Fuzzy Clasification Of Time Series Data

A Dissertation submitted to the
Rajiv Gandhi University of Knowledge Technologies
in partial fulfillment of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

G Anil Kumar (R101076)
N Salauddin (R101070)
A Surendra Kumar (R101100)

Rajiv Gandhi University of Knowledge Technologies
R.K. Valley-516330, Vempalli, Kadapa
Andhra Pradesh, India

CERTIFICATE

This is to certify that the dissertation entitled **” Fuzzy Classification of Time Series Data”** submitted by **G Anil Kumar (R101076) , N Salauddin (R101070) , A Surendra Kumar (R101100)**, in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under my supervision and guidance. The dissertation has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Md. Abdul Aziz
Project Guide &
Head of the Department
RGUKT, RK-Valley

DECLARATION

We G Anil Kumar , N Salauddin , A Surendra Kumar hereby declare that this dissertation entitled **"Fuzzy Classification of Time Series Data"** submitted by us under the guidance and supervision of **Md Abdul Aziz** is a bonafide work. We also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date:
Place:

G Anil Kumar(R101076),
N Salauddin(R101070),
A Surendra Kumar(R101100)

Acknowledgment

We would like to express our sincere gratitude to **Md Abdul Aziz**, our project supervisors, for valuable suggestions and keen interest through out the progress of our course of research.

We are highly obligated to each and every one who helped us in successfully completing our Final Year Project. Our thanks and appreciations to the people who have willingly helped out with their abilities. At the outset, we would like to thank The Rajiv Gandhi University of Knowledge Technologies for providing all the necessary resources for the successful completion of our course work..

With Sincere Regards,
G Anil Kumar ,
N Salauddin ,
A Surendra Kumar.

Abstract

Fuzzy Classification is a problem of classification of time series data is an interesting problem in the field of data mining. Even though several algorithms have been proposed for the problem of time series classification we have developed an innovative algorithm which is computationally fast and accurate. In our method we are calculating the fuzzy membership of each test pattern to be classified to each class. We have experimented with 4 benchmark datasets.

We have also implemented One nearest neighbour Euclidean(1NN-ED) classifier and one nearest neighbourMaximum-norm(1NN-L ∞) classifier and compared results with Fuzzy Classification. In the process of classification every individual organization which gives input to the classification, tends to preserve their privacy. Privacy preserving data mining became trending topic in current generation. So security is also necessary to preserve the privacy of sensitive data of an organization. We makes data sets as vertical partition, and we applied secure sum by using elgamal encryption method for providing privacy to the users data.

Table of Content

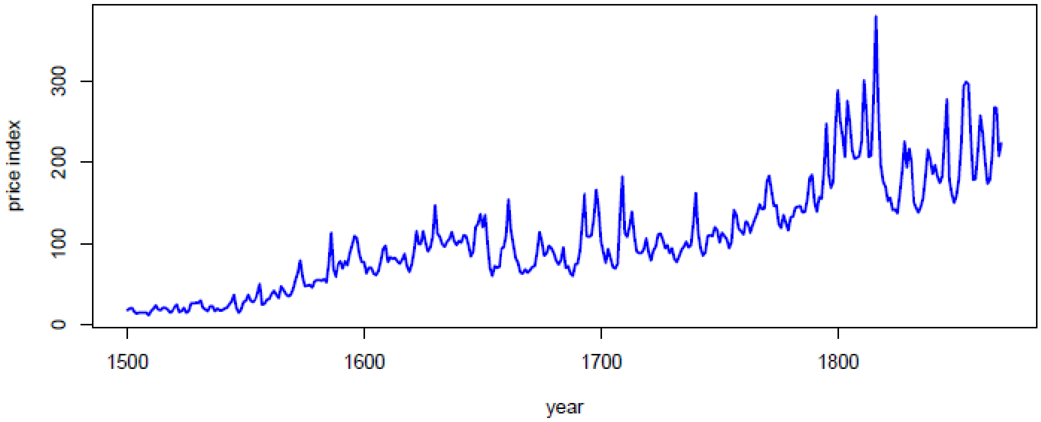
1.Introduction

1.1 Time Series:

A time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals.

Example:

Average wheat price (1500-1869)



1.2 Data Mining:

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Data mining is an interdisciplinary subfield of computer science It is the computational process of discovering patterns in large data sets ("big data") involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

Need of Data mining:

Here are the reasons listed below:

- In field of Information technology we have huge amount of data available that need to be turned into useful information.
- This information further can be used for various applications such as market analysis, fraud detection, customer retention, production control, science exploration etc.

Knowledge Discovery:

Some people treat data mining same as Knowledge discovery while some people view data mining essential step in process of knowledge discovery. Here is the list of steps involved in knowledge discovery process:

- **Data Cleaning:**

  In this step the noise and inconsistent data is removed.
- **Data Integration:**

  In this step multiple data sources are combined.
- **Data Selection:**

  In this step relevant to the analysis task are retrieved from the database.
- **Data Transformation:**

  In this step data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- **Data Mining:**

  In this step intelligent methods are applied in order to extract data patterns.
- **Pattern Evaluation:**

  In this step, data patterns are evaluated.
- **Knowledge Presentation:**

  In this step, knowledge is represented.

1.3 Classification: :

Classification is the process of assignment of class label to a test pattern based on training data with known class labels.

Where we use classification?

- A bank loan officer wants to analyse the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyse a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

Classifier:
Classifier is an algorithm used to assign a class label to every test tuple based on training data by assigning a weight based on different methods such as Euclidean distance and Maximum-Norm distance.

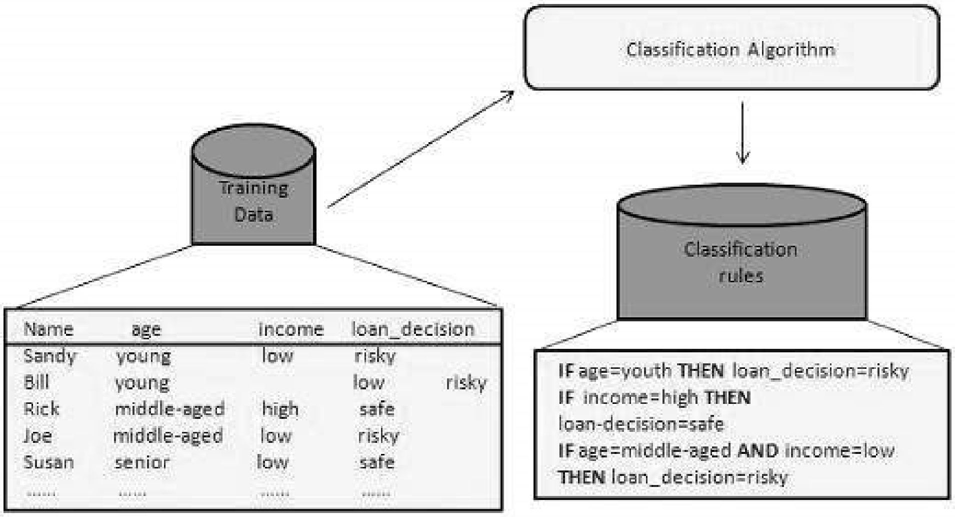
How Does Classification Works:
Let us understand the working of classification. The Data Classification process includes two steps:

- Building the Classifier or Model (Training Phase)
- Using Classifier for Classification(Classification Phase)

Building the Classifier or Model

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.

Learning Phase:



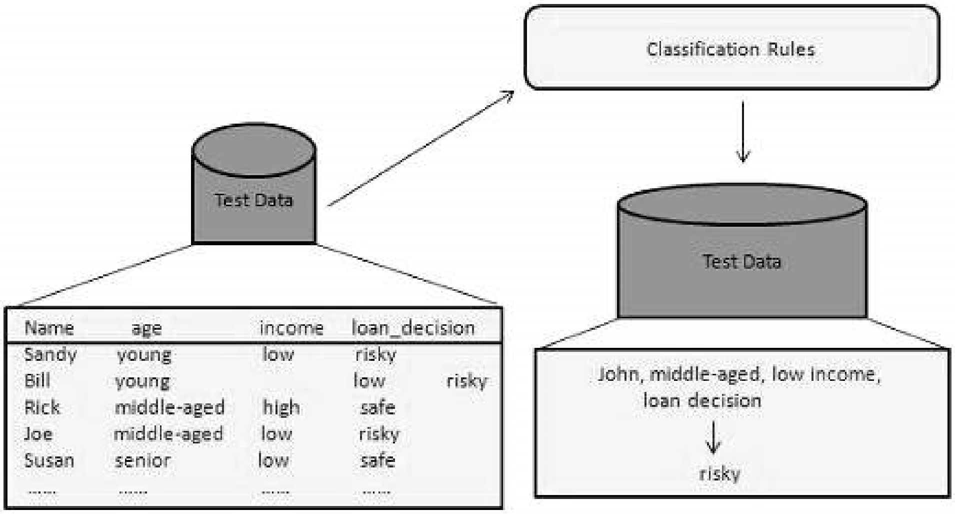
Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

IF age=youth **THEN** loan_decision=risky
IF income=high **THEN** loan-decision=safe
IF age=middle-aged **AND** income=low **THEN** loan_decision=risky

Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.

Learning Phase:



Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

John, middle-aged, low income, loan decision
↓
risky

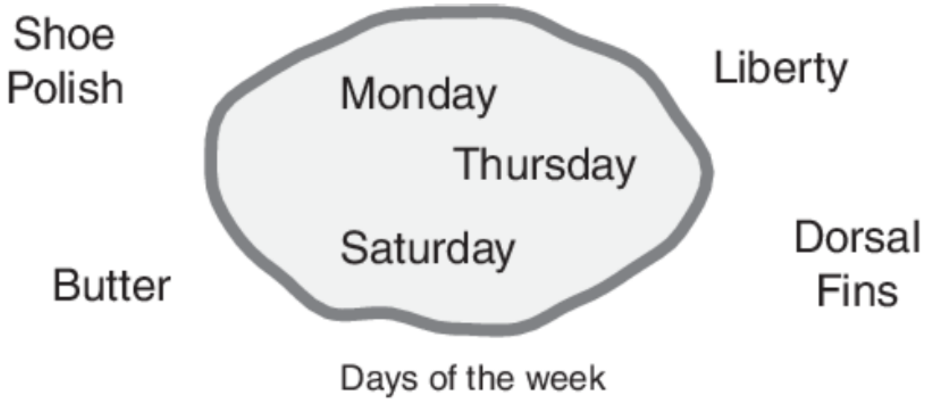
1.4 Fuzzy Classification:

Fuzzy logic is an extension of Boolean logic based on the mathematical theory of fuzzy set. Truth values between True and False. Not everything is either/or, true/false, black/white, on/off etc. For example to say that the water is hot, you need to define the range that the water's temperature can be expected to vary as well as what we mean by the word hot. In fuzzy logic, the truth of any statement becomes a matter of degree.
Fuzzy classification is the process of grouping elements into a fuzzy set whose membership function is defined by the truth value of a fuzzy propositional function.

Fuzzy Set:
Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a clearly defined boundary. It can contain elements with only a partial degree of membership.

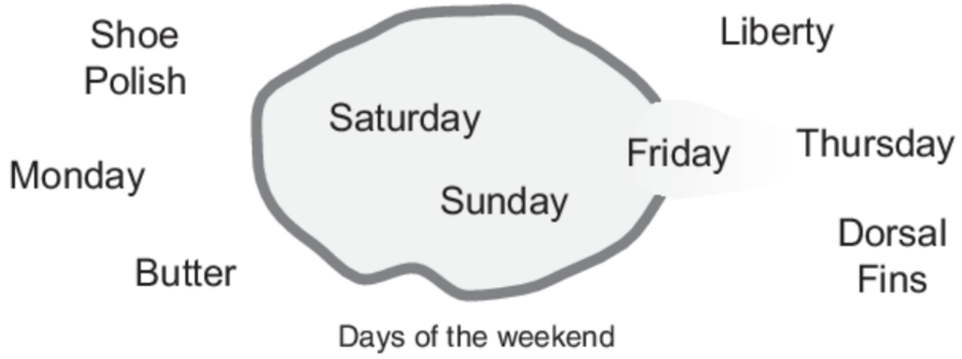
To understand what a fuzzy set is, first consider the definition of a classical set. A classical set is a container that wholly includes or wholly excludes any given element. For example, the set of days of the week unquestionably includes Monday, Thursday, and Saturday. It just as unquestionably excludes butter, liberty, and dorsal fins, and so on.

Example for Classical set



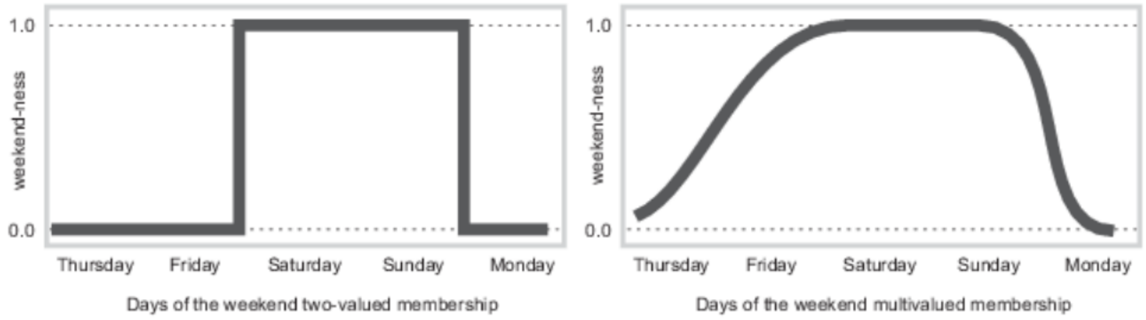
For example consider the set of weekends in a week. In classical set we consider only Saturday and Sunday as weekends. In Fuzzy set Saturday and Sunday and Friday feels like a part of the weekend, but somehow it seems like it should be technically excluded.

Example of Fuzzy Set:



Membership Function:

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept.
This figure shows the member ship function for above example.



1.5 KNN Classifier:

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes.

Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k nearest neighbors of the unknown tuple. (Where k=1,2,3,. . . . ,n)

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

Euclidean Distance:

Closeness is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples. Let us consider an example: X1 = (x11, x12,. . ., x1n) and X2 = (x21, x22, ... , x2n), is $\sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$

1.6 Maximum norm Classifier:

The maximum norm classifier is the maximum of the distance along individual coordinate axes

$L_\infty(x, y) = max_{i=1,2,...,d}|X_i - Y_i|$

1.7 Privacy Preserving Data Mining:

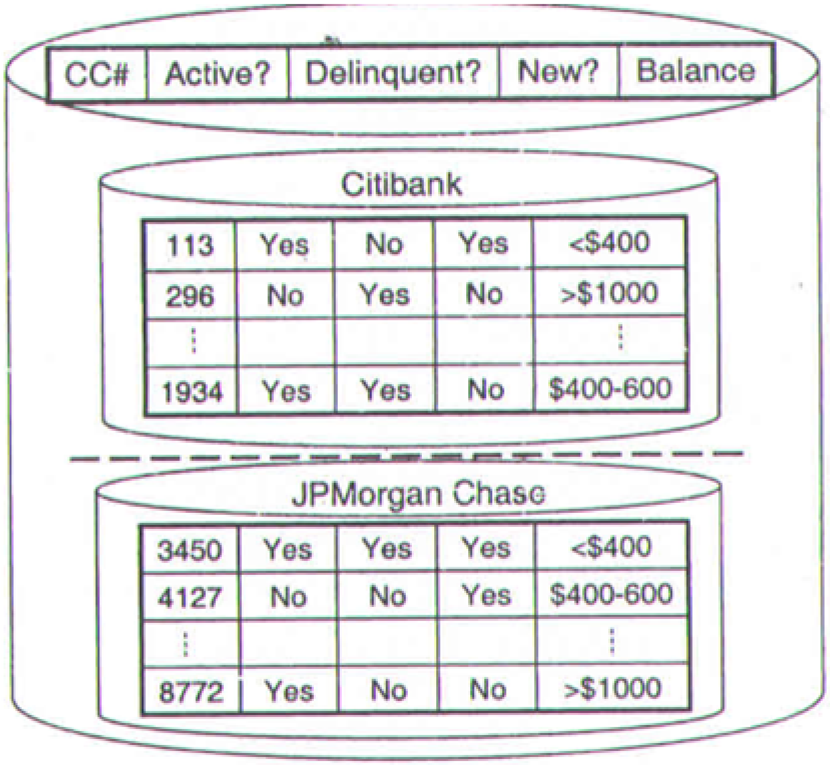
We consider a scenario where two parties having private databases wish to cooperate by computing a data mining algorithm on the union of their databases. Since the databases are confidential, neither party is willing to expose any of the contents to the other. So to make their data as confidential we have to provide privacy to their data by using methods such as secure sum.

Data Partition Models :

It is necessary to first model the different ways in which data is distributed in the real world. There are two basic data partitioning/data distribution models. Horizontal partitioning (homogeneous distribution)and vertical partitioning(heterogeneous distribution) . We now formally define these models. We define a dataset D in terms of the entities for whom the data is collected and the information that is collected for each entity. Thus, D=(E,I) where E is the entity set for whom information is collected sites p1..pk collecting datasets D1=(E1,I1).Dk=(Ek,Ik) respectively.

Horizontal Partitioning

Horizontal partitioning of data assumes that different sites collect the same sort of information about different entities. Therefore, in horizontal partitioning $E_g = E_1 \cup E_2 \cup ... \cup E_k$ and $I_g = I_1 \cap I_2 \cap ... \cap I_k$ many such situations exist in real life. For example all banks collect very similar information however the customer base for each bank tends to be quite different figure 1 demonstrates the horizontal partitioning of data. The figure shows two banks, Citibank, jpmorgan chase, each of which collects credit card information for their respective customers. Attributes such as the account balance, whether the account is new, active, delinquent are collected by both merging the two databases together should lead to more accurate predictive models used for activities like fraud detection.



CC#	Active?	Delinquent?	New?	Balance

Citibank

113	Yes	No	Yes	<$400
296	No	Yes	No	>$1000
⋮				⋮
1934	Yes	Yes	No	$400-600

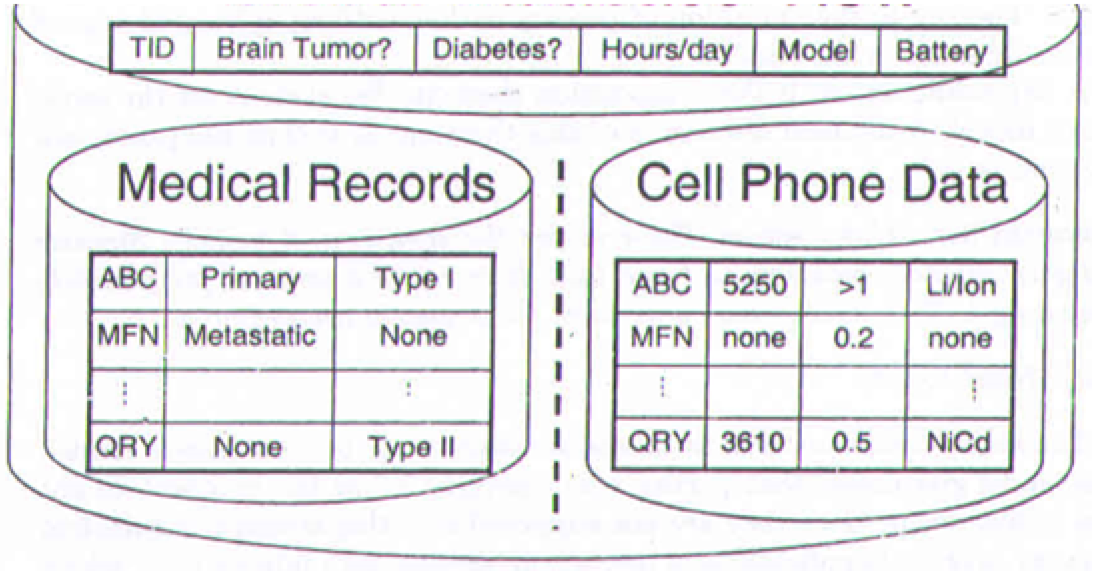
JPMorgan Chase

3450	Yes	Yes	Yes	<$400
4127	No	No	Yes	$400-600
⋮				⋮
8772	Yes	No	No	>$1000

Horizontal partitioning / Homogeneous distribution of data

Vertical Partitioning:

On the other hand vertical partitioning of data assumes that different sites collects different feature sets for the same set of entities. Thus , in vertical partitioning $E_g = E_1 \cap E_2 \cap ... \cap E_k$ and $I_g = I_1 \cup I_2 \cup ... \cup I_k$ for example ford collects information about vehicles manufactured. Fire stone collects information about tires manufactured. Vehicles can be linked to tires. This linking information can be used to join the databases. The global database could then be mined to reveal useful information figure demonstrates vertical partitioning of data.



TID	Brain Tumor?	Diabetes?	Hours/day	Model	Battery

ABC	Primary	Type I
MFN	Metastatic	None
⋮		⋮
QRY	None	Type II

ABC	5250	>1	Li/Ion
MFN	none	0.2	none
⋮			⋮
QRY	3610	0.5	NiCd

Vertical partitioning / Heterogeneous distribution of data

Secure Multi Party Computation:

The basic idea of Secure Multiparty Computation is that a computation is secure if at the end of the computation, no party knows anything except its own input and the results. One way to view this is to imagine a trusted third party everyone gives their input to the trusted party, who performs the computation and sends the results to the participants. Now imagine that we can achieve the same result without having a trusted party. Obviously, some communication between the parties is required for any interesting computation.

Secure Sum:

The secure sum problem rather simple but extremely useful. Distributed data mining algorithms frequently calculates the sum of values from individual sites from thus use it as an underlying primitive.

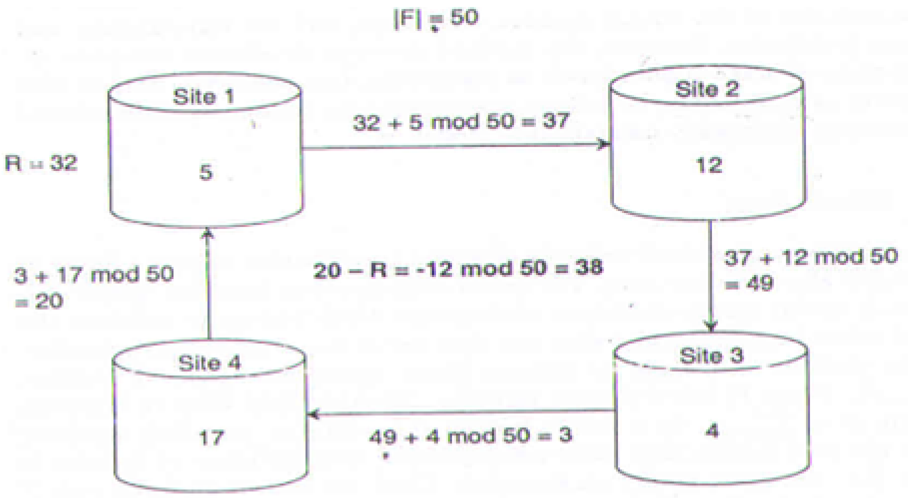
The problem is defined as follows. We assume k parties, p1.pk. party pi has private value xi. Together they want to compute the sum S= $\sum_{i=1}^{k} x_i$ in a secure fashion i.e without revealing anything except the final result. One other sum is known (i.e upper bound on the sum) and we assume that sum is S is a number in the field F at least 3 parties. The following protocol computes such a sum.

Algorithm for Secure Sum:

- P1 generates a random number r from a uniform random distribution over the field F.
- P1 computes the $S_1 = x + rmod \mid F \mid$ and sends it to P2
- For parties p2 . . . P_{k-1}

  Pi receives $S_{i-1} = r + \sum_{j=1}^{i-1} x_i mod \mid F \mid$

  Pi computes mod $\mid F \mid$ and sends it to site P_{i+1}.
- P_k computes $S = S_k - rmod \mid F \mid = \sum_{j=1}^{k} x_i mod \mid F \mid$ and sends it to all other parties as well.



Secure computation of a sum.

Elgamal CryptoSystem:

The Elgamal Algorithm provides an alternative to the RSA for public key encryption. The encryption algorithm is similar in nature to the Diffie-Hellman

key agreement protocol.
Alice chooses,

1. A large prime p_A
2. A primitive element $\alpha_A modulo p_A$,
3. A (possibly random) integer d_A with $2 \leq d_A \leq p_A - 2$
4. $\beta_A \equiv \alpha_A^{d_A} (mod p_A)$

Alice's public key is (p_A, α_A, β_A).Her private key is d_A
Bob sends his encrypted message(r,t) to Alice.
When Alice receives the encrypted message (r, t), she decrypts (using her private key d_A) by computing tr^{d_A}.
Note

$$\begin{aligned} tr^{-d_A} &\equiv \beta_A^k M(\alpha_A^k)^{-d_A} (mod p_A) && (1)\\ &\equiv (\alpha_A^{d_A})^k M(\alpha_A^k)^{-d_A} (mod p_A) && (2)\\ &\equiv M(mod p_A) && (3)\\ & && (4) \end{aligned}$$

Even if Eve intercepts the cipher text (r, t), she cannot perform the calculation above because she doesnt know d_A.

$\beta_A \equiv \alpha_A d_A (mod p_A)$, so $d_A \equiv L_{\alpha_A}(\beta_A)$

Eve can find d_A if she can compute a discrete log in the large prime modulus p_A, presumably a computation that is too difficult to be practical.

2. Fuzzy Classification of time series data algorithm

2.1 Introduction:

In this algorithm we have taken input from UCR Time series classification archive. In input, we trained our algorithm with training data set by calculating maximum, minimum, mean values. And after that, we performed classification algorithm on the given test data set.

Variables in algorithm:

D_tr	Training Data set
D_test	Test Data Set
pc	Predicted class
M	No.of.classes
n	Time series length
n_1	No.of test patterns
N	No.of training patterns
T_s	Time series
CA	Classification Accuracy
ms	Time in milli seconds
X_{ij}	i^{th} example j^{th} dimension value of training data
Y_{ij}	i^{th} example j^{th} dimension value of test data
μ_{ji}	Membership of i^{th} dimension of j^{th} class
mem_j	Membership of the test pattern to the j_{th} class
T_{si}	I^{th} dimension value of the test pattern T_S

2.2 Training Phase

1. For each class k we create three arrays each of size 1 n which are named as
   - $min_{ki} \leftarrow$ which holds the min value of training data of class k of dimension i
   - $max_k \leftarrow$ which holds the max value of training data of class k of dimension i

- $mean_{ki} \leftarrow$ which holds mean value of training data of class k of dimension i

2. For every dimension of time series we calculate the min, max and mean of each class from training data which is used to perform the classification on test data.

Pseudo code for Training Phase

Input:D_{tr}
Output:min,max,mean of every class of every dimension

```
for i← 1: m do
    for j←1: n do
        $min_{ij} \leftarrow \infty$
        $max_{ij} \leftarrow \infty$
        $total_{ij} \leftarrow 0$
    end for
end for
for i← 1: N do
    for j← 1: n do
        Let k ← class of pattern $X_i$
        if $X_{ij} < minkj$ then
            $min_{kj} \leftarrow X_{ij}$
        end if
        if $X_{ij} < max_{kj}$ then
            $max_{kj} \leftarrow X_{ij}$
        end if
        $total_{kj} = total_{kj} + X_{ij}$
    end for
end for
for k ← 1: m do
    for j ← 1: n do
        $mean_{kj} \leftarrow total_{kj}$ /No of patterns of class k
    end for
end for
```

2.3 Classification Phase:

In the classification phase of the classifier, we calculate the score for every class of the test data. We try to find the class to which the test pattern has the highest membership value. The test pattern is classified as belonging to that class.

1. This subroutine will take test data as well as the above calculated min, max and mean of the each class as input.

2. For every test pattern,
   For every dimension i we find the fuzzy membership function of the test pattern to every class
   For a class j, if T_{Si} is less than $mean_{ji}$ then

   $$\mu_{ji} = 1 - \frac{Mean_{ji} - T_{S_i}}{Mean_{ji} - Min_{ji}}$$

   else

   $$\mu_{ji} = 1 - \frac{T_{S_i} - Mean_{ji}}{Min_{ji} - Mean_{ji}}$$

3. For each class j we add up the $_{ji}$ values to get a $score_j$

4. Find the membership function of the test pattern for every class j
   $$Mem_j \leftarrow \frac{score_j}{\sum_{i=1}^{n} Score_i}$$

5. We classify the test pattern as belonging to the class j with maximum mem_j value.

6. We repeat the step 2, 3 and 4 for every remaining test examples

Pseudo code for Classification phase:
Input:D_{test}
Output:No.of correctly predicted examples

```
for i ← 1: $n_1$ do
    for j ← 1: n do
        for each class k ← 1: m do
```

```
            if $Y_{ij} < mean_{kj}$ then
                $\mu_{kj} = 1 - fracMean_{kj} - Y_{ij}Mean_{kj} - Min_{kj}$
            else
                $\mu_{kj} = 1 - \frac{Y_{ij} - mean_{kj}}{Min_{kj} - mean_{kj}}$
            end if
        end for
    end for
    for k←1: m do
        for i←1: n do
            $score_k \leftarrow score_k + \mu_{ki}$
        end for
        $mem_k \leftarrow \frac{score_k}{\sum_{j=1}^{m} Score_j}$
    end for
    $pc \leftarrow k \mid max(mem_k), k : 1, 2, 3, ..., m.$
    if predicted class (pc) is same as actual class then
        correct ←correct + 1
    end if
end for
return correct
```

3.Privacy Preserving Data Mining

3.1 Introduction:

Let us consider two users and they have vertical partition data, they dont want to share their data with each other. But they want to classify the vertical partition data.

Both users compute their train data (Min, max, mean) and test data (Score values) separately. After they are using secure sum with elgamal encryption for adding their score values without knowing each other. Based on score values we are computing Membership values after that we are predicting the class of their data.

3.2 Algorithm: Securely computing the sum of two Integers:

Step 1: Party A generates a random number R and computes $M + i - R$ for each i, such that $-n < i <= n$. Note $m_i = M + i - R$. A encrypts each mi using ElGamal scheme and gets $E(m_i, r_i)$, where each ri is a new random number. Then party A sends each $E(m_i, r_i)$ to party B in the increasing order of i.

Step 2: Party B picks $E(m_N, r_N)$. She re randomizes it and sends $E(m_N, r^{‘})$ back to A, where $r‘ = rN + s$, and s isonly known to party B.

Step 3: Party A partially decrypts$E(m_N; r‘)$ and sends the partially decrypted message to B.

Step 4: Party B finally decrypts the message (by doing partial decryption on the already partially decrypted message) to get $m_N = M + N - R$. Note R is only known to A and m_N is only known to B. Furthermore, $m_N + R = M + N$.

Example: Alice chooses $p_A = 107, \alpha_A = 2, d_A = 67$ and she computes $\beta_A = 267 \equiv 94(mod107)$. Her public key is $(p_A, \beta_A, \alpha_A) = (2, 67, 94)$, and her private key is $d_A = 67$.

Bob wants to send the message ”B” (66 in ASCII) to Alice.

He chooses a random integer k= 45 and encrypts M = 66 as given below
$(r, t) = (\alpha_A^k, \beta_A^k M) \equiv (2^{45}, 94^{45} 66) \equiv (28, 9)(mod107)$.

He sends the encrypted message (28, 9) to Alice.

Alice receives the message (r, t) = (28, 9), and using her private key $d_A = 67$ she decrypts to $tr^{-d_A} = 9.28^{-67} \equiv 9.28^{106-67} \equiv 9.43 \equiv 66 (mod 107)$.

4. Results

4.1. Fuzzy Classification of Time series data

Input-1: Training Data set
For Fuzzy classification algorithm we are given a Training data set as input.

Class Label	T-1	T-2	T-3	T-4	T-5
2	0.99364	2.2738	2.9139	0.35356	0.14021
1	-0.06078	-0.34439	1.9246	0.22284	0.22284
2	0.61165	1.3174	2.0231	0.14115	0.14115
2	0.6445	1.2038	1.9495	0.085223	0.085223
1	0.8053	1.4248	1.2183	0.39233	0.59882
2	1.2042	2.5392	2.5392	-0.13077	0.44137
2	1.1821	2.0396	2.0396	-0.10412	0.32462
2	1.3424	2.0666	2.0666	-0.10604	0.075007

Phase-1: Training Phase

By using training dataset, we calculated the Maximum, Minimum, Mean values for each class label.

Class 1

	T-1	**T-2**	**T-3**	**T-4**	**T-5**
Minimum	-0.060775	-0.34439	1.2183	0.22284	0.22284
Maximum	0.8053	1.4248	1.9246	0.39233	0.59882
Mean	0.3722625	0.540205	1.57145	0.307585	0.41083

Class 2

	T-1	**T-2**	**T-3**	**T-4**	**T-5**
Minimum	0.61165	1.2038	1.9495	-0.13077	0.075007
Maximum	1.3424	2.5392	2.9139	0.35356	0.44137
Mean	0.99641	1.9067	2.25531	0.039833	0.2012

Input-2: Test Data set

In the Classification Phase, we are given a test data set as an input.

Class Label	T-1	T-2	T-3	T-4	T-5
1	-0.27475	-0.27475	2.2068	0.34565	0.34565
2	1.6528	2.4973	2.4973	0.38606	0.17494
2	0.17451	1.3269	1.7879	0.86596	0.86596
1	0.3664	0.98443	3.4566	0.057389	0.057389
1	0.049486	0.7423	1.2042	-0.18145	0.049486
1	0.80169	0.80169	0.80169	-0.01557	-0.01557
1	-2.1091	-0.98212	1.7227	0.82112	1.0465
2	1-0.25174	0.21199	1.1394	0.67571	0.67571

Phase-2: Classification Phase

From the training data set, we trained our algorithm to predict the class label. Based on training values, our class labels will be predicted for every test data tuple and it would be compared to actual class of the test data tuple.

Actual class 2:
Analyzing test pattern of class 2

	T-1	**T-2**	**T-3**	**T-4**	**T-5**	**Scores**	**Mem**
	0.80169	0.80169	0.80169	-0.01557	-0.01557	**Values**	**Values**
Class-1	0.00833	0.704401	-1.17969	-2.813	-1.2681	-4.5483	0.54006
Class-2	0.49391	-0.57204	-3.75326	0.67526	-0.7173	-3.8735	0.4599

Predicted class = Class of Maximum value of Membership value=1

Actual class, predicted classes are same.
After calculating the all values of test data patterns we get the result as
Total Test patterns= 8
No.of Correctly Predicted Examples: 3
Accuracy: 37.5 8

4.2. Fuzzy Classification of vertical partition data

4.2.1. User A training data set:

Class label	T-1	T-2	T-3
2	0.99364	2.2738	2.9139
1	-0.06078	-0.34439	1.9246
2	0.61165	1.3174	2.0231
2	0.6445	1.2038	1.9495
1	0.8053	1.4248	1.2183
2	1.2042	2.5392	2.5392
2	1.1821	2.0396	2.0396
2	1.3424	2.0666	2.0666

By using User A training dataset, calculated the Maximum, Minimum, Mean values for each class label.

class 1

	T-1	**T-2**	**T-3**
Minimum	-0.061	-0.34	1.218
Maximum	0.805	1.424	1.924
Mean	0.372	0.540	1.571

class 1

	T-1	**T-2**	**T-3**
Minimum	0.611	1.203	1.949
Maximum	1.342	2.539	2.913
Mean	0.996	1.906	2.255

User A Testing Phase:

Class Label	T-1	T-2	T-3
1	-0.27475	-0.27475	2.2068
2	1.6528	2.4973	2.4973
2	0.17451	1.3269	1.7879
1	0.3664	0.98443	3.4566
1	0.049486	0.7423	1.2042
1	0.80169	0.80169	0.80169
1	-2.1091	-0.98212	1.7227
2	-0.25174	0.21199	1.1394

User A Classification:

	Time Series-1	Time Series-2	Time Series-3	Score Values
	0.3664	0.98443	3.4566	
Class-1	0.986	0.497	-4.33	-2.853
Class-2	-0.637	-0.312	-0.825	-1.773

4.2.2. User B training data set:

Class Label	**T-1**	**T-2**
2	0.35356	0.14021
1	0.22284	0.22284
2	0.14115	0.14115
2	0.085223	0.085223
1	0.39233	0.59882
2	-0.13077	0.44137
2	-0.10412	0.32462
2	-0.10604	0.075007

User B Training Phase:

Class-1

	T-1	**T-2**
Minimum	0.223	0.223
Maximum	0.392	0.598
Mean	0.307	0.411

Class-2

	T-1	**T-2**
Minimum	-0.130	0.075
Maximum	0.353	0.441
Mean	0.039	0.201

User B Testing Phase:

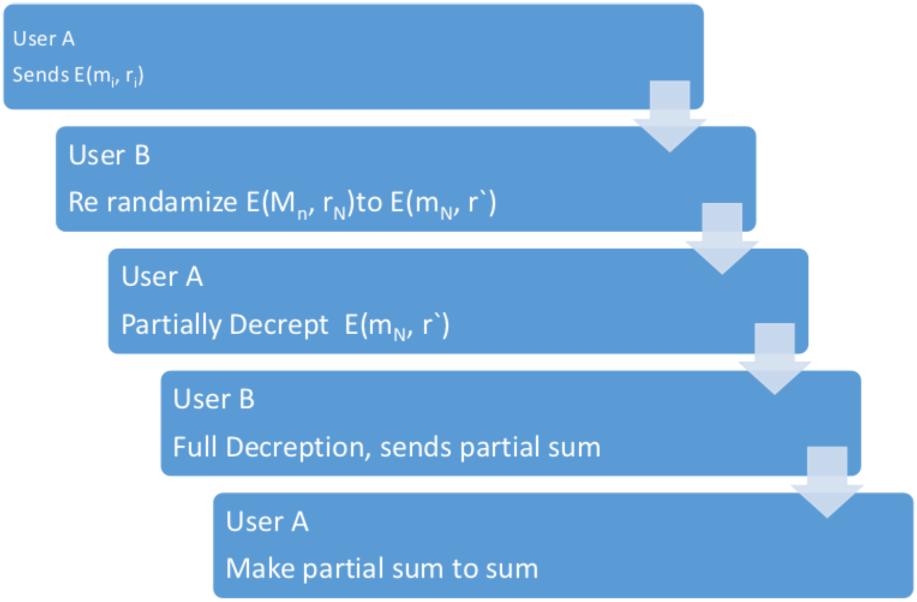
Class Label	**T-1**	**T-2**
1	0.34565	0.34565
2	0.38606	0.17494
2	0.86596	0.86596
1	0.057389	0.057389
1	-0.18145	0.049486
1	-0.01557	-0.01557
1	0.82112	1.0465
2	0.67571	0.67571

User B Classification:

	Time Series-1	**Time Series-2**	**Score Value**
Class-1	-1.952	-0.88	-2.83
Class-2	0.944	-0.139	0.805

User A, user B will have separate score values, based on the score values we are adding those score values to find out membership values by using secure sum for Fuzzy Classification without sharing their data.
Secure Sum:

Score= secure sum of (user A score + user B score)


User A
Sends $E(m_i, r_i)$
User B
Re randamize $E(M_n, r_N)$to $E(m_N, r`)$
User A
Partially Decrept $E(m_N, r`)$
User B
Full Decreption, sends partial sum
User A
Make partial sum to sum

After calculating the both score values by using secure sum, the membership values are:

Class Label	Membership values
1	0.8541
2	0.1458

Predicted class is maximum of membership values, so predicted class is 1. And actual class is also 1.

After calculating all the test data sets, no..of correctly predicted classes are three and the Accuracy is 37.5%.

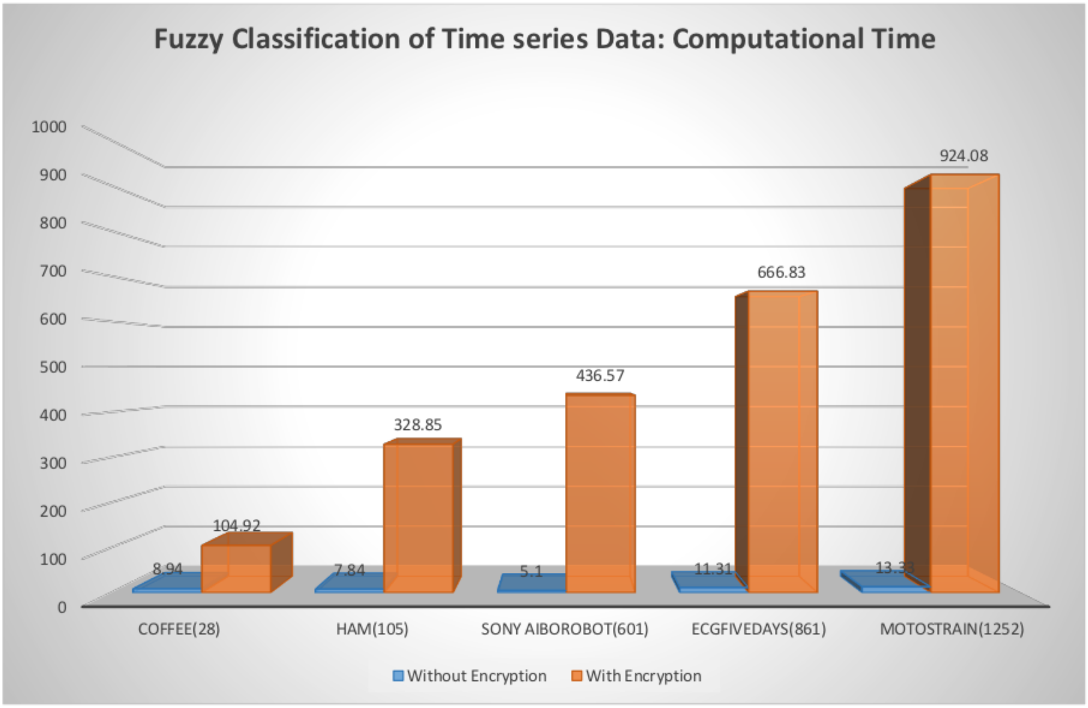
4.3. Comparison of classifications

4.3.1. Comparison of classifications accuracy:

In Fuzzy Classification algorithm, we experienced with 6 benchmark data sets. Our results of accuracy is displayed in given table.

Data Set	**Size**	**Our Approach**	**1NN ED**	**1NN Max Norm**
Sony AIBO Robot	621	98.51	69.55	68.72
Mote Strain	1272	86.10	87.85	73.56
Star Light Curves	9236	82.73	84.88	84.14
ECG Five Days	884	69.91	71.67	71.08
Coffee	28	89.28	93.0	100.0
Ham	105	66.66	60.0	53.33

4.3.2. Comparison of computational time between classifications with and without encryption



5. Conclusion

In this academic year, we have implemented an algorithm for fuzzy classification of time series data. After detailed study on time series data and fuzzy classification, we have implemented them in a programming language in which we are comfortable. And for secure multi-party computation, we provide an encryption scheme called Elgamal encryption combining with secure sum protocol, where it enables the classification of test data without exposing their sensitive data to others after training given to our classification algorithm with training data of all parties involved in classification.

And we tested the user data with our algorithm and represented the results in our report. As far as we concerned, this algorithm gives the best and accurate results which is also efficient in terms of time and space.

And by this report we concluded that, this algorithm is correct and suitable for fuzzy classification of data especially time series data and Elgamal encryption with the combination of secure sum protocol makes the best approach towards secure multi-party fuzzy classification of time series data.

Bibliography

1. IEEE paper by P.Ravi Kumar Department of CSE Fuzzy Classification of Time Series Data in the field of Data Mining in 2010.

2. P.S.P. Cowpertwait and A.V.Metcalfe, Introductory Time Series with R, 1st ed. Springer Publishing Company,Incorporated,2009.

3. X.Xi,E.Keogh, C.Shelton, L.Wei, and C.A.Ratanamahatana, Fast time series classification using numerosity reduction, in Proceedings of the 23rd international conference on Machine learning, ser. ICML 06. New York, NY, USA: ACM,2006, pp. 10331040.

4. E. Keogh and S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ser. KDD 02. New York, NY, USA: ACM, 2002, pp. 102111.

5. Wikipedia

6. UCR Test Archive