PRIVACY PRESERVING NAIVE BAYES CLASSIFIER FOR HORIZONTALLY PARTITIONED DATA USING SECURE DIVISION

Sumana M[1] and Hareesha K S[2]

[1]Department of Information Science and Engineering, M S Ramaiah Institute of Technology, Bangalore, Karnataka,India
[2]Department of Computer Applications, Manipal Institute of Technology, Manipal, Karnataka, India.

ABSTRACT

In order to extract interesting patterns, data available at multiple sites has to be trained. The data available in these sites should not be revealed while extorting patterns. Distributed Data mining enables sites to mine patterns based on the knowledge available at different sites. In the process of sites collaborating to develop a model, it is extremely important to protect the privacy of data or intermediate results. The features of the data maintained at each site are often similar in nature. In this paper, we design an improved privacy-preserving distributed naive Bayesian classifier to train the horizontal data. This trained model is propagated to sites involved in computation to assist classify a new tuple. We further analyze the security and complexity of the algorithm.

KEYWORDS

Privacy Preservation, Naive Bayesian, Secure Sum, Secure Division, Classification

.

1. INTRODUCTION

Distributed Computing environment allows sites to learn not only its own training dataset but also other sites training datasets. The outcome is considerably better than training at individual sites. Privacy concerns are large when sites collaborate in a distributed system[5,6]. One solution to perform this form of data mining is to have a trusted learner who builds a learning model by collecting all the data from the data holders [5,6,9,10]. However, in many real world cases, it is impossible to locate a trusted learner. Hence this approach is not considered feasible.

Researchers from various sectors such as medical, bank, security systems, finance are keen to obtain the result of cooperative learning without seeing the data available at other parties. For example, three banks in the same city want to know more information about the credit risk evaluation of the customers with the customer information they hold. These banks need to only communicate essential information during the training phase. After the training, the final model is broadcasted to the banks. The customer data held by the individual banks contain lot of private information such as age, marital status, annual wages and amount invested which are protected by law and cannot be revealed without the customer’s consent. In another situation, consider a medical research where doctors the different hospitals want to identify whether the right treatment is given for a medical diagnosis without revealing the individual patient’s details.

Our solution avoids revealing data beyond its attributes, while still developing a model corresponding to that learned on an integrated data set. Hence we assure that the data maintained at each of the sites are secure. In this paper, we propose privacy preserving Naive Bayesian classifier on horizontally partitioned data maintained at different sites. We handle both numeric and categorical attributes. Our method is based on performing addition using homomorphic encryption technique and uses a secure division protocol on these encrypted values. We have tested our protocol on real datasets.

Our main contributions can be summarized as follows:

- Enhanced privacy while computing the Naive Bayesian Classifier.
- Use of homomorphic property of Paillier cryptosystems to perform Secure sum.
- Use a Secure Division for Numeric and Categorical attributes of the dataset.

Researchers developed protocols to facilitate data mining techniques to be applied while preserving the privacy of the individuals. One approach[1] adds noise to the data before data mining. Agrawal and Srikant[5] proposed data perturbation techniques for privacy preserving classification model construction on centralized data. [1] Discusses building association rules from sanitized datasets. Such methods also called as data distortion methods assume that the values must be kept private from the data mining party. Also obtaining the exact results is a tedious process.

Another form of privacy preservation data mining uses cryptographic techniques to protect privacy. This approach includes secure-multiparty computations to realize perfect privacy. Methods for privacy preserving association rule mining in distributed environments were proposed by Kantarcioglu and Clifton[12]. [7][11][13] Constructs a classifier model using secure multiparty protocols. Classification using neural networks and preserving privacy is discussed in [14][15][16][20]. Another essential data mining tasks developed for privacy preservation has been discussed in [8].

Kantarcioglu and Vaidya [11] proposed a privacy-preserving naïve Bayes classifier for horizontally partitioned data. For the computation of probability p summations are computed by site1 adding a random number to its value and forwarding it to its neighbor. Other sites add their value to this value and forward it in a circular manner. The first site will interpret the result by subtracting the value received with the random value. Further to obtain the probability = $\sum i=1k$ $Pi/\sum i=1k$ Ci where k is the number of sites. Pi and Ci is the sum of values present at site i is computed by maintaining Pi in site 1 and Ci in last site and using the ℓn() protocol [9]. Though this protocol assumes no collusion among the sites, it is still vulnerable to the eavesdropping attack where any attacker who intercepts all transmissions among all sites is able to derive each site i's secret values. Also this protocol is not suitable if the number of sites $n< 3$.

In section II we briefly provide the background and related work required to develop our protocol. Section III discusses our algorithm. Security analysis of our protocol is elaborated in section IV.

2. PRELIMINARIES

2.1. Naïve Bayesian Classification

Naïve Bayesian Classifier [11] uses the Bayes Theorem to train the instances in a dataset and classify new instances to the most probable target value. Each instance is identified by its attribute set and a class variable. Given a new instance X with an attribute set, the posterior probability

P(Class1/X),P(Class2/X) etc has to be computed for each of the class variable values based on the information available in the training data. If P (Class1/X)>=P (Class2/X)>=……>=P (ClassN/X) for N class values, then the new instance is classified to Class1or Class2…or ClassN accordingly. This classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label y. The conditional independence can be obtained as follows: P(X|Y=y) = $\prod_{i=1}^{d} P$(Xi|Y=y), where each attribute set X = {X1,X2,…..,Xd} consists of d attributes.

Each of the d attributes can be categorical or numeric in nature. Algorithm 1 indicates the computation of the probability for a categorical attribute and Algorithm 2 indicates the computation of mean, variance and standard deviation required for calculating probability.

```
Algorithm 1 : Handling a categorical attribute
Input: r  -> # of class values, p -> #of attribute values
Cxy –> represents #of instances having class x and attribute value y.
Nx – > represents # of instances that belong to class x
Output: Pxy –> represents the probability of an instance having class x and attribute value y
For all class values y do
  {Compute Nx
    For every attribute value x
   {Compute Cxy
Calculate Pxy = Cxy/ Nx}}
Algorithm 2 : Handling  a numeric attribute
Input: r  -> # of class values, xjy -> value of instance j having class value y.
Sy -> represents the sum of instances having class value y
Ny -> represents # of instances having class value y
For all class values y do
{Compute Sy = $\sum_i x$jy
Compute ny
Compute Meany = Sy/ ny
Compute Vjy = (xjy – Meany ) 2 for every instance j  that belongs to the class y
Compute Varj= $\sum_j V$jy
Compute Stan_dev2y = Varj / (Ny-1)
}
```

Once the Variance and Standard Deviation is computed the probability for the numeric value provided in the test record for each of the class can be computed as follows:

P (given that (attribute_value = test_record_numeric_value)| Classy)

= 1$\frac{1}{\sqrt{2\pi} * Stan\ dev}$exp-$\frac{(test_record_numeric_value - Mean)}{2\ X\ Stan_dev\ (of\ class\ y)}$ y

On obtaining the Probabilities for each of the attributes with respect to each of the classes the class-conditional probabilities can be computed as follows:
For each of the class value I
Probability (test record having z attribute values | classI)= P(Attr1_value|classI) *P(Attr2_value|classI) *…….* P(Attrz_value|classI)
The test record belongs to the class has the maximum class-conditional probability.

2.2. Paillier Encryption

In our algorithms, a homomorphic cryptographic scheme of Paillier is utilized. This asymmetric public key cryptography [2,18,19] approach of encryption is largely used in privacy preserving data mining methods. The scheme is an additive homomorphic cryptosystem that are used in algorithms where secure computations need to be performed. Paillier is a public key encryption scheme which can be defined on any cyclic group. The original cryptosystem provides semantic security against chosen-plaintext attacks. Let G be a cyclic group of prime order q with generator g.

Key generation

Obtain two large prime numbers p and q randomly selected big integers and independent of each other such that gcd(pq,(p-1)(q-1)) = 1. Compute
n=pq and $\lambda = lcm(p \quad 1, q \quad 1)$
Select random integer g where $g \in Z^*_{n^2}$. Check whether n divides the order of g as follows
Obtain ℓ=((p-1)*(q-1))/gcd(p-1,q-1)
If(gcd(((g^{ℓ} mod n^2)-1)/n),n)!=1) then select g once again.

Encryption

Encrypts the plaintext m to obtain the Cipher text c = $g^m * r^n$ mod n^2 .where m plaintext is a BigInteger and ciphertext is also a BigInteger

Decryption

Decrypts ciphertext c to obtain plaintext m = L(g^{ℓ}mod n^2) * u mod n, where u = (L(g^{ℓ}mod n^2))^(-1) mod n.
Paillier schemes have probabilistic [19] property, which means beside the plain texts, encryption operation needs a random number as input. Under this property there can be many encryptions for the same message. Therefore no individual party can decrypt any message by itself.

2.3. Homomorphic Encryption

Homomorphic encryption is a form of encryption which allows specific computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers. Encryption techniques such as ElGamal[24] and Paillier[19]have the homomorphic property i.e for messages m1 and m2
D(E(m1,r1). g^{m2}) = m1+m2 mod n without decrypting any of the two encrypted messages.

Also D(E(m1,r1)*E(m2,r2) mod n^2) = m1 + m2 mod n.
D indicates decryption and E indicates encryption.

In our algorithms, a homomorphic cryptographic scheme of Paillier is utilized. This asymmetric public key cryptography approach of encryption is largely used in privacy preserving data mining methods. The scheme is an additive homomorphic cryptosystem that are used in algorithms where secure computations need to be performed. Paillier is a public key encryption scheme which can be defined on any cyclic group. The original cryptosystem provides semantic security against chosen-plaintext attacks. Let G be a cyclic group of prime order q with generator g.

Key generation

Obtain two large prime numbers p and q randomly selected big integers and independent of each other such that gcd(pq,(p-1)(q-1)) = 1. Compute
n=pq and λ= lcm(p-1,q-1)
Select random integer g where gϵ $Z^{*}_{n}{}^{2}$. Check whether n divides the order of as follows
Obtain ℓ =((p-1)*(q-1))/gcd(p-1,q-1)
If(gcd(((g^{ℓ} mod n^2)-1)/n),n)!=1) then select once again.

Encryption

Encrypts the plaintext m to obtain the Cipher text c = $g^m * r^n$ mod n^2 .
where m plaintext is a BigInteger and ciphertext is also a BigInteger

Decryption

Decrypts ciphertext c to obtain plaintext m = L(g^{ℓ}mod n^2) * u mod n, where u = (L(g^{ℓ}mod n^2))^(-1) mod n.
Paillier schemes have probabilistic [10] property, which means beside the plain texts, encryption operation needs a random number as input. Under this property there can be many encryptions for the same message. Therefore no individual party can decrypt any message by itself.

2.4. Secure Multiparty Protocols

To solve our problem of secure computation[11] we have used secure protocols for computing the sum and divide. Some of the secure computations have been discussed in [3]. The parties could apply the algorithm to add two values maintained by them without revealing their values to other parties. This protocol has been implemented by utilizing cryptographic schemes with the additive homomorphic property. Secure Division is performed by a single party with the numerator and the denominator in their encrypted form. A detailed description regarding the usage of these protocols is discussed in the next section.

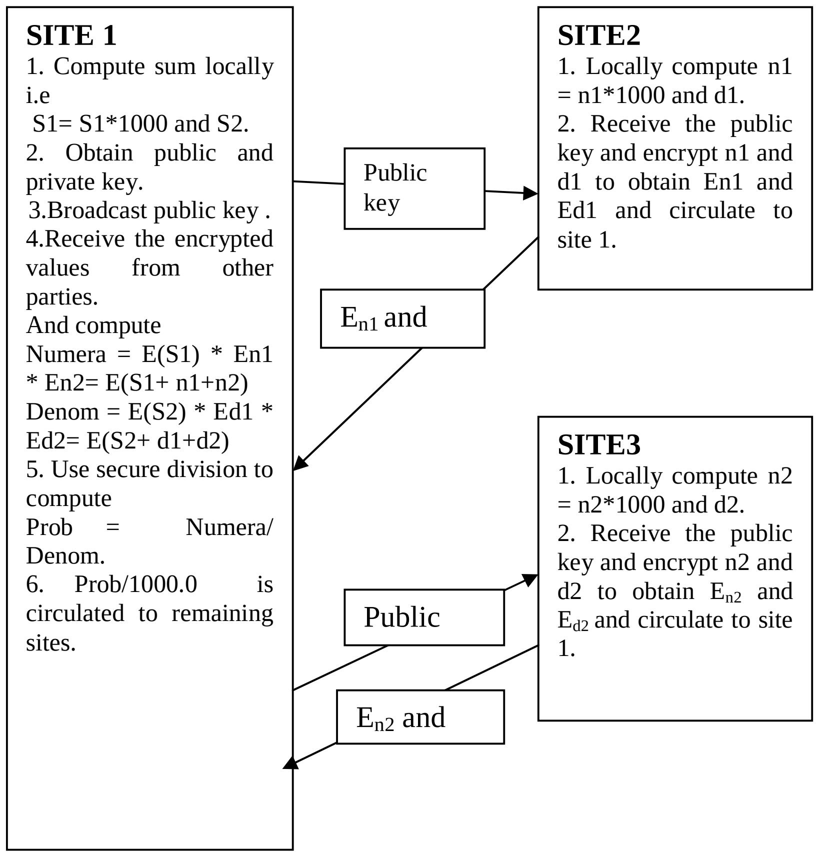
3. MODEL CONSTRUCTION

In this paper we focus on secure training of a horizontally partitioned dataset to build a Naïve Bayesian Classifier model. This constructed allows each of the party to classify a new instance.
Multiple banks hold information about the age, Class of Worker , education ,wage per hour, marital stat, major industry code, major occupation code, race, sex ,full or part time employment stat, capital gains, capital losses, dividends from stocks, tax filer stat, region of previous residence, state of previous residence ,detailed household and family stat, num persons worked for employer, family members under 18, country of birth self, citizenship, own business or self employed, veterans benefits, weeks worked in year . This information is collected from people residing in the locality that the banks exist. The characteristics of the individual are either numeric or nominal in nature. Each of the banks has thousands of records holding the information. In order to conclude on a loan decision salary of a person is an important data. Two or more banks want to predict the salary of an individual based on the age, Class of Worker, tax filer status, marital status, qualification, residing region and number of persons in the family. But these banks want to disclose the result of their computation without revealing any other information to a third party or to each other. The above task can be performed by training the horizontally partitioned data in a secure manner.

The protocols presented below are efficient and do not compromise on security. In the process, even the numerator and the denominator of the fractions are not known to any of the parties. Secure Division is performed in a single site. In the following sections we discuss the approaches where only the final classifier is broadcasted to all the parties.

Since all the attributes needed for classifying a new instance are known to all the parties we need not hide any of the attributes or their values. Hence once classifier is given to all the parties, parties need not collaborate to classify a new instance. Also we need not conceal the model. Figure 1 provides a scheme of building the model for 3 parties.

Figure 1: Model Building Process using 3 parties


SITE 1
1. Compute sum locally i.e
S1= S1*1000 and S2.
2. Obtain public and private key.
3.Broadcast public key .
4.Receive the encrypted values from other parties.
And compute
Numera = E(S1) * En1 * En2= E(S1+ n1+n2)
Denom = E(S2) * Ed1 * Ed2= E(S2+ d1+d2)
5. Use secure division to compute
Prob = Numera/ Denom.
6. Prob/1000.0 is circulated to remaining sites.
Public key
E_{n1} and
Public
E_{n2} and
SITE2
1. Locally compute n1 = n1*1000 and d1.
2. Receive the public key and encrypt n1 and d1 to obtain En1 and Ed1 and circulate to site 1.
SITE3
1. Locally compute n2 = n2*1000 and d2.
2. Receive the public key and encrypt n2 and d2 to obtain E_{n2} and E_{d2} and circulate to site 1.

In this section we discuss the methods for constructing models for both categorical and numerical attributes. Since the procedures for learning is dissimilar for both the types of attributes, we define different methods for each.

3.1. Categorical Attributes

For categorical attributes, the conditional probability has to be computed. Conditional probability gives the probability that an instance belongs to a class ‘c’ for an attribute A having an attribute value ‘a’ indicated asP(C= ‘c’/A = ‘a’)= n_{ac}/n_a

where n_{ac} – number of instances in the training set(in all the collaborating parties) that have the class value ‘c’ and value of the attribute value as ‘a’ and n_a - number of instances(in all the collaborating parties) where attribute A = ‘a’.

As the datasets are horizontally partitioned, parties are aware of some or all of the values of their categorical attributes. To compute the sum n_{ac} and n_a for all the parties, each party locally counts the number of instances and then parties collaborate to use the paillier homomorphic secure sum protocol (algorithm 5) to compute the global count. During collaboration, local counts are not revealed to any of the intermediate parties. The party that has initiated the training phase has the encrypted results (both numerator and denominator).These encrypted values are then securely divided to obtain the probability. As observed in algorithm 3 we multiply the numerator by 1000 and further divide the result with 1000 to round the result obtained by 3 decimal points.

Algorithm 3: Handling a categorical attribute

```
Input: k parties, r class values, n attribute values
$C^i_{ac}$ – number of instances with party $P_i$ having class c and attribute value a.
$n^i_c$ – number of instances with party $P_i$ having class c.
$p_{ac}$ - Probability of an instance having class c and attribute value a.
for all class value c do
   for i= 1 to k do  // for each party
    for every attribute value a, party $P_i$ locally computes $C^i_{ac}$ .Then Perform $C^i_{ac}= C^i_{ac} * 1000$.
    Party $P_i$ locally computes $n^i_c$ // local computation by each party
 end for
end for
 All parties collaborate using secure sum protocol to obtain $EC_{ac} = E(\sum_{i=1}^{k} C^i_{ac})$ .
For every class value c, all parties collaborate using secure sum protocol,$En_c = E(\sum_{i=1}^{k} n^i_c)$.
Party 1 which initiated the model construction computes $p_{ac}$ using the $EC_{ac}$ and $En_c$ using secure division protocol(algorithm 6) . Final $p_{ac} = p_{ac} / 1000$.
```

3.2. Numeric Attributes

For numeric attributes the mean value has to be securely computed. Mean value of a class ‘c’ = S_c/n_c , where S_c is the sum of all the instances in the multiple parties belonging to class ‘c’ and n_c is the number of instances belonging to class ‘c’.

All parties at first locally compute the mean of its numeric attribute value. They also obtain the sum of all the instances that belong to the class ‘c’. Further algorithm 5 is used to find the encrypted result of the global sum of S_c and n_c. Algorithm 4 is used to give the mean of instances belonging to class ‘c’.

Using this mean the parties then collaboratively calculate variance.

Algorithm 4 : Handling a Numeric Attribute

```
Input : k parties, r class values
$x_{icj}$ - the values of instances j from party i having class value c
$s^i_c$ -  the sum of instances from party i having class value c
$n^i_c$ - the number of instances with party $P_i$ having class value c
for all the class values  c do
   for i= 1 to k do
```

```
            Party $P_i$ locally computes $s^i_c = \sum_j x_{icj}$. Performs $s^i_c = s^i_c * 1000$.
            Party $P_i$ locally computes $n^i_c$
   end for
All parties in collaboration perform secure sum protocol to compute $E(s_c) = E(\sum_{i=1}^{k} s^i_c)$ .
All parties in collaboration perform secure sum protocol compute $E(n_c) = E(\sum_{i=1}^{k} n^i_c)$
Party 1 computes the mean $\mu_c = E(s_y)/E(n_y)$ using secure division protocol.
mean $\mu_c = \mu_c/1000$ ;
end for
$\mu_c$ is circulated to all the other sites.
//to compute total variance
for i=1 to k do
  for every instance j, $v_{icj} = x_{icj} - \mu_c$ and $v_{ic} = \sum_j v^2_{icj}$
end for
All the parties then collaborate using secure sum protocol to compute variance
$E(v_c) = E(\sum_{i=1}^{k} v_{ic})$
$D(E(v_c))$ is performed by party 1 to obtain $v_c$.
Finally party 1 computes stan_dev $\sigma^2_c = \frac{1}{nc-1} . v_c$
```

3.3. Secure Sum Protocol

This algorithm is used to securely compute the sum of the values maintained at individual sites. This protocol makes use of the homomorphic property of Paillier's

Algorithm 5: Secure Sum Protocol

Party P_1 uses randomgenerator to obtain a random number r_1, uses Paillier cryptosystem to obtain public key P_k. This public key is circulated to all the other parties. All the other parties uses this public key to encrypt its value S_i to $E(S_i,r_i)$.

```
for i= 2 to k
Use RandomGenerator to obtain the random number $r_i$ .
Uses the public key to obtain $E(S_i,r_i)$,
and forwards it to party $P_1$.
end for
```

Party P_1 finally computes Encrypt_prod $= \pi^{k}_{i=1} E(S_i,r_i)$.

Note: $\pi^{k}_{i=1} E(S_i,r_i) = E(S_1+S_2+S_3+\ldots\ldots+S_k)$.

3.4. Secure Division

Since the numerator (n) and the denominator (d) are in the encrypted form we use this method. The encrypted values are of BigInteger type that exceeds the size of 512 bits. The Logic used is the working [23] is as follows:

I. Compute an encrypted approximation [a~] of a = $[2^k/[d]]$

II. Compute [n/d] as $([a\sim]*[n])/ 2^k$.

To compute the k shift approximations of 1/d we use the concept of Taylor's series to define the desired approximation of $2^k/d$ as

$a\sim = 2^{k-\ell d(w+1)} * \sum_{i=0}^{w} (2^{\ell d} - d)i * 2^{\ell d(w-i)}$.

Further we compute using Z_M arithmetic, with $M = p * q$, which is the Paillier key whose secret key is held jointly by the parties. Hence a~ is modified as follows

$a\sim = 2^{k-\ell d} * \sum_{i=0}^{w}((2^{\ell d} - d)*2^{\ell d})^{i}$.
Algorithm 6 discusses the secure division protocol. This protocol is being executed at a site with no communication with other parties.

Algorithm 6: Secure Division on encrypted values

```
Input: Encrypted numerator [n] and encrypted denominator [d](ℓ-bit value)
1. Compute $2^{\ell d}$ from [d]
 count =1
obtain binary representation of [d]. Initialize $p_0$=1
while(count<=$\log_2 \ell$)
begin
c1 =0
if( $2^{\ell/2}.p_0$ <= [d])
c1 = 1
$p_0 = p_0*(c1*(2^{\ell/2}-1)+1)$
end
compute $2^{\ell d} = 2*p_0$.
2. Obtain $2^{-\ell d}$ = Inverse($2^{\ell d}$)
3. Obtain Poly (p) for p = $(2^{\ell d}-d) * (2^{-\ell d})$ as follows
Use square and multiply method to evaluate $\sum_{i=0}^{w} p^{i}$ where w = $2^R$ for some integer R.
4. Compute a~ = $2^k * 2^{-\ell d}$ * Poly (p).
5. Further we find q^ = [n]. a~
6. Truncate q^ by k to acquire q~ is approximately equal to $(q/2^k)$ as follows
Obtain [z] -> q^+ r , where r is a random number $\in Z_2^{k+\ell}$.
Decrypt [z] and generate q~ = $(z/2^k)-(r/2^k)$
7. Eliminate errors generated as follows
r = [n]-[d]* q~
if([d]+[d]<=r+[d])
pos_err = 0.1 else pos_err = 0.0
if([d] >r+[d])
neg_err = 0.1 else neg_err=0.0
8. Finally compute q <- q~ + pos_err + neg_err.
```

3.5. Classifying an instance

As we have implemented our protocols for horizontally partitioned dataset all the attributes are known to all the parties. The party that wants to evaluate an instance simply uses the probability values obtained for categorical attributes, mean and variance computed for numeric attributes and locally classifies it. It need not interact with the other parties. Hence there is no compromise in privacy.

4. SECURITY ANALYSIS

In this section we elaborate on why our algorithms are secure in the semihonest model. In the semihonest model, the parties during computation are curious and try to analyze the intermediate values and results. Hence in a secure model we must show that the parties learn nothing except their outputs from the information they obtain during the process of execution of the protocol. The encryption scheme, Paillier, used in the protocol is semantically secure as the result each ciphertext can be simulated by a random ciphertext.

Algorithm 1 securely computes the probability p_{ac} without revealing anything (i.e. either the global count C_{ac} or global number of instances n_c). The only communication occurs while computing the global sum using homomorphic encryption. When there is no collision between the parties each party's view of the protocol is simulated based on its input and its output. Algorithm 2 securely computes the mean μ_c and variance v_c without revealing anything except μ_c and v_c . The communication in this algorithm occurs while computing the global sum while mean and variance but the global sum is not revealed to any of the parties.

In algorithm 1and 2 parties 2 to k communicate with each other with their encrypted values and multiply and forward it to their neighboring party to obtain the encrypted global sum. Party 1 performs an additional step of computing the division of Paillier encrypted values. After the secure division protocol party 1 sees only the result of division which is broadcasted to the other parties.

Even though the public key is known to all the parties and each of the parties encrypt their data to assist in computation because of the probabilistic property of Paillier parties cannot decrypt the other parties' data. Hence we propose that our approach is secure. As mentioned in [23] the secure division protocol does not reveal any information about the inputs(other than the desired encryption of the result).

4.1. Effect of Collusion on Privacy

In our solution, in the process of secure sum additions involving k parties, if C_{ac} and n_c can be evaluated even if k-1 parties collude with each other. However if all of the k parties collude, privacy protection is irrelevant.

For the secure division protocol, since only 1 party performs the computation colluding of the other parties will not affect the protocol. Also if party 1 colludes with the other parties, it only has the encrypted values hence it cannot reveal anything to the other parties.

4.2. Communication and Computation Cost

The secure division protocol requires only $O((\log^2 \ell)(\alpha + \log\log \ell))$ arithmetic operations in $O(\log^2 \ell)$ rounds where α is the correctness parameter and ℓ is the size of the numerator and denominator. The computation of $2^{\ell d}$ for encrypted d requires $\log_2 \ell$ iterations, each involving one comparison and one multiplication. Hence the complexity is $O(\log^2 \ell)$. The round complexity of poly (p) is O (log w) where $w = 2^{\lambda}$ approximately equal to ℓ. Further the round complexity of truncating is $O(\log \ell)$.

For calculating conditional probability privately we require k secure additions and one division for k parties. Compared to non-secure version of the conditional probability calculation the secure version is much slower. Computation using homomorphic secure sum protocol involves only k encryptions and k summations; hence the computation cost is dominated by the secure divide protocol. Given a dataset having n_1 categorical attributes with an average of n_a values, the number of global computations performed are $2*(n_{1*} n_a)*$ k secure additions and $(n_{1*} n_a)$ secure divisions by party 1. For n_2 numeric attributes, global computations are $3*(n_2)*k$ secure additions and n_2 secure divisions by party1. The local computations of sum performed by each of the party's depend majorly on the number of tuples they have. We have implemented our approach with n sites Intel(R) core ™ 2 CPU, 6400 @ 2.13GHz, 2GB ram with a Java program to enable the n sites to interact with each other during secure sum computation.

Given in Table 1 is the computation time for calculating conditional probabilities worked on the census dataset (Salary as class attribute, with occupation, education, marital status, dependency as categorical attributes and age, capital gains as numeric attributes) and breast cancer (Diagnosis as class attribute, all 8 attributes are numeric in nature) from the UCI repository. The table summarizes the approximate computation time for conditional probabilities for different database sizes. The time required to classify a new instance is the same as that in a non-privacy version of the classifier.

For test samples Table 2 indicates the accuracy of our approach. Accuracy is computed as the total number of correctly classified tuples divided by the total number of tuples in test sample.

Table 1: Estimated Computation Time for conditional probabilities

Security Parameter (in bits)	Total tuples in all sites	Degree of the Polynomial	Estimated Time (seconds) Census Dataset	Estimated Time (seconds) Breast Cancer Dataset
512	10^5	10	0.68	0.72
512	10^5	20	0.71	0.75
512	10^6	10	0.84	0.90
512	10^6	20	0.88	0.94
512	10^7	10	0.91	0.96
512	10^7	20	0.95	1.06
1024	10^5	10	2.75	2.83
1024	10^5	20	2.86	2.92
1024	10^6	10	3.51	3.69
1024	10^6	20	3.72	3.81
1024	10^7	10	4.56	4.62
1024	10^7	20	4.64	4.78

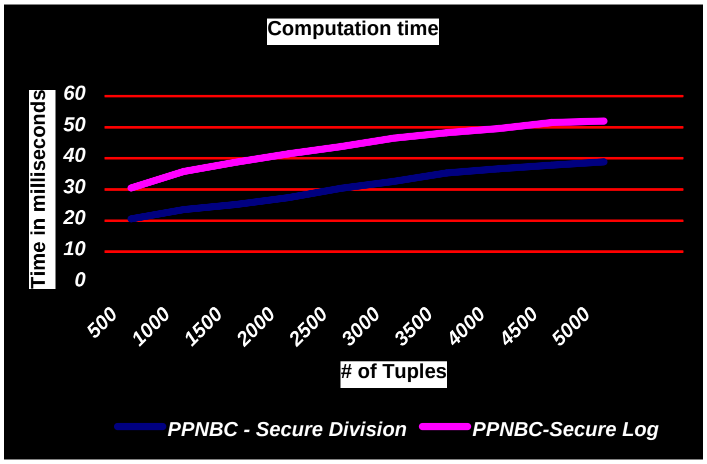
Table 2: Accuracy of the classifier

Size of test samples	Accuracy(%)
10^3	83
10^4	85
10^5	87

4.3. Performance Comparison

On comparison with Vaidya[11] our protocol provides better computational time and accuracy as shown in figure 2 and 3. Better computation time is because of the secure division performed in a single site rather than a distributed manner of computing division.

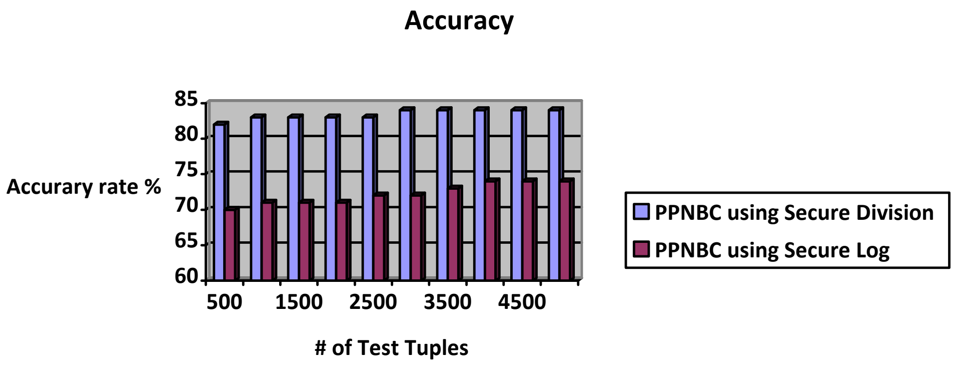
Figure 2: Computation time



The accuracy of a classifier is defined as

$$\text{Accuracy} = \frac{\textit{Number of test tuples properly classified}}{\textit{Number of test tuples properly classified} + \textit{Number of test tuples misclassified}} * 100$$

Figure 3 : Accuracy



5. Conclusion

This paper concentrates on building a secure Naïve Bayesian classifier with multiple parties without revealing any information during summation and division. It uses the homorphic property of Paillier to compute the sum. With the Paillier encrypted secure division is performed to obtain probability or mean. The probability, mean and variance obtained securely are circulated to all the parties for classifying a new instance.

Our approach even though expensive than the non-privacy version of the protocol thrives to achieve a model that is secure and efficient. The algorithm guaranteed privacy in a standard cryptographic model, the semi honest. In future we intend to explore privacy preservation approaches for other classifiers.

REFERENCES

[1] L. Wan,W.K.Ng,S.Han, and V.C.S.Lee,"Privacy Preservation for gradient descent methods,"in Proc.ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,2007,pp.775-783.

[2] O.Goldreich, "Foundations of Cryptography".Cambridge, U.K.:Cambridge Univ. Press, 2001-2004, vol. 1 and 2.

[3] A.Yao, "How to generate and exchange secrets," in Proc. 27th IEEE Symposium Foundations, Computer Science., 1986, pp. 162-167

[4] T.ElGamal,"A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions of Information Theory, vol.IT-31, no.4, pp.469-472, Jul.1985.

[5] D.Agrawal and R.Srikant,"Privacy-preserving data mining," in Proc. ACM SIGMOD, 2000, pp.439-450.

[6] Y.Lindell and B.Pinkas, "Privacy preserving data mining," in Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, 2000, pp.36-44.

[7] N.Zhang, S.Wang, and W.Zhao, " A new scheme on privacy-preserving data classification, " in Proc. ACM SIGKDD International Conference of Knowledge discovery and data mining,2005,pp.374-383.

[8] G. Jagannathan and R.N. Wright,"Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," In Proceedings of ACM SIGKDD International Conference of Knowledge discovery and data mining 2005,pp 593-599.

[9] Y.Lindell and B.Pinkas, "Privacy preserving data mining," Journal of Cryptography, volume 15, no 3,2002, pp177-206.

[10] R Agrawal , R Srikant: "Privacy-preserving data mining" In Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, pp.439-450. ACM, Dallas, TX(2000). http://doi.acm.org/10.1145/342009.335438.

[11] J Vaidya.,M Kantarcioglu,C Clifton.,: Privacy Preserving Naïve Bayes Classification. In: The VLDB Journal (2008) 17: 879-898, DOI 10.1007/s00778-006-0041-y.

[12] M Kantarcioglu, C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, IEEE TKDE 16(9) (2004).

[13] S. Samet, A. Miri, Privacy Preserving ID3 using Gini Index over horizontally partitioned data ,In: Proceeding of the 6th ACS/IEEE International Conference on Computer Systems and Applications(AICCSA), Doha, Qatar, 2008,pp.645-651.

[14] R. Wright and Z. Yang , " Privacy-preserving Bayesian network structure computation on distributed heterogeneous data," in Proc. 10th ACM SIGKDD Int. Conf.knowledge.Disc. Data Mining, 2004,pp.713-718.

[15] A. Bansal, T Chen, S Zhong., Privacy Preserving Back-propagation neural network learning over arbitrarily partitioned data, Neural Computing and Appications(2011) 20: 143-150.

[16] T. Chen and S Zhong., Privacy-Preserving Backpropagation Neural Network Learning, IEEE Transactions on Neural Networks, Vol., 20, No10, 2009.

[17] Mitchell, T.: Machine Learning, 1st edn. McGraw-Hill Science/Engineering/Math, New York(1997).

[18] P.Paillier, Public-key cryptosystems based on composite degree residuosity classes, In: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic,1999,pp. 223-238.

[19] Blum, M., Goldwasser, S.: An efficient probabilistic public-key encryption that hides all partial information. In: R. Blakely(ed.) Advances in Cryptography-Crypto 84 Proceedings, Springer, Heidelberg(1984).

[20] S. Samet, A.Miri, Privacy Preserving back-propagation and extreme learning machine algorithms, Data and Knowledge Engineering, 79-80(2012) 40-61

[21] Blake CL, MErz CJ(1998) UCI Repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA. http://www.ics.uci.edu/mlearn/MLRepository.html.

[22] O Goldreich " The Foundations of Cryptography", vol.2, chap. General Cryptographic Protocols. Cambridge University Press, Cambridge University Press, Cambridge (2009).

[23] M Dahl, Chao Ning, T Toft, "On Secure Two-party Integer Division", Financial Cryptography and Data Security ,Lecture Notes in Computer Science, Volume 7397, 2012, pp 164-178.

[24] ElGamal T.,"A Public key cryptosystem and a signature based scheme on discrete logarithms," IEEE Transactions, Inf.Theory, vol.IT-31,no.4,pp.469-472, Jul 1985.