NAVIE BAYESIAN CLASSIFICATION FOR PRIVACY-PRESERVING PATIENT-CENTRIC CLINICAL DECISION SUPPORT SYSTEM

Jebas Sinthiya. I
PG Scholar
Computer Science and Engineering
Anna University Regional Campus

Sanjeeve Kumar. S
Assistant professor
Computer science and Engineering
Anna University Regional Campus

Abstract— Clinical decision support system is a disease diagnosing tool that helps clinician to make accurate decision. An expansive measure of information is created each day. Naive Bayesian, a data mining technique is utilized to unearth and classify the required information. The existing experiences more hazard including data security and protection. A new clinical decision support system is proposed to help doctor to diagnose the risk of patients' disease in a privacy-preserving way. The past patients' historical health data are stored and can be used to train the Naive Bayesian classifier without leaking any individual patient data. This trained classifier can be applied to compute the disease risk for new coming patients. To avoid disclosure of patients' data, by using advanced encryption standard.

***Keywords*-Data Mining, Security, Naive Bayesian, Clinical Decision Support System, Advanced Encryption Standard.**

I. INTRODUCTION

The improvement of Information Technology has produced huge measure of databases in different zones. This valuable information can be controlled utilizing data mining methods. Data mining holds guarantee in numerous regions of health care and medical research, with applications extending from medical diagnosis to quality affirmation. The power of data mining lies in its ability to allow users to consider data from a variety of perspectives in order to discover apparent or hidden patterns. With the advent of computing power and medical technology, large and diverse data sets and elaborate methods for data classification have been developed and studied. As a result, data mining has attracted considerable attention for the past decade, and has found its way into a large number of applications that have included both data mining and clinical decision support systems. Decision support system refers to a computer-based system that aids the process of decision making. Clinical Decision Support System (CDSS), with various data mining techniques being applied to assist physicians in diagnosing patient diseases with similar symptoms, has received a great attention recently. The main purpose of modern CDSS is to assist clinicians at the point of care. Naive Bayesian classifier, one of the popular machine learning tools, has been widely used recently to predict various diseases in CDSS. It has been developed as robust tool for classification and regression in noisy, complex domains. A key problem that arises in any massive collection of data is that of confidentiality. The need for privacy is sometimes due to law (e.g., for medical databases) or can be motivated by business interests. To address the privacy issues lying in the clinical decision support system, we propose a Privacy Preserving Patient-Centric Clinical Decision Support System, called PPCD, which is based on naive Bayesian classification to help physician to predict disease risks of patients in a privacy-preserving way. Encryption has primarily been used to prevent the disclosure of confidential information, but can also be used to provide authenticity of the source of the message. So, we propose a Clinical Decision Support System on Naive Bayesian in a privacy preserving way using AES scheme. The reminder of this paper is divided as follows: In section II, Proposed System. In section III, Related work. In section IV, Experiment. In section V, conclusion.

II. PROPOSED SYSTEM

The system model mainly focuses on how to securely train Naive Bayesian classifier and use the classifier to clinically decide patients' disease without leaking their private information. Specifically, we define the system model by dividing CDSS into four parties: Database (DB), Data Provider (DP), Processing Unit (PU), and Undiagnosed Patient (PA).

1) *Database* (DB): It stores and manages all the data in the system

2) *Data Provider* (DP representing as administrator): DP can provide historical medical data that contain patients' symptoms and confirmed diseases, which are used for training Naive Bayesian classifier. All these data are stored in the database.

3) *Processing Unit* (PU): PU can be a hospital which can provide online direct-to-consumer service and offer individual risk prediction for various diseases based on client's symptoms. PU uses historical medical data to construct Naive Bayesian classifier and then use the model to predict the disease risk of undiagnosed patients.

4) *Undiagnosed Patient* (PA): PA has some symptom information which is collected during doctor visits or directly provided by patient (e.g., blood pressure, heart rate, weight, etc.). The symptoms can be sent to PU for diseases diagnosis.

To develop the software, information of the previous patient should be given as the dataset, which is provided by the Data provider. DP will encrypt the data set and stores in the database. When the undiagnosed or new coming patient inputs his/her symptoms, the main process takes place. Previously, the Naive Bayesian classifier will be trained using the training data set given by DP. The PU will perform two stages of decryption and retrieve the data from the database. The decrypted data and the new patient's data are compared, predict the disease risk and generate the result to patient privately. This information can be viewed only by the corresponding doctor of the patient. Even the administrator of the system cannot view the medical details of the patient, he can able to view only the profile details.

Based on the system architecture, the system can be split into three modules:
1. Privacy preservation by encryption teachnique
2. Training SVM classifier and
3. Risk computation and Result generation.

1.Privacy Preserving By Encryption Technique

1.1 Privacy Requirements

Privacy is crucial for the success of patient's diseases diagnosis. In our privacy model, we consider DP is trustable which provides correct historical medical data. The internal party PU is considered as curious-but-honest which is interested in DP's individual historical medical data and PA's medical data, but strictly follows the protocols executed in the system. PA is curious about PU's classifier while CP is curious about all the other parties data in the system. Moreover, an external adversary is interested in all data transmitted in the system by eavesdropping. Therefore, in order to prevent both internal party from information leakage and external adversary from eavesdropping, the following privacy requirements should be satisfied in PPCD.

A. DP's privacy: DP's historical medical data contain confirmed case records of patient's symptoms and confirmed diseases. These individual data contain some sensitive information which are highly related to patient's privacy. It cannot be directly exposed to untrusted parties during the transmission and storage. Otherwise, DP will not provide its own data to the other parties due to the privacy information leakage. Therefore, privacy of DP should be preserved in our system.

B. PU's privacy: PU uses historical medical data to train SVM classifier and gets conditional probabilities about the classifier. These probabilities are considered as an asset of PU which cannot directly be sent to patients or leaked to other parties during the disease diagnosis.

C. PA's privacy: PA contains some symptom data which are sensitive and cannot directly expose to other parities. In addition, the diagnosis results are also highly sensitive information which cannot be leaked to other parties. If needed, PA can let the authorized person (authorized clinician) disclose the diagnosis results for further processing.

1.2 AES Algorithm

AES cipher:

Like DES, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES' block and key size. The block and key can in fact be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES- 256 respectively. As well as these differences AES differs from DES in that it is not a feistel structure. Recall that in a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations. A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key. This description of the AES algorithm therefore describes this particular implementation.

Rijndael was designed to have the following characteristics First resistance against all known

attacks. Second speed and code compactness on a wide range of platforms. Third design Simplicity. The overall structure of AES can be seen in(fig 1) The input is a single 128 bit block both for decryption and encryption and is known as the in matrix. This block is copied into a state array which is modified at each stage of the algorithm and then copied to an output matrix . Both the plaintext and key are depicted as a 128 bit square matrix of bytes. This key is then expanded into an array of key schedule words (the w matrix). It must be noted that the ordering of bytes within the in matrix is by column. The same applies to the **w** matrix.

Inner Workings of a Round

The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of it's counterpart in the encryption algorithm. The four stages are as follows:

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round simply leaves out the Inverse Mix Columns stage. Each of these stages will now be considered in more detail.

A. Substitute Bytes

This stage (known as SubBytes) is simply a table lookup using a 16×16 matrix of byte values called an s-box. This matrix consists of all the possible combinations of an 8 bit sequence (28 = 16 × 16 = 256). However, the s-box is not just a random permutation of these values and there is a well defined method for creating the s-box tables. The designers of Rijndael showed how this was done unlike the s-boxes in DES for which no rationale was given. We will not be too concerned here how the s-boxes are made up and can simply take them as table lookups.

Again the matrix that gets operated upon throughout the encryption is known as state. We will be concerned with how this matrix is effected in each round. For this particular round each byte is mapped into a new byte in the following way: the leftmost nibble of the byte is used to specify a particular row of the s-box and the rightmost nibble specifies a column. For example, the byte {95} (curly brackets represent hex values in FIPS PUB 197) selects row 9 column 5 which turns out to contain the value {2A}. This is then used to update the state matrix.

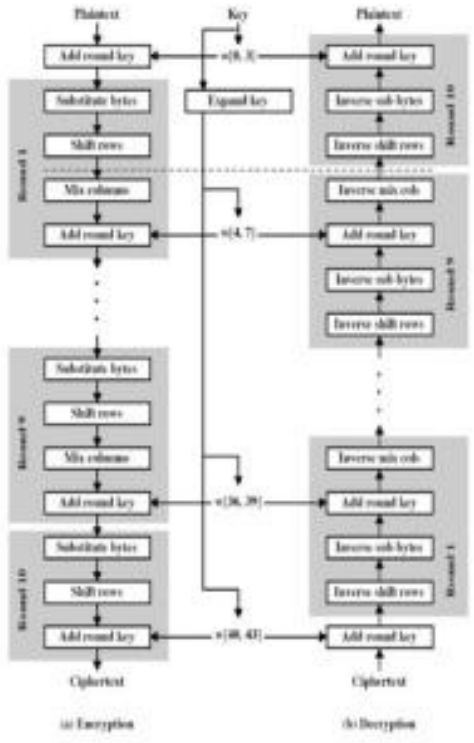

Fig. 1 Overall structure of the AES algorithm

B. Shift Row Transformation

This is a simple permutation an nothing more. It works as follow:

i The first row of state is not altered.
ii. The second row is shifted 1 bytes to the left in a circular manner.
iii. The third row is shifted 2 bytes to the left in a circular manner.
iv. The fourth row is shifted 3 bytes to the left in a circular manner.

The Inverse Shift Rows transformation (known as Inv Shift Rows) performs these circular shifts in the opposite direction for each of the last three rows (the first row was unaltered to begin with). This operation may not appear to do much but if you think about how the bytes are ordered within state then it can be seen to have far more of an impact. Remember that state is treated as an array of four byte columns, i.e. the first column actually represents bytes 1, 2, 3 and 4. A one byte shift is therefore a linear distance of four bytes. The transformation also ensures that the four bytes of one column are spread out to four different columns.

C. *Mix Column Transformation*

This stage (known as Mix Column) is basically a substitution but it makes use of arithmetic of GF(28). Each column is operated on individually. Each byte of a column is mapped into a new value that is a function of all four bytes in the column. The transformation can be determined by the following matrix multiplication on state.

D. Add Round Key Transformation

In this stage (known as AddRoundKey) the 128 bits of state are bitwise XORed with the 128 bits of the round key. The operation is viewed as a column wise operation between the 4 bytes of a state column and one word of the round key. This transformation is as simple as possible which helps in efficiency but it also effects every bit of state.

2.Training Naive Bayesian Classifier

The Naive Bayesian is a widely used tool in classification problems. The Naive Bayesian trains a classifier by solving an optimization problem to decide which instances of the training data set are support vectors, which are the necessarily informative instances to form the Naive Bayesian classifier. NB are intact tuples taken from the training data set for classification.

3.Risk Computation and Result Generation

The trained classifier will compare the training data set and the patient data to predict the class labels i.e. it will compute the disease risk of the patient and generates the report as shown in the figure 2. The result will be sent to the patient and the doctor.

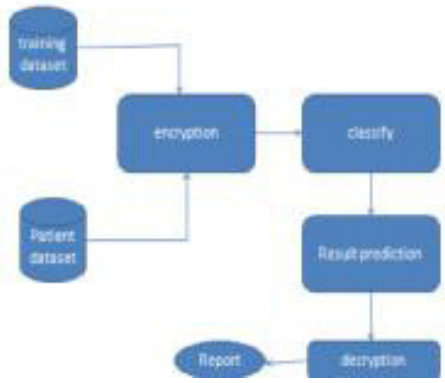

Fig. 2 Risk Computation

III Related Work

The computer-assisted clinical decision support systems was proposed by Ledley and Lusted [1] who found that physicians have an imperfect knowledge of how they solve diagnostic problems. This article dealt with Bayesian and decision-analytic diagnostic systems and experimental prototypes appeared within a few years [2]. Warner et al. [3] developed the first operational Bayesian CDSS for the diagnosis of congenital heart diseases based on history, physical exam, and cardiac catheterization findings. Schurink et al. [4] discussed computer-based decision-support systems to assist Intensive Care Unit (ICU) physicians in the management of infectious diseases. In this paper, they described several computer models (such as bayesian networks) that may be used in clinical practice in the near future.

As the privacy of the patient's information becomes more and more important, na¨ıve

Bayesian classification were considered as a challenge to privacy-preservation due to their natural tendency to use sensitive information about individuals. Privacy-preserving na¨ıve Bayesian classifier was first proposed where data are horizontal partitioned (different patient's tuples with same attribute are partitioned) and stored distributively in different sites. Kantarcıo˘glu et al.

[5] achieved na¨ıve Bayesian classifier by using the secure sum protocol [6] which can support both nominal attributes and numeric attributes. Later, Yi et al. [7] improved the [5] by both efficiency and privacy and this scheme could prevent eavesdropping attack. Different from horizontal partition, another kind of data partition called vertically partition (one patient's different attributes are partitioned) were introduced to privacypreserving na¨ıve Bayesian classifier by using secure scalar product protocol [8] [9]. Vaidya et al. [10] gave us a comprehensive study on both vertically as well as horizontally partitioned data. The data in the existing privacy-preserving na¨ıve Bayesian classifier scheme were distributively stored in different parties as a part of the whole data space.

IV Experiment

To validate the efficiency of the proposed system, a custom simulator is built in Java. It demonstrates that our can efficiently help patient to diagnose the disease with high predict success rate and it also minimizes privacy disclosure. A real dataset is used from the UCI machine learning repository [11] called Acute Inflammations. We use this dataset to test the performance of the Naive Bayesian classifier by using our Clinical Decision Support System.

Real Dataset (Acute Inflammations Dataset): The acute inflammations dataset (AID) was created by a medical expert as a dataset to test the expert system, which was used to perform the presumptive diagnosis of two diseases of the urinary system. This dataset contains 120 instances. Each instance contains 6 attributes [Temperature; Occurrence of nausea; Lumbar pain; Urine pushing; Micturition pains; Burning of urethra, itch, swelling of urethra outlet] and

two decisions [Inflammation of urinary bladder (IUB); Nephritis of renal pelvis origin (NRPO)]. All the attribute and decisions can be expressed as binary bit 1 (YES) or 0 (No) except for temperature. The value of temperature is varied from 35:5_C to 41:5_C in the dataset. Before we use this dataset to classify, all the records should be normalized to 56 attributes (including 51 attributes represent for temperature) and 2 decisions. We first use PPCD to train naive Bayesian classifier and then use this classifier and AID to test the success rate of the classifier. However, there exists false positive diagnosis in AID. The false positive rate of IUB and NRPO diagnosis are 26:23% and 14:29% respectively. We also test the running efficiency about PPCD. It takes 8.848s for PU to train the classifier (including 1.368s for DP to encrypt all the symptoms and diseases offline)

Attribute	Domain
Temperature	(35:5C to 41:5C)
Occurrence of nausea	(0,1)
Lumbar pain	(0,1)
Urine pushing	(0,1)
Micturition pains	(0,1)
Itch, swelling of urethra outlet	(0,1)

Table. 1Accute Data Set

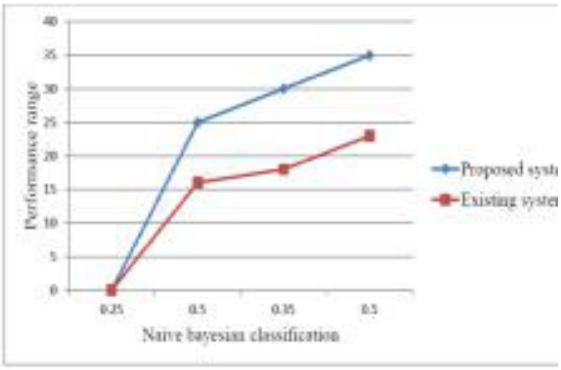

Fig. 3: Comparison Graph

V Conclusion

Privacy preserving Clinical Decision Support System using Naive Bayesian classification is proposed. The system helps the clinician to diagnose the disease risk based upon the symptoms provided by the patient. CDSS on Naive Bayesian increases the accuracy of the diagnosis and reduces the diagnosis time. The system overcomes the information security and privacy challenges through Advanced encrytions standard.

References

[1] R. S. Ledley and L. B. Lusted, "Reasoning foundations of medical diagnosis," Science, vol. 130, no. 3366, pp. 9–21, 1959.

[2] M. A. Musen, B. Middleton, and R. A. Greenes, "Clinical decisionsupport systems," in Biomedical informatics. Springer, 214, pp. 643– 674

[3] H. R. Warner, A. F. Toronto, L. G. Veasey, and R. Stephenson, "A mathematical approach to medical diagnosis: application to congenital heart disease," Jama, vol. 177, no. 3, pp. 177–183, 1961.

[4] C. Schurink, P. Lucas, I. Hoepelman, and M. Bonten, "Computerassisted decision support for the diagnosis and treatment of infectious diseases in intensive care units," The Lancet infectious diseases, vol. 5, no. 5, pp. 305–312, 2005.

[5] M. Kantarcıoglu, J. Vaidya, and C. Clifton, "Privacy preserving naive bayes classifier for horizontally partitioned data," in IEEE ICDM workshop on privacy preserving data mining, 2003, pp. 3–9.

[6] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," ACM SIGKDD Explorations Newsletter, vol. 4, no. 2, pp. 28–34, 2002.

[7] X. Yi and Y. Zhang, "Privacy-preserving naive bayes classification on distributed data via semi-trusted mixers," Information Systems, vol. 34, no. 3, pp. 371–380, 2009.

[8] A. Amirbekyan and V. Estivill-Castro, "A new efficient privacypreserving scalar product protocol," in Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70. Australian Computer Society, Inc., 2007, pp. 209–214.

[9] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," IEEE Network, vol. 28, no. 4, pp. 46–50, 2014.

[10] J. Vaidya, M. Kantarcioglu, and C. Clifton, "Privacy-preserving na¨ıve bayes classification," VLDB J., vol. 17, no. 4, pp. 879–898, 2008.

[11] "Accute data set, UCI machine learning repository"