system, efficient global models can be constructed from collaborative participants. But this collaborative participation is challenging due to the privacy concerns of participants, as sharing of data among the participants is required. Thus, various distributed data mining algorithms have been proposed in literature to mine different patterns extracted from data shared among different participants without revealing the original data.

Data shared among different participants may have the same attributes at each participant location; such data is said to be horizontally partitioned. For example, medical data of patients who

DOI: 10.4018/IJEHMC.20211101.oa8

suffer from a common disease will have the same attributes maintained with each medical facility. On the other hand, data belonging to a specific entity may be shared among different participants such that different participants store different attributes of the same entity. Such data is said to be vertically partitioned data. For example, medical data of a patient may be stored by a medical facility whereas data regarding medical bill data, health cover information, etc. of the same patient may be stored by an insurance company. Various distributed privacy preserving approaches based on different machine learning algorithms to mine horizontally and vertically partitioned data have been proposed in the literature. One such approach is to perform local data mining at different participant locations in parallel to produce local data models and keep the disjoint datasets to their respective locations. These local models are then transmitted to a central site that combines them into a global model (Myneni and Patel (1999), Chawla*et al*. (2004), Tsoumakas (2003)). The second approach is that, from each local site original data is sub-sampled and then accumulated at a central site to form a global subset (Chawla*et al*. (2004)). Another approach is to introduce perturbation in local data of participants with the help of a third-party coordinator in order to preserve the privacy of data. The perturbed data from each participant can then be published in the form of a centralized database to perform different data mining tasks as done by Sheela and Vijayalakshmi (2017). Distributed data mining algorithms that work in a fully decentralized manner have also been proposed in literature. The participants involved, mine shared data by using message passing mechanism. Such algorithms are characterized by the distribution of data on each participant site and asynchronous communication so as to enable learning from participants that aren't available at a given time. Such algorithms should also be scalable so as to work with more participants and therefore more data which may be added to the system at a later time. An important consideration while using decentralized distributed data mining algorithms is to preserve the privacy of data local to each participant. There are potential weaknesses in above mentioned techniques that may put the privacy of the data at risk. Moreover, different privacy preserving methods used in these techniques have certain limitations discussed in Hassan *et al.* (2017).

Privacy in distributed data mining systems becomes a critical issue when sensitive data like health data of patients is involved. Maintaining the confidentiality and privacy of information regarding a patient's healthcare data is a very difficult task(Bisui and Misra(2019)).Privacy concerns hamper the transferring as well as sharing of sensitive data. Many healthcare facilities have adopted electronic health records (EHR) to store and maintain patient data in order to enhance the quality of healthcare service delivery. There is a huge potential to enhance healthcare services further and make predictions about diseases, diagnostics and medications more accurate if systems are designed that are capable of integrating different healthcare facilities so that the data maintained by these facilities is accessible for data mining. However, one of the main obstacles in using distributed health data for disease detection is patients' privacy as the EHR may contain patient information about demographics, diagnostics, medications and other health related information. If such data is not properly utilized, it could put the privacy of patients at risk. To protect individual privacy, government agencies, e.g.; HIIPA of United States and ECHR of European Union, have endorsed many laws to protect an individuals' privacy. The basic requirements for security and privacy in Indian healthcare system are provided in the standard "ISO/TS 14441:2013 Health Informatics Security and Privacy Requirements of EHR Systems for Use in Conformity Assessment". A law has been proposed by India's Ministry of Health to govern data security in healthcare sector that would give individuals complete ownership of their health data. On March 11, 2018, the draft of digital information security in healthcare Act was proposed by the Ministry of India. To preserve the privacy of patients, EHR must abide these rules and must be certified by certain institutes like Certification Commission for Healthcare Information Technology (CCHIT), Office of the National Coordination for Health Information Technology (ONC) and Ministry of Health and Family Welfare, Government of India.

The problem definition of the work that is being developed in this paper is as follows: The healthcare facilities are rapidly adapting the electronic health records because of the potential

benefits they seem to offer but there are many obstacles that hamper their effective use. Firstly, the electronic health records contain the private and sensitive data of the patient, and as such sharing of such data will result in revealing of patient privacy. Secondly the lack of sufficient patient records in a newly established healthcare facility will affect the development of a robust decision making system. In order to receive maximum benefit of electronic health records, a distributed data mining approach is required but there are many challenges that need to be dealt with. In the distributed data mining environment all the healthcare facilities develop a local model. Most of the distributed data mining models need a third party to revise and aggregate the local models to construct the integrated final model. This leads to an extra cost in model implementation and each participator must have frequent contact with the central third party, As such a significant portion of the time is spent on the communication between the participators rather than the actual computation itself.

To deal with these challenges, we propose a privacy preserving random forest classification on horizontally portioned data such that each party need not disclose its data to other parties while acquiring the same accuracy as when the data is centralized. In this work, an ensemble model is formed by a set of more basic models (decision trees) and the prediction of new instance is computed by merging together the basic predictions at the local participator site. The local participators share their local models to obtain the final integrated model for better prediction performance without revealing any sensitive information.

The contributions of this research can be summarized as follows:

- In the beginning, we examine how the medical research can be enhanced by the distributed data mining on horizontally partitioned healthcare data and simultaneously how privacy of patients can be preserved.
- Propose a random forest-based approach for decentralized health data mining to diagnose different diseases under privacy constraints.
- We scrutinize the proposed scheme with cardiotocography dataset (CTG) and Thyroid disease dataset (TDD) available at UCI machine learning repository. The analysis of results shows that accuracy of the integrated model is better than the accuracy of the individual local models.

The rest of the paper is organized as follows: In section 2, we provide important relevant works on privacy preserving data mining. In section 3 we propose our decentralized random forest based approach to diagnose different diseases. Section 4 illustrates dataset details and results. Finally section 5 concludes our discussion with some future research directions.

2. LITERATURE REVIEW

Various data mining techniques have been proposed in literature to identify and extract the useful knowledge and patterns from massive amounts of data. For example, different healthcare facilities would like to analyze the health records via data mining techniques to identify patterns of some diseases. But the data from which the useful knowledge is mined may also contain sensitive information and the mining of this data may become threat to privacy of the patients.Various prediction models have been proposed to predict different diseases that can be deployed in different healthcare facilities (Sali*et al*. (2016), Meng*et*

al. (2018), Ahn*et al*. (2018)). Although, these models lack scalability because of privacy constraints and must be implemented at a local level, with suitable modifications they may be extended to take advantages of distributed data mining. In order to achieve distributed data mining of such sensitive data, different privacy preserving data mining techniques that deals with protecting the privacy of sensitive data without sacrificing the utility of data have been proposed. Different privacy preserving data mining techniques have been proposed in literature that include randomization, *k*-anonymization and distributed privacy preserving data mining (Agarwal and Yu (2008)) to ensure the privacy of

sensitive data. The randomization method is a privacy preserving data mining technique in which the original records are perturbed for concealing certain sensitive information. The techniques used for perturbation in randomization technique include additive perturbation (Agarwal and Agarwal (2001)), multiplicative perturbation (Samarati (2001)), data swapping (Fienberg and McIntyre (2004)), principal component based analysis (Huang *et al.* (2005)) etc. The problem of linking may arise due to some publicly available data which may reveal the hidden sensitive information in processed records and hence the privacy will be compromised. For example, in healthcare applications, sensitive information about any individual can be revealed by any publicly available voter data, so randomization is inadequate to protect the health data. It is estimated that 87% of the population in the United States can be uniquely identified using the seemingly innocuous attributes of gender, date of birth, and 5-digit zip code. This possibility of indirect de-identification of records from public databases has lead to the development of *k*-anonymity model (Samarati (2001)). With the help of two commonly used anonymization operations i.e. generalization and suppression (Samarati and Sweeney (1998)), individually identifiable information is reduced sufficiently that any given record can be indistinguishably backtracked to at least *k* other records. The *k*-anonymity procedure protects identity disclosure, but it does not protect attribute disclosure sufficiently (Machanavajjhala*et al*. (2006)). Homogeneity attack and background knowledge attack will allow an attacker to identify the individual records. To protect against these attacks, Machanavajjhala*et al*. (2006) introduced *l*-diversity as a stronger notion of privacy. The *l*-diversity principle ensures *l* well represented values for sensitive attributes in every equivalence class. If every equivalence class is *l* diverse, then a table is said to be *l*-diverse. Although, the *l*-diversity principle protects against attribute disclosure, but it may be difficult to achieve. The attacks like skewness attack and similarity attack makes *l*-diversity principle insufficient to prevent attribute disclosure. To overcome the limitations of *l*-diversity, Li *et al*. (2007) proposed a novel privacy notion called *t*-closeness, which requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of attribute in the overall table. The third category is distributed privacy preservation that allows computation of useful cumulative statistics over the whole data without compromising the privacy of individual dataset within multiple parties. Thus, for obtaining the cumulative statistics, the data sets can be horizontally or vertically partitioned. In this work, data set is horizontally partitioned and this patient data is available at multiple healthcare facilities with each record having same set of attributes.

Lindell and Pinkas (2000) presented an important research in which the decision tree using *ID3* algorithm is extended to two parties without having to reveal any sensitive data in the process. Subsequently, a variety of approaches have been proposed to solve the problem of horizontally partitioned privacy preserving data mining, for example, Naïve Bayes Classifier (Kantarcıoglu*et al*. (2003)), SVM classifier(Yu *et al*. (2006)) and ensemble based classifiers, association rule mining on horizontally partitioned data (Evfimievski*et al*. (2004)) and ensemble based classifiers. Ensemble methods are considered the most influential development in machine learning and data mining among all the above mentioned methods. Ensemble methods merge multiple models into one model that is more accurate than the rest of its components. Ensemble methods also provide a boost to real world application challenges from investment timing to drug discovery, and fraud detection to recommendation systems – where predictive accuracy is more vital than model interpretability. Ensemble methods have also been observed to show good performance in preserving privacy. Gambs*et al*. (2007) proposed BiBoost and MultiBoost privacy preserving algorithm that allows two or more participants to construct a boosting classifier without explicitly sharing their data set. Each entity obtains a weak classifier from its own data and thus ensures local data privacy. Different models including decision trees, logistic regression and neural network models have also been proposed for health data mining (Hassan et al. (2017)). Evfimievski et al. (2004) use neural network ensembles to obtain strong classifier from weak classifiers. Privacy preserving distributed algorithm for training neural network ensembles is designed using AdaBoost.Kou *et al.* (2004) use data separation techniques to preserve privacy in classification of medical data. To protect the privacy of data, both horizontal

and vertical, portioning approaches are used to mine the data at multiple sites. Distributed results are assembled at central trusted party using a majority vote ensemble method.Bialy*et al.* (2016) developed an intelligent decision support system for heart disease diagnosis using an ensemble model (Bagging, Boosting and stacking) based on six classifiers; Naive Bayes, Bayesian Net, Multilayer perceptron (MLP), Sequential Minimal Optimization for Support Vector Machine, C4.5 Decision Tree and Fast Decision Tree, in order to select the best combination for the heart disease prediction. Outliers and extreme values were removed using the Inter-quartile range to enhance the performance of the model. Bashir*et al.* (2014) proposed an ensemble classifier for the classification and prediction of heart disease data. Five classifiers; Naive Bayes, decision tree based on Gini Index, decision tree based on information gain, memory-based learner and support vector machine are used for construction of an ensemble classifier. Five data sets from different repositories have been used for testing.Cheon*et al.* (2018) proposed a new method called ensemble GD for logistic regression, which reduces the number of iterations of GD and results in substantial improvement on the performance of logistic regression based on HE in terms of speed and memory.Sheela and Vijayalakshmi (2017) proposed a way to perturb the individual data over vertically partitioned data set with a third party coordinator. Each participant perturb their data by finding the mean until the threshold value is reached and then the perturbed data is published by each participant to perform requisite data mining.

Techniques discussed above achieve the distributed data mining requirements; however there are many problems that need to be addressed. All the participants rely on a central agent, usually owned by a third party. Participants share their local models or representational data sets to revise and integrate them into final collaborative models. Such a model implementation usually incurs additional costs. Participants need to communicate frequently with the central agent and this communication is a pure overhead and no productive computation is done during this time. A more severe problem with these techniques involve their inability to adapt; whenever a new participant is added or when the model requirements change, whole model needs to be learned again. Many techniques rely on privacy preserving techniques to anonymize data before sharing, but anonymization is not enough to protect the data completely. Anonymized data may come under different attacks and put privacy of subjects at risk.Random forests are ensemble learners and make use of decision trees as weak learners. Random forests are particularly advantageous in terms of interpretability compared to neural networks. Because of their comprehensible learning mechanism, random forests are preferable for applications like investigating patient data and suggesting medical treatment. This paper presents a random forest based distributed data mining technique for mining horizontally partitioned data under privacy constraints that addresses the above mentioned challenges by making all participants to train local models from the respective local data in a standalone manner.

3. PROPOSED METHOD

In the proposed system, *n* parties collaborate to facilitate data mining in a distributed manner by sharing random forest based-local models trained on their local data in a decentralized manner. A party P_i learns its own model and receives models from $n - 1$ parties to integrate them into the final integrated model. The symbols and their meaning used in the proposed method are given in Table 1. Framework of the proposed system is described below.

3.1 Local Parties and Local Datasets

Let $P = \{P_1, P_2, \ldots, P_n\}$ denote the set of *n* parties; each party P_i is in possession of alocal data set D^i gathered from real applications over time. Local data set of each partycontains certain amount of sensitive data and direct sharing of this data isn't feasible forprivacy reasons.Each local data set D^i of party P_i can be represented as shown in eq. (1).

Table 1. List of Notations used in this paper.

Symbol	Meaning
n	Number of participating parties
P_i	*i*th party, where $1 \leq i \leq n$
P	P is the set of all participating parties
D^i	Dataset owned by party P_i
U_j^i	*j*th data instance in dataset D^i
m	Number of attributes in each instance
v_j^i	Class label of U_j^i
C	Set of class labels
t	Number of trees in the Random Forest
RF_i	Random forest based local model belonging to party P_i
α_i	Classification accuracy achieved using local model RF_i
n_i	Training sample size used to train RF_i
η_i	Normalized training sample size
DT_l^i	*l*th weak learner (decision tree) belonging to RF_i
V_C	Vector used to store number of votes obtained by each class label using local model
VG_C	Vector used to store number of votes obtained by each class label using integrated model
ζ_j	Integrated model belonging to pj
X	Test instance
κ	Cohen's kappa coefficient

$$
\begin{aligned}
&D^i = \{(U_1^i, v_1^i), (U_2^i, v_2^i), ..., (U_k^i, v_k^i)\} \\
&where, \\
&k = \mid D^i \mid \\
&U_j^i = < u_{j1}^i, u_{j2}^i, ..., u_{jm}^i >; 1 \leq i \leq n, 1 \leq j \leq k
\end{aligned}
\tag{1}
$$

In case of binary classification, $v_j^i \in C = \{-1, +1\}$ is the label for each instance and used to train the ensemble learner. In case of multiclass classification, $v_j^i \in C = \{1, 2, 3, ...\}$ is the label for each instance.

3.2 Local Model

Each party creates a local bootstrapped data set with replacement. The data set D^i with *m* attributes owned by party P_i, a subset of attributes, typically equal to sqrt(*m*), is chosen randomly at each step of the decision tree construction. Attribute that is better at separating the sample is selected at this step and excluded from being chosen at next steps. Whole decision tree is constructed by repeating the same procedure at each node. Decision trees act as weak learners in a random forest model and as such *t* (*typically in hundreds*) number of decision trees is built to construct a random forest ensemble model. A wide variety of decision trees are obtained due to random selection of attributes. A test instance is fed to each weak learner and results obtained are bagged into classes. Class with maximum number of votes is the label of input test instance predicted by the local ensemble model. The model can be represented as follows.

Let, $RF_i = \{DT_1^i, DT_2^i, ...DT_t^i\}$ is random forest based local model of party P_i with *t* number of decision trees. Vector V_c with size equal to |C| is initialized with 0 for each class label. Test input instance U_j^i from the data set D^i of party P_i is input to each of the *t* number of decision trees in RF_i. Predicted class for U_j^i by each decision tree is used to increment the respective label in the vector V_C. Predictions made by a decision tree DT_1^i to classify test instance U_j^i is denoted by *Predict*(DT_1^i, U_j^i). The local ensemble model RF_i use the following methods to evaluate the predicted output for an input test instance U_j^i .If $v_i^j \in C = \{-1, +1\}, i.e. \mid C \mid = 2,$ then, the output of RF_i denoted by $\tau(RF_i, U_j^i)$ is given by eq. (2).

$$\tau(RF_i, U_j^i) = \sum \text{Predict}(DT_l^i, U_j^i) \quad where, 1 \le l \le t \tag{2}$$

It is obvious from eq. (2) that $\tau(RF_i, U_j^i) < 0$, if label -1 scores more votes than label +1; otherwise, $\tau(RF_i, U_j^i) > 0$. Voting tie is usually handled at implementation level. If $v_j^i \in C;$ where, C is the set of class labels and |C| ≥3; then output of the local ensemble model can be computed using eq. (3). Let V_c with size equal to |C| store votes obtained for each label.

$$\tau(RF_i, U_j^i) = MaxLabel(V_c) \tag{3}$$

where, MaxLabel(V_c) returns the label with maximum votes.

The *f-measure*, discussed under section *Performance Metrics*, is a preferable metric to evaluate the performance of local ensemble models using balanced as well as imbalanced data. Training sample size taken from the dataset D^i by party P_i to train the local model RF_i is denoted as n_i. Local model RF_i along with $f\text{–}measure_i$ and n_i are shared with other parties to build the integrated model (ζ_i).

3.3 INTEGRATED MODEL

Each party build its local model on horizontally partitioned data and in order to build the integrated model ζ_j, party P_j request and use local models from remaining *n–1* parties. Each party P_i share its local model with corresponding $f\text{–}measure_i$ and sample size as a *3*-tuple (RF_i, $f\text{–}measure_i$, n_i) where, RF_i is the local model, and n_i is the number of training instances chosen from D^i.Party P_j use local models of *n–1* parties along with its local ensemble model RF_j to build the integrated model ζ_j. Therefore,

the integrated model ζ_j can be represented as $\zeta_j = \{RF_1, ..., RF_j, ..., RF_n\}$ in its simplest form.Let *X*be a test instance with *m* attributes and C = {−1, +1}, ζ_j can be used to predict the class of *X* using eq.(4).

$$G(\zeta_j, X) = \sum \tau(RF_i, X) \tag{4}$$

where, *j* represents the *j*th party on which the integrated model is constructed, and $1 \le i \le n$.$G(\zeta_j, X)$ is the predicted class label of *X* obtained using ζ_j and $\tau(RF_i, X)$ is the predicted class labels of *X* using local ensemble models RF_i.

In case there are more than two classes in which the training data can be classified, the integrated model $\zeta_j$use eq.(5) to predict the class.Let VG_C with size equal to |*C*| store votes obtained for each label using integrated model ζ_j.

$$G(\zeta_j, X) = MaxLabel(VG_C) \tag{5}$$

where, $MaxLabel(VG_C)$ returns the label with maximum votes.Inclusion of local models exhaustively to obtain the integrated model may have an adverse effect on the performance obtained using the integrated model. In this paper, we discuss two potential ways to fine tune and optimize the performance integrated model as presented in the following subsection.

3.3.1 Integrated Model Optimization

The integrated model is built using a subset of local models using an optimization criterion to prevent negative impact of local model integration that arise as a result of poor performance shown by some of the local models. In the first case, f-measure is used as the optimization criterion to choose and select local models for integration into the integrated model. Party Pj calculate the f-measurej of its own model RFj and use it as threshold to choose models RFi to obtain the final integrated model ζj as follows. Choose RFi and add to ζj, for all i if f-measurei≥f-measurej. Integrated model thus obtained can be represented as ζj = {RFj, RFp, RFq, ...}. The integrated model ζj so obtained can be used to make predictions over the test data. Let X be a test instance with m attributes, ζj is used to classify X using eq. (6).

$$G(\zeta_j, X) = \sum \tau(RF_i \in \zeta_j, X)$$

where, j presents jth party where integrated model is constructed (6)

$i = j, p, q, ...$ and $f - measure_i \ge f - measure_j$

X is the test instance

The second approach to optimization is used if there is significant difference between the sample sizeni on which the local models are trained for each of the parties involved. In this approach, training sample size ni of party Pi is normalized. Normalized sample size of Pi is represented as ηi and calculated using eq. (7).

$$\eta_i = n_i / MaxSample(n_i)$$
$$where, 1 \le i \le n \quad (7)$$
$$MaxSample(n_i) \text{ returns the maximum sample size}$$

In case $v_i^j \in \{-1, +1\}$, then output of the integrated model is calculated using eq.(8).

$$G(\zeta_j, X) = \sum(\tau(RF_i, X) * \eta_i)$$

$$where, j \text{ is the } jth \text{ party where the integrated model is constructed} \quad (8)$$
X is the test instance
η_i is normalized sample size

Here $G(\zeta_j, X)$ can be either positive or negative. In case $G(\zeta_j, X)$ is positive, x is predicted to be $+1$ and otherwise, x is -1.

4. RESULTS AND DISCUSSION

In this section, results of the proposed system are presented and discussed. In the following subsection, we describe the datasets used followed by the performance metrics used to evaluate the performance of the proposed system. Finally, results of the study are presented.

4.1 Dataset Description

In our experiments, we use the Cardiotocography Dataset (CTG) available at UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets/cardiotocography/) that consists of measurements of fatal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. There are total of 2126 instances in the dataset with 22 attributes. Description of these attributes is available on the above given url address. Classification labels used in the dataset are Normal (represented by 1), Suspect (represented by 2), and Pathologic (represented by 3).

The dataset is horizontally partitioned into three disjoint datasets D1, D2 and D3. D^1contains first 900 instances of the *Cardiotocography Dataset*; next 900 instances are storedin D^2 and remaining last 326 of the *Cardiotocography Dataset* instances are stored in D^3.Out of 900 instances in D^1, 574 belong to class 1, 245 belong to class 2 and 81 belong toclass 3.Similarly in D^2, there are 809 instances belonging to class 1, 45 instances are inclass 2 and remaining 46 belong to class 3. There are 272 instances in class 1, just 5 in class2 and 49 instances are in class 3.

The second dataset used in this study is the *Thyroid Disease Dataset* (TDD) available at UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets/thyroiddisease/). TDD is made available by the Garavan Institute, Australia. There are a total of 7200 instances in the dataset with 22 attributes. Classification labels used in the dataset are Normal (represented by 1), Hyperthyroidism (represented by 2), and hypothyroidism (represented by 3). The TDD is horizontally partitioned into 7 disjoint datasetsD1, D2, D3, D4, D5, D6 and D7. Additionally, one disjoint *test* dataset with 500 test instances is also obtained from the TDD. The total number of instances and the number of instances belonging to a particular class are given in Table 2.

Table 2. Horizontal partitioning of TDD dataset

Dataset	Total number of instances	Number of class 1 instances	Number of class 2 instances	Number of class 3 instances
D^1	2000	47	106	1847
D^2	1500	34	86	1380
D^3	900	25	45	830
D^4	800	15	41	744
D^5	700	18	35	647
D^6	500	10	21	469
D^7	300	7	7	286
test	500	10	27	463

4.2 Performance Metrics

Accuracy, *f-measure*, and *Cohen's Kappa statistics* are used as performance measures to evaluate the performance of the proposed local as well as integrated data mining models.

Accuracy (α) is calculated as the total number of correct predictions divided by the total number of instances in the dataset. Accuracy of a local model RF_i is calculated using the formula given in eq.(9).

$$\alpha_i = \frac{\text{Number of instances correctly classified by } RF_i}{\text{Total number of test instances taken from } D^i} \tag{9}$$

Accuracy (α) is a popular metric to evaluate the performance of a classifier and works well on balanced data. But, in case the data is imbalanced in the test dataset, accuracy may give misleading information regarding accuracy of models.

f-measure also called as *F1-score* or *F-score* is considered a more accurate way to measure the performance of the classifier. *f-measure* is calculated as shown in eq. (10).

$$f-measure = \frac{2(precision \times recall)}{precision + recall}$$

$$where,$$

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$recall = \frac{TP}{TP + FN}$$

$$TP - \text{ True Positive}, FP - \text{ False Positive}, FN - \text{ False Negative}$$

Therefore, *f-measure* is basically the harmonic mean of precision and recall and thus effective for handling the imbalanced data distribution in different classes. *f-measure* for each class is computed and $f\text{-}measure_i$ of local model RF_i is computed by averaging the f-measures obtained for each class using eq. (10).

Cohen's Kappa Coefficient: In imbalanced class classification task, accuracy is not good enough as a main evaluation. So to solve the problem of multiclass and imbalanced data we have used the Cohen's kappa metric. (κ) is an effective metric to evaluate the performance of a classifier and evaluate classifiers themselves as well. Measures like accuracy and precision/ recall do not provide complete picture of performance of multi-class classifiers. κ also performs comparatively better to evaluate performance of classifiers in case of imbalanced classes. κ is evaluated using eq.(11).

$$\kappa = \frac{(\text{Observed Accuracy} - \text{Expected Accuracy})}{(1 - \text{Expected Accuracy})} \tag{11}$$

4.3 Local Model Performance Analysis of The CTG Dataset

Local ensemble models RF_1, RF_2 and RF_3 are trained using the training data obtained from datasets D^1, D^2 and D^3 respectively. Each dataset is partitioned into training and test datasets with probability of 0.7 and 0.3 respectively. Training datasets used to train RF_1 and RF_2 have 634 instances each and remaining 266 for each model are in the test dataset. Training dataset obtained from D^3 used to train RF_3 contain 233 instances while the remaining 93 instances are in the test dataset. *Accuracy*, *f-measure* and κ of each local model is obtained using the random forest configuration settings as follows. Number of trees, $t = 300$ in the random forest model and number of variables tried at each split is equal to 8.

RF_1 achieves accuracy$\alpha_1 = 0{:}9135$ and the confusion matrix for classification obtained using RF_1 is given in Figure 1(a). Local ensemble model RF_2 getsaccuracy$\alpha_1 = 0.9398$. The confusion matrix for RF_2 is given in Figure 1(b). RF_3 achieves an accuracy of 100% with $\alpha_1 = 1.0000$. The confusion matrix for RF_3 is given in Figure 1(c). It can be observed that there are 93 test instances that are input to RF_3 with 81 instances belonging to class 1, just 1 instance in class 2 and only 11 instances in class 3. There is certainly problem of class imbalance in the test data and even if the only single instance of class 2 is misclassified, we can still achieve accuracy equal to 1. We used 266 test instances from D^2 and 93 test instances from D^3 and merged them together to create a new test dataset to test the performance of RF_3. As such, 359 instances with 318 instances in class 1, 16 instances in class 2 and 25 in class 3. It is in place to mention that RF_3 is trained using 233 training instances obtained from D^3. Accuracy of the RF_3 is 0.9471 and the confusion matrix is given in Figure 1(d).

The *f-measure* for the local models suggest that RF_1 perform better than rest of the models and both RF_2 and RF_3 perform nearly same. RF_3 trained with 359 instances is shown to have maximum accuracy but a careful examination of the corresponding confusion matrix reveals that this model performs quite bad classifying instances belonging to class 2. This fact is highlighted by calculating *f-measure* for the classifier. The *f-measure* and κ values for trained local models is given in Table 3. Insight into the comparative performance of all these local models using *f-measure* is further advocated by κ values for these classifiers. Cohen's kappa coefficient (κ) for each classifier given in Table 3 also suggest that performance of local model RF_1 is better than rest of the local ensemble models.

4.3.1 Local Model Parameter Tuning in CTG Dataset

Two important parameters that effect the performance of local random forest models are number of decision trees (t) in the random forest and number of attributes considered at each node of the decision tree.In order to analyze the effect of number of trees on error rate, all the local models are trained with *number of trees*, $t = 500$. The plot of *out of bag* (OOB) error rate along with the error rates for each class at varying number of trees for all the local models is given in Figure 2. It can observed from the figure that OOB error rate and error rates for each independent class start to stabilize with $t \geq 300$. Therefore, 300 trees for each RF_i in this case is sufficient because more number of trees do not enhance the performance of these local models further. Accuracy of each local model using $t =$

Figure 1. Confusion matrices of local and integrated models obtained from the CTG dataset; (a) Confusion Matrix of RF_1 obtained using test dataset from D^1, (b) Confusion Matrix of RF_2 obtained using test dataset from D^2, (c) Confusion Matrix of RF_3 obtained using test dataset from D^3, (d) Confusion Matrix of RF_3 obtained using test dataset from D^2 and D^3, and (e) Confusion Matrix of ζ_j obtained using test dataset with 235 instances.

ACTUAL

PREDICTED

	1	2	3
1	169	13	0
2	5	49	1
3	0	4	25

(a)

	1	2	3
1	231	8	2
2	5	7	0
3	1	0	12

(b)

	1	2	3
1	81	0	0
2	0	1	0
3	0	0	11

(c)

	1	2	3
1	315	15	1
2	0	1	0
3	3	0	24

(d)

	1	2	3
1	206	6	2
2	1	4	0
3	1	0	15

(e)

300 and $t = 500$ is recorded and the results are shown as bar chart in Figure 3. It can be observed from the figure that there is a very small difference between model accuracies obtained using 300 and 500 decision trees respectively.

Second important parameter that influences accuracy of the local ensemble random forest models is the *number of attributes* (*mTry*) considered at each node of the decision tree. The *tuneRF* function in R is used to search for the optimal *mTry* value till the relative improvement in OOB error is at least 0.5. The OOB error at different *mTry* values for local models is listed in Table 4. It can be observed from the table that OOB error vary with varying *mTry* values. RF_1 achieve lowest error rate with $mTry = 8$, RF_2 show lowest OOB error rate with $mTry = 4$, and finally, lowest error rate for RF_3 is recorded with $mTry = 8$. Therefore, a careful selection of *mTry* value can be used to enhance the performance of the local model. Figure 4 show the plot of *mTry* and OOB error for the local ensemble models.

Table 3. Performance measurement of local models

Model	α	*f-measure*	κ
RF_1	0.9135	0.8920	0.8237
RF_2	0.9398	0.7913	0.6794
RF_3	0.9471	0.7975	0.7026

4.4 Integrated Model Performance Analysis of The CTG Dataset

Local ensemble models RF_1, RF_2, and RF_3 are trained using 80% data instances in D^1, D^2 and D^3 respectively. Next, all the local models are tested with 237 test instances and the predictions using each model are recorded. Integrated model ζ_j running on party P_j use its own local model and local models from remaining participating parties to test each instance through all these ensemble models and bag the prediction results. The class with maximum number of votes is the predicted output of the final integrated model ζ_j for each test instance.

Confusion matrix of the predicted results obtained using the final integrated model for the test instances is given in Figure 1(e). Out of the total 237, 235 test instances are successfully classified using ζ_j. There are three local models and the data is multivariate with 3 possible classes. Therefore, 2 instances are not at all classified by ζ_j because no majority vote for any particular class is obtained for these instances. However, this tie rate for the test instances using ζ_j is 0.008439. Tie rates that tend to 0 are desirable in the integrated model.

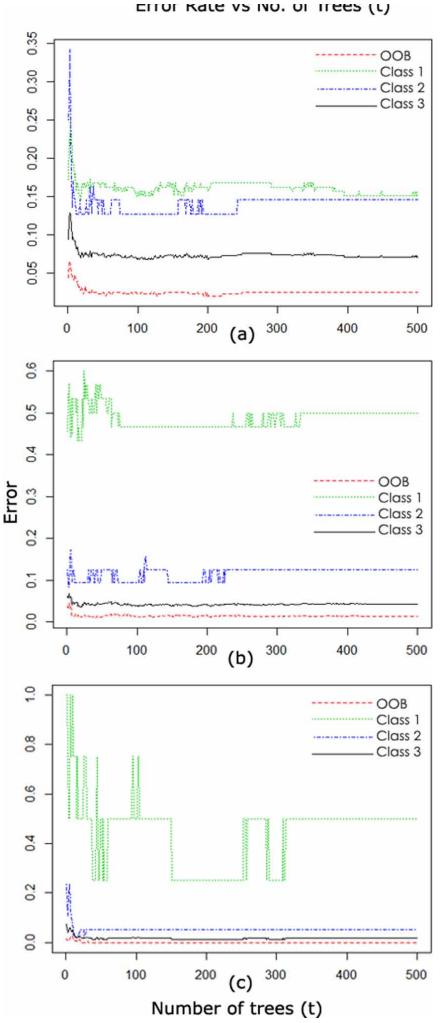
The performance metrics for the integrated model ζ_j are given in Table 5. The model achieves an accuracy of over 95% which is good. The f-*measure* for the classification done by the integrated model is 0.8230 which means that all the classes are fairly predicted correctly by the integrated model. The comparative performance on the basis of α, *f-measure* and κ of the integrated model with local ensemble models is shown as bar chart in Figure 5. It can be observed that the accuracy achieved using integrated model is higher than the accuracy achieved with each individual local model. The *f-measure* and κ metrics for the integrated model is also higher than these metrics for local models.

4.5 Local Model Performance Analysis of The TDD Dataset

The local ensemble models RF_i (where, $1 \leq i \leq 7$) are trained using the training data obtained from the respective datasets (D^i). Parties P_i use 100% data from the respective datasets (D^i) for training their local ensemble models RF_i. The total number of instances in the dataset and the number of instances belonging to a specific class are given in Table 2. The test data set named *Test* in Table 2 with 500 test instances is fed to each local model to analyze the performance of each model. The performance metrics *accuracy* (α), *f-measure* and κ for each local model is obtained using the following random forest configuration settings. Number of decision trees (t)= 500, and *number of variables tried at each split* (mTry)= 8.

The confusion matrices for local models RF_1 to RF_7 are given in Figure 6(a)-(g) and the performance metrics obtained for these models are given in Table 6. The performance metric values given in Table 6 shows an increasing trend with increase in dataset size ($|D^i|$) given in Table 2. Local models RF_1 and RF_2 achieve an accuracy of 99% and 99.2% respectively. The *f-measure* and *Kappa coefficient* values obtained for these models are also good indicating capabilities of these models to classify all the classes with relatively high accuracy. Local model RF_2 perform slightly better than RF_1 because RF_1mis-classify a single instance belonging to class 1 as an instance of class 2. Local model RF_3 performs poor with regard to classifying class 1 instances; 40% of the class 1 instances are misclassified by RF_3 as class 2 instances 6. This has an adverse effect on the overall performance of the local model RF_3. The values of all performance metrics decrease for local models RF_4 to RF_7. As discussed above, this decrease in classification performance can be attributed to the decrease in the size of the training datasets. The training dataset size for each local model is given in Table 2. For example, the training dataset (D^7) of the local model RF_7 has a total of 300 instances. D^7 has just 7 instances belonging to class 1, 7 instances belonging to class 2 and 286 training instances belonging to class 3. One can, therefore, expect the model to classify class 3 more accurately compared to classes 1 and 2. Figure 6(g) shows the confusion matrix obtained after classifying 500 test instances in the Test dataset. Local model RF_7mis-classifies 40% of the class 1 instances, 92.6% of the class 2 instances and only 1.5% instances belonging to class 3. The misclassification rate of class 2 instances by RF_7 is alarmingly high making it unsuitable for sensitive applications like medical data mining. However, poorly trained models like RF_7 can be improved by using integrated models as proposed

Figure 2. Effect of number of decision trees on error rate. (a) Error rate for RF_1, (b) Error rate forRF_2, (c) Error rate for RF_3.



in this work. The *f-measure* and *Kappa coefficient* (κ) values for $RF_7$are very low because of high misclassification rates for classes 1 and 2. There is a positive correlation between training dataset size and performance of the classifer as shown in Figure 7.

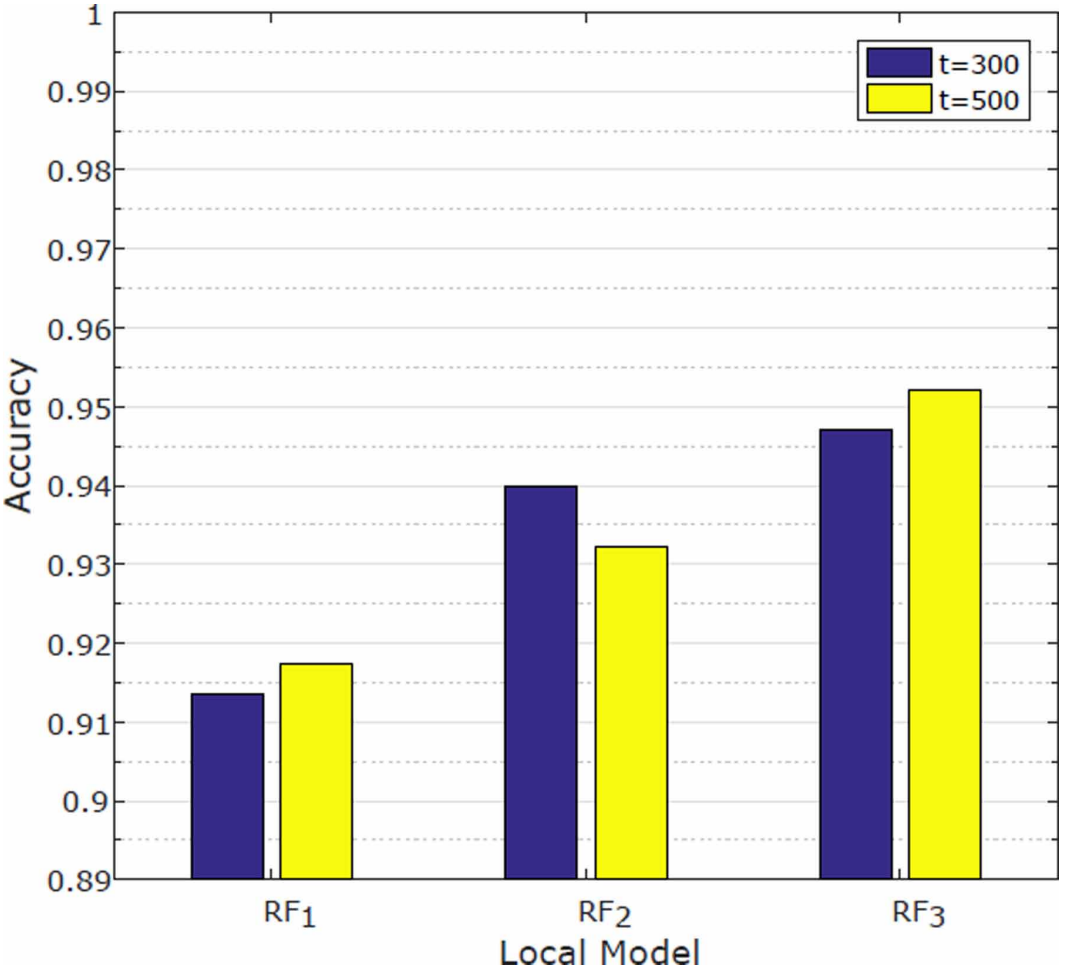
The performance of all the local models (RF_i) was checked with different permutations of *number of decision trees* (*t*) and the *number of variables tried at each split* (*mTry*) values and it was found that all the TDD local models perform best with $t = 500$ and $mTry = 8$.

4.6 Integrated Model Performance Analysis of The TDD Dataset

The local models RF_1 and RF_2 obtained from the TDD dataset perform very accurate classification overall as well as at the class level. However, local model RF_7 perform very poor classification at class level as discussed in Section *Local Model Performance Analysis of the TDD Dataset*.

An integrated model $\zeta_j = \{RF1, RF_2, RF_3, RF_4, RF_5, RF_6, RF_7\}$ is constructed by the party P_j, where $j \in \{x \mid 1 \leq x \leq 7\}$. The confusion matrix obtained by using the integrated model ζ_j to classify the test instances in the *Test* dataset is given in Figure 6(h). The performance metric values for the same *Test* dataset obtained using the integrated model ζ_j are given in Table 7. It can be observed from Table 6

Figure 3. Effect of number of decision trees (t) on local model accuracy



and Table 7 that the integrated model ζ_j shows a considerable improvement in performance compared to the local models RF_3, RF_5, RF_6 and RF_7. The integrated model ζ_j, however, fails to improve the performance of local models RF_1, RF_2 and RF_3 and more or less reduce their own performance. This is because of integrating poor performing models in the final integrated model. In order to avoid the negative impact caused by the integration of these poor models, the optimization criteria presented in the *Integrated Model Optimization* section are used. Moreover, the integrated model ζ_j fails to classify a single instance in the Test dataset because of a tie between majority votes for two or more than two classes. The tie rate however is quite low and equals 0.002. The bar chart of the performance of the local model and the integrated model is shown in Figure 8.

4.6.1 Integrated Model Performance Optimization of The TDD Dataset

Integration of local models to develop the integrated model ζ_j results in deteriorated performance of local models RF_1, RF_2 and RF_4. Therefore, it becomes important to adopt some optimization criterion to select and integrated models that make sure to improve the performance of the integrated model. In this section, performance of the integrated model ζ_4 developed by party P_4 by following the first

Table 4. Effect of mTry on local model performance

Local Model	*mTry*	OOB Error Rate
RF_1	2	10.73%
	4	7.57%
	8	6.78%
	16	6.78%
RF_2	2	4.26%
	4	3.79%
	8	4.73%
RF_3	2	2.58%
	4	2.15%
	8	1.72%
	16	2.58%

Figure 4. Effect of mTry on OOB error rate. (a) mTryvs OOB Error rate for RF_1, (a) mTryvs OOB Error rate for RF_2, and (c) mTryvs OOB Error rate for RF_3

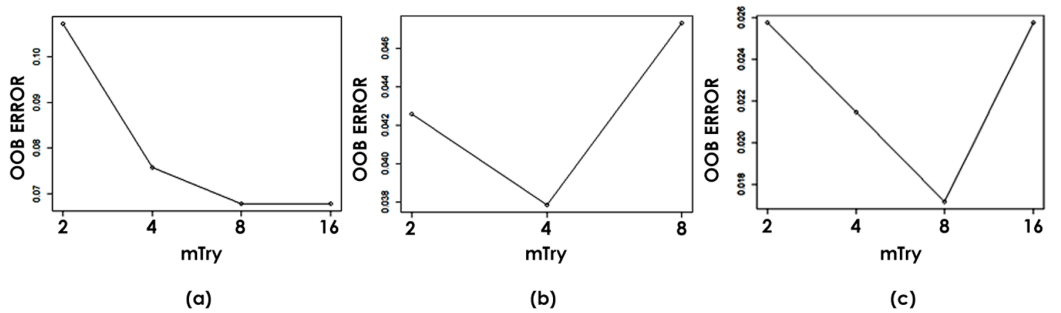


Table 5. Performance measurement of Integrated Model (ζ_j)

Model	ζ_j
α	0.9574
f-measure	0.8230
κ	0.7170

Figure 5. Performance comparison of CTG integrated model with local ensemble models

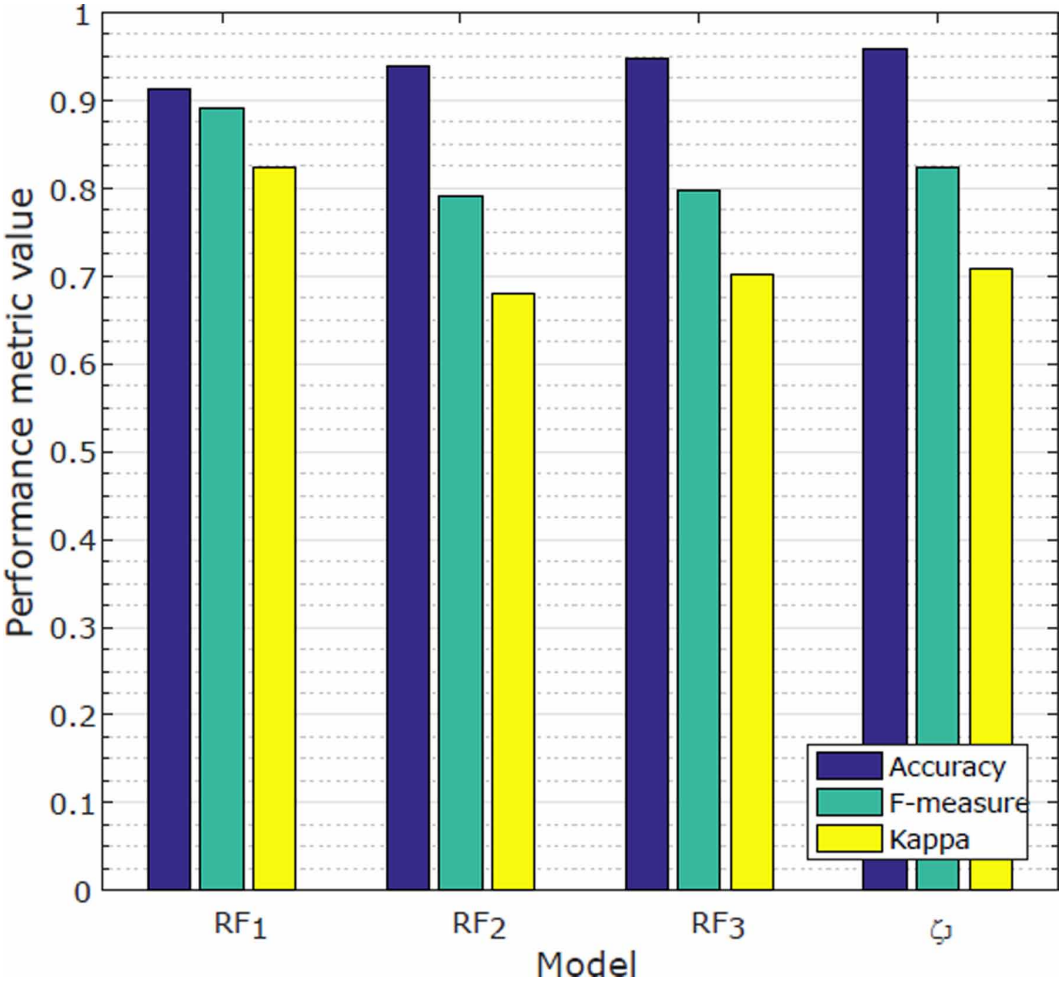


Table 6. Performance measurement of TDD local models

Model	α	*f-measure*	κ
RF1	0.990	0.9316	0.9433
RF2	0.992	0.9453	0.9658
RF3	0.982	0.8782	0.8548
RF4	0.988	0.9148	0.9200
RF5	0.982	0.8738	0.8663
RF6	0.964	0.7411	0.8279
RF7	0.982	0.3322	0.5653

optimization criterion presented in the *Integrated Model Optimization* section. The integrated model ζ_4 can thus be represented as given in eq. (11).

$$\zeta_4 = \{RF_1, RF_2, RF_4\} \tag{11}$$

Integrated model ζ_4 produces better performance metric values compared to ζ_j as given in Table 7. The comparison between all the performance metrics obtained from RF_4 and ζ_4 also suggest considerable improvement. $\zeta_4$classifies each test instance successfully and thus the tie rate equals 0 in this case.

4.7 Comparative Analysis of The Proposed Technique

Performance of the integrated model is by and large influenced by the performance of the constituent local models. Therefore, in order to enhance performance of the integrated models, it is imperative to strive for improving the performance of the respective local models. There are two major concerns that need to be addressed while choosing a classification learner for predicting sensitive data like the medical data. First, the learning model must achieve high performance while making predictions in the practical environment. Second, the model should be comprehensible, i.e. humans should be able to understand, what the models are doing, especially when they are responsible for the consequences of their application.

Learning models like the Neural Networks are trained models but are difficult to interpret. Neural networks in particular consist of hundreds to millions of different parameters depending on the size of the network, all interacting in a complex way. Lack of comprehensibility makes it complicated to use Neural Networks in areas where trustiness and reliability of the predictions are of great importance. Random Forests are also difficult to interpret because they consist of many (usually hundreds) of individual trees. Even if a single tree is easy to understand, the large number of trees makes the ensemble difficult to understand. More recently, however, approaches have been developed to identify the most representative trees in an ensemble (Banerjee et al. (2012)). By means of their analysis, the ensemble can finally be interpreted.

In general, ensemble random forest models perform better on simple data whereas neural networks show comparatively better performance with complex data. In this section, we present comparison between performance of the ensemble random forest model with the performance of some popular learners. The Support Vector Machine (SVM) model is used with linear and quadratic kernal functions. The second model used for comparison is the K-nearest neighbors model (KNN); the number of neighbors used by the Fine KNN model is 1 and Medium KNN uses 10 neighbors. Finally the ensemble boosted tree model (*AdaBoost*) is used with maximum number of splits set to 8 and number of learners equal to 300.The comparative performance of the learning models are evaluated using the CTG and the TDD datasets. 70% of the instances in each dataset are used to train the learning models whereas remaining 30% instances are used to test the performance of the models.

5. CONCLUSION

In this paper, a decentralized privacy preserving distributed random forest-based data mining technique is proposed to mine the sensitive health data maintained with different healthcare facilities without revealing any patient specific information in the process. A collaborative framework is proposed to take advantage of the knowledge obtained from the data of each healthcare facility to build an integrated model. Experimental results show that accuracy of the integrated model is better than the accuracy of the individual local models. The models are trained with small datasets and as such only three local models and thus the integrated model obtained using these three local models is used in the experimental study. A sufficiently large dataset and multiple local models can enhance performance

Figure 6. Confusion matrices of local and integrated models obtained from the TDD dataset; Figures (a-g) are the confusion matrices of local models RF_i ($1 \le i \le 7$), Figure (h) is the confusion matrix of global integrated model, and Figure (i) is the confusion matrix of integrated model $\zeta_4 = \{RF_1, RF_2, RF_4\}$

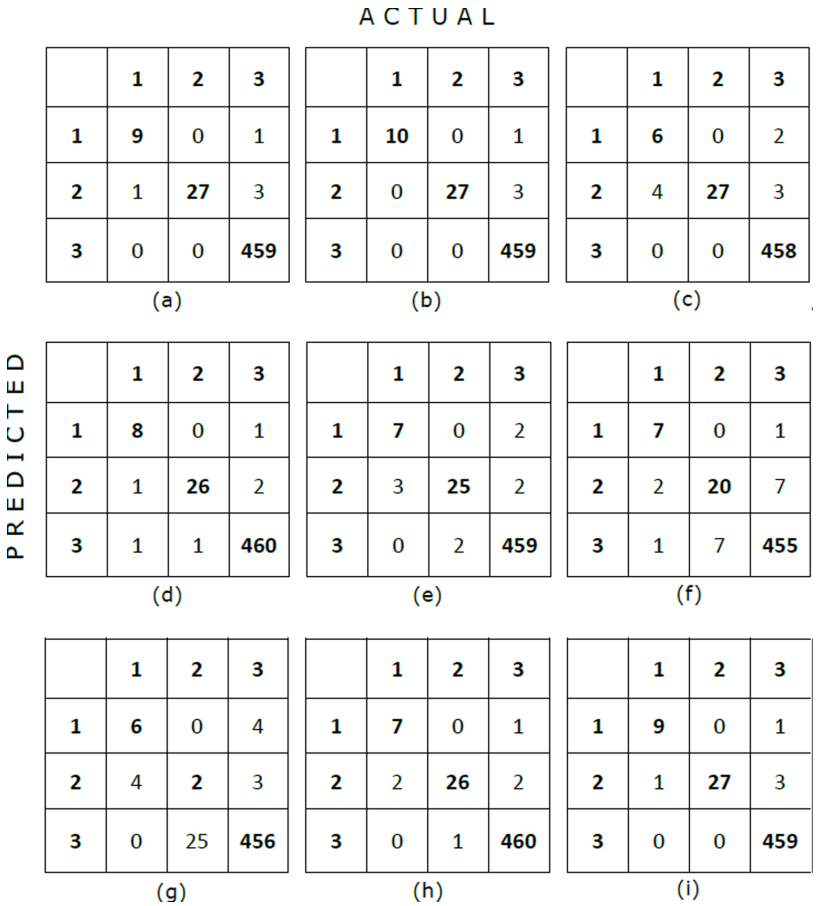


Table 7. Performance measurement of TDD integrated models

Model	α	*f-measure*	κ
ζ_j	0.9860	0.9137	0.9129
ζ_4	0.9900	0.9320	0.9340

of the proposed model further. In future, we intend to try different optimization criteria at both local as well as integrated model level to increase the performance of the model.

6. FUNDING SOURCE(S)

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Figure 7. Effect of the training dataset size on the local model (RF_i) performance

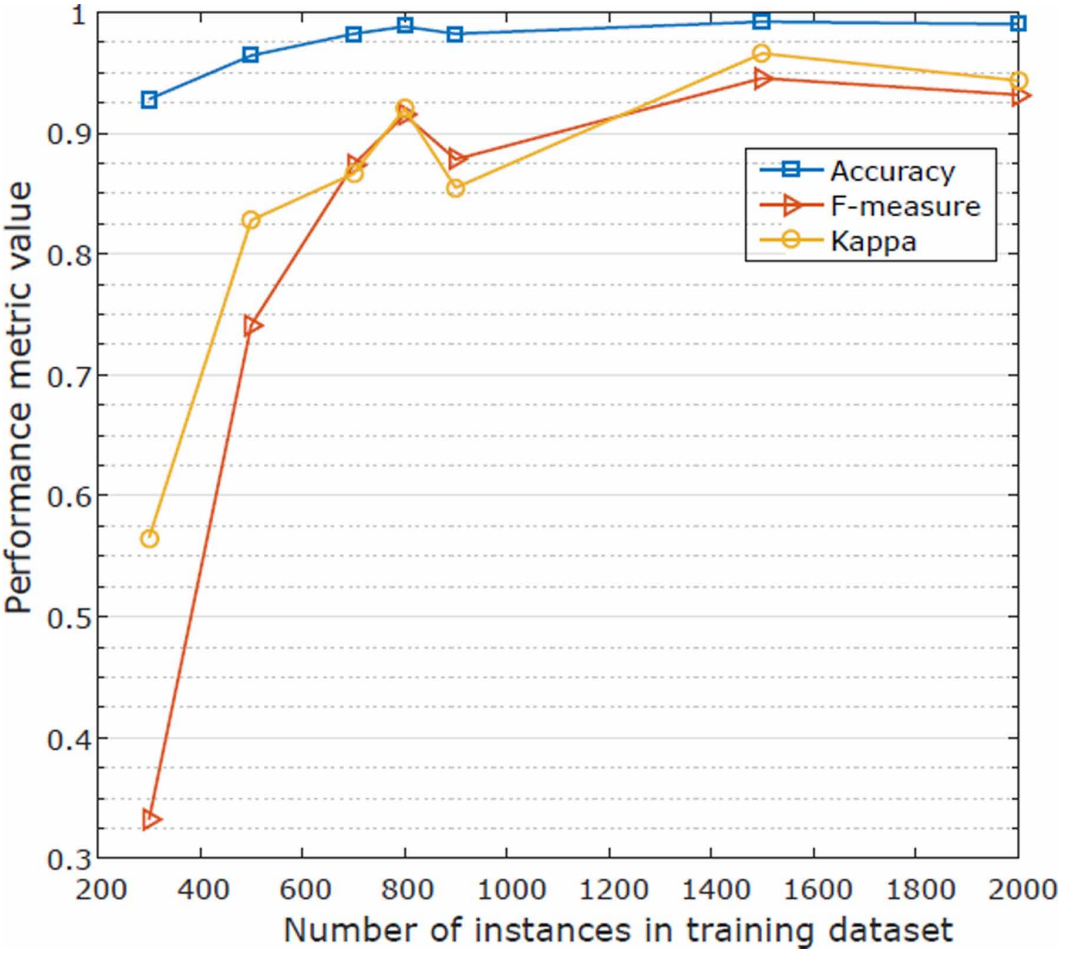
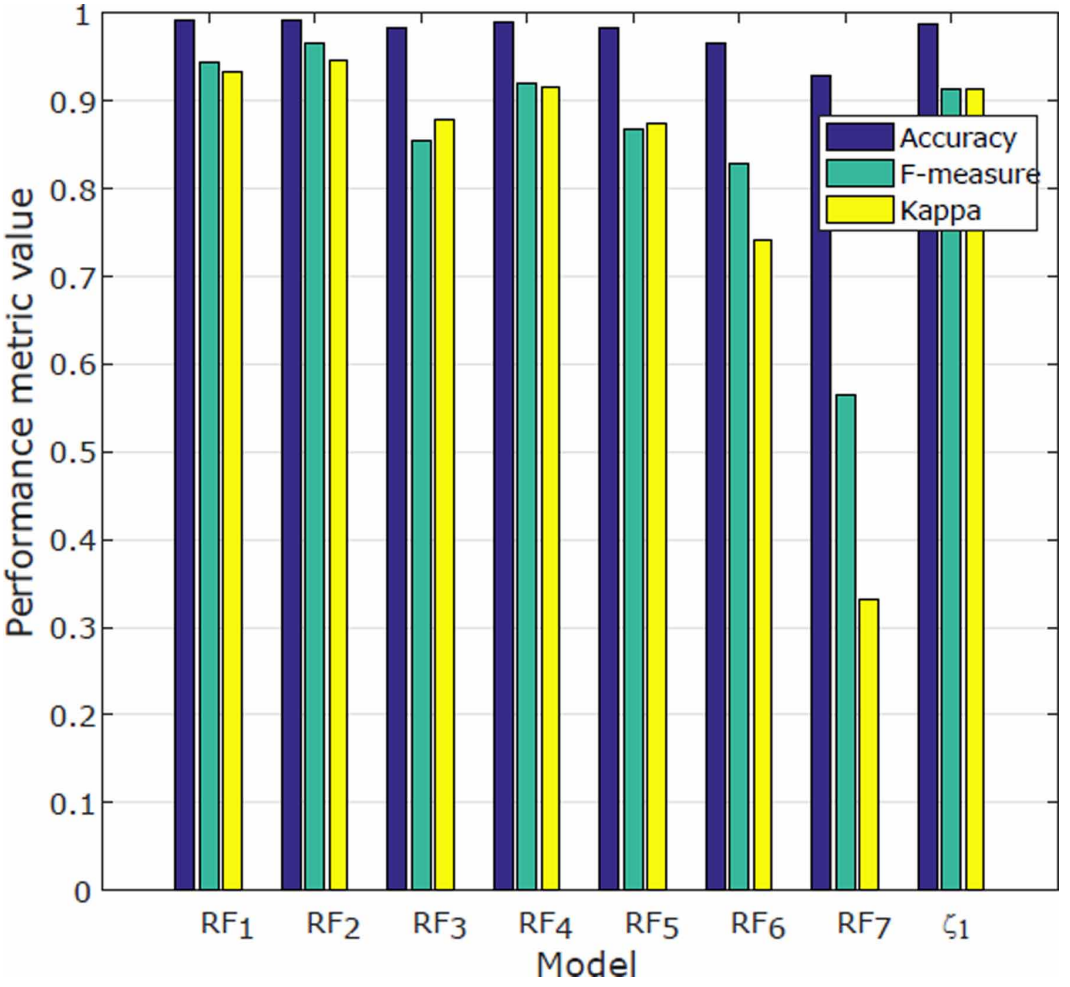


Figure 8. Performance comparison of TDD integrated model with local ensemble models



REFERENCES

Agarwal, C. C., & Yu, P. S. (2008). A General Survey of Privacy-Preserving Data Mining Models and Algorithms. *Privacy-Preserving Data Mining*, *34*, 11–52. doi:10.1007/978-0-387-70992-5_2

Agarwal, D., & Agarwal, C. C. (2001). On the design and quantification of privacy preserving data mining algorithms. *Proceedings of the twentieth ACM SIGMODSIGACT-SIGART symposium on Principles of database systems*, 247–255. doi:10.1145/375551.375602

Ahn, T., Kang, N., Kim, Y., Kim, S. I., Song, Y. S., & Park, T. (2018). Predicting survival outcomes in ovarian cancer using gene expression data. *International Journal of Data Mining and Bioinformatics*, *21*(4), 339–351. doi:10.1504/IJDMB.2018.098943

Banerjee, M., Ding, Y., & Noone, A. (2012). Identifying representative trees from ensembles. *Statistics in Medicine*, *31*(15), 1601–1616. doi:10.1002/sim.4492 PMID:22302520

Bashir, S., Qamar, U., Khan, F. H., & Javed, M. Y. (2014). MV5: A Clinical Decision Support Framework for Heart Disease Prediction Using Majority Vote Based Classifier Ensemble. *Arabian Journal for Science and Engineering*, *39*(11), 7771–7783. doi:10.1007/s13369-014-1315-0

Bialy, R. E., Salama, M. A., & Karam, O. (2016). An ensemble model for Heart disease data sets. *Proceedings of the 10th International Conference on Informatics and Systems- INFOS '16*, 191–196. doi:10.1145/2908446.2908482

Bisui, S., & Misra, S. C. (2019). Impact of Privacy Issues on Successful Implementation of Personalized Medicare System: An Empirical Study. *International Journal of E-Health and Medical Communications*, *10*(3), 96–115. doi:10.4018/IJEHMC.2019070106

Breiman, L. (1999). Pasting small votes for classification in large databases and on-line machine learning. *Machine Learning*, *36*(1–2), 85–103. doi:10.1023/A:1007563306331

Chawla, N. V., Hall, L. O., Bowyer, K. W., & Kegelmeyerand, W. P. (2004). Learning ensembles from bites: A scalable and accurate approach. *Journal of Machine Learning Research*, *30*, 421–451.

Cheon, J. H., Kim, D., Kim, Y., & Song, Y. (2014). Ensemble Method for Privacy-Preserving Logistic Regression Based on Homomorphic Encryption. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 46938–46948. doi:10.1109/ACCESS.2018.2866697

Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2004). Privacy preserving mining of association rules. *Information Systems*, *29*(4), 343–364. doi:10.1016/j.is.2003.09.001

Fienberg, S. E., & McIntyre, J. (2004). *Data Swapping: Variations on a Theme by Dalenius and Reiss*. Privacy in Statistical Databases.

Gambs, S., Kégl, B., & Aïmeur, E. (2007). Privacy-preserving boosting. *Data Mining and Knowledge Discovery*, *14*(1), 131–170. doi:10.1007/s10618-006-0051-9

Hassan, M., Butt, M. A., & Baba, M. Z. (2017). Logistic Regression Versus Neural Networks: The Best Accuracy in Prediction of Diabetes Disease. *Asian Journal of Computer Science and Technology*, *6*(2), 33–42.

Hassan, M., Butt, M. A., & Baba, M. Z. (2017). Privacy Preserving Data Mining for Healthcare Record: A Survey of Algorithms. *International Journal of Trend in Scientific Research and Development*, *2*(1), 33–42. doi:10.31142/ijtsrd7191

Huang, Z., Du, W., & Chen, B. (2005). Deriving private information from randomized data. *Proceedings of the 2005 ACM SIGMOD International conference on Management of Data - SIGMOD '05*, 37–48.

Kantarcıoglu, M., Vaidya, J., & Clifton, C. (2003). Privacy preserving naive bayes classifier for horizontally partitioned data. IEEE ICDM Workshop on Privacy Preserving Data Mining, 3–9.

Kou, G., Peng, Y., Shi, Y., & Chen, Z. (2004). Privacy-Preserving Data Mining of Medical Data Using Data Separation-Based Techniques. *Data Science Journal*, *6*, S429–S434. doi:10.2481/dsj.6.S429

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *2007 IEEE23rd International Conference on Data Engineering*, 106–115.

Lindell, Y., & Benny Pinkas, B. (2000). Privacy Preserving Data Mining. *Advances in Cryptology — CRYPTO 2000*, 36–54.

Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, A. (2006). L-diversity: privacy beyond k-anonymity. *22nd International Conference on Data Engineering (ICDE'06),* 24-24.

Meng, J., Jiang, D., Zhang, J., & Luan, Y. (2018). Ensemble classification for gene expression data based on parallel clustering. *International Journal of Data Mining and Bioinformatics*, *20*(3), 213–329. doi:10.1504/IJDMB.2018.094779

Sali, R., Shavandi, H., & Sadeghi, M. (2016). A clinical decision support system based on support vector machine and binary particle swarm optimisation for cardiovascular disease diagnosis. *International Journal of Data Mining and Bioinformatics*, *15*(4), 312–327. doi:10.1504/IJDMB.2016.078150

Samarati, P. (2001). Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, *13*(6), 1010–1027. doi:10.1109/69.971193

Samarati, P., & Sweeney, L. (2005). *Protecting privacy when disclosing information: kanonymity and its enforcement through generalization and suppression. Technical report*. SRI International.

Sheela, A., & Vijayalakshmi, K. (2017). Partition Based Perturbation for Privacy Preserving Distributed Data Mining. *Cybernetics and Information Technologies*, *14*(2), 44–55. doi:10.1515/cait-2017-0015

Tsoumakas, G., Angelis, L., & Vlahavas, I. (2003). Clustering classifiers for knowledge discovery from physically distributed databases. *Data & Knowledge Engineering*, *49*(3), 223–242. doi:10.1016/j.datak.2003.09.002

Yu, H., Jiang, X., & Vaidya, J. (2006). Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. *Proceedings of the 2006 ACM symposium on Applied computing - SAC '06*, 603–610. doi:10.1145/1141277.1141415

Musavir Hassan is a research scholar in the Department of Computer Sciences, University of Kashmir. She received M.Phil. in Computer Sciences in 2016 from University of Kashmir. Her current research interests include data mining and machine learning.