Chapter- 4

Extended Frequent Inter-Transaction Association Rule Mining and Its Performance Analysis

Increasing computerization is resulting broad use of computer applications in finance, bio-informatics, medicine and technology. To unhide true knowledge from given data, various techniques are suggested by different researchers. Data Mining is one of the techniques to obtain detail information from given database. For time dependent database, various time series analysis techniques are available in literature.

Literature Survey has been carried out on extended association rules and their usefulness and also techniques and computational complexity aspects of various association rule mining algorithms. There have been algorithms which either generate rules under other requirements or extend the basic definition of what an association rule is. Researcher examines some of them to extend the basic algorithms. Most initial research into association rules has assumed that the data is categorical. The quantitative association rule problem assumes that data may be both categorical and quantitative.
While the amount of data increases gradually, the frequent itemsets of inter-transaction association rule will become larger and larger and hard to handle. At least there is no algorithm available that deals with quantitative inter-transaction association rules. Very less work has been done to discover this kind of rules in financial domain and this area is still emerging. During the last decade, stocks and futures traders have come to rely upon various types of intelligent systems to make trading decisions.

The investigation performed by financial researchers, showed that there is no actual difference between a sequence of random numbers and sequence characterizing changes of real market price of stocks - except that changes of stock market prices unlike random numbers influenced by many factors. Hence we can say that they are not evenly distributed in the increasing and decreasing directions. This fact suggests that they can be predicted.

Computational completeness results open up the possibility that the sufficiently large database of change in price of a real stock could be sufficiently deterministic for various prediction purposes. This is in contrast to the most popular perception that the future price of a stock cannot be predicted from its historical prices because stock prices are like random numbers and contain no information. Various data mining techniques are suggested by various authors. FITI and frequent transaction algorithms are suggested in literature. Hitesh Raval and Dr. Vikram kaushik [35], [60] had introduced a Frequent Inter – Transaction Association Rule Mining.
Till there are some gaps in these algorithms which can be fulfilled. With this assumption, present study is planned to carry forward the work still done by various researchers in data mining. Present study is carried out to find alternative way to get precise results in data mining in effective way (less time and space).

Researcher proposed efficient framework called Extended Frequent Inter-Transaction Association Rule Mining (EFITARM) for mining inter transaction association in the transactional dataset which is contains constant number of items in each transaction. Researcher approach is to predict the movement of stock price with user defined minimum support and confidence threshold value and also predicts the probable variation in stock closing price and traded volume based on historical data of characteristic.

Researcher introduces new logic for data mining and pruning with different conditions in simpler way.

Present study has been planned to get efficient pattern mining from temporal data using correlation of time dependent data set. A typical problem of stock market is taken to understand and solve this task.

In first face opening and closing price with volume of various companies listed in National Stock Exchange of India (NSE) is collected for the period of 1244 Working days i.e. 4th January 2010 to 31st December 2014, to get in-depth outcome, large scale companies of various sectors are selected.

The sector wise list of selected companies is given below.
<table>
<thead>
<tr>
<th>Sr.No</th>
<th>Sector</th>
<th>Company Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Auto</td>
<td>Bajaj Auto</td>
<td>Hero Motocorp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bharat Forg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mahindra &amp; Mahindra</td>
</tr>
<tr>
<td>2</td>
<td>Banking</td>
<td>HDFC</td>
<td>PNB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SBI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ICICI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BOB</td>
</tr>
<tr>
<td>3</td>
<td>IT</td>
<td>APTECH</td>
<td>INFOSYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WIPRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HCL Infosystems</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HCL Technologies</td>
</tr>
<tr>
<td>4</td>
<td>Oil &amp; Gas</td>
<td>EASSR OIL</td>
<td>ONGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GAIL</td>
</tr>
<tr>
<td>5</td>
<td>Parma</td>
<td>CADILA</td>
<td>CIPLA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SUNPHARMA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DR REDDY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLAXO</td>
</tr>
</tbody>
</table>

Table – 4.1: List of selected companies
4.1 Introduction of Problem

Time dependent data are very difficult to mine. Various algorithms are proposed by different researchers. Algorithm and association rule given by Hitesh Raval and Dr. Vikram Kaushik [35], [60]. This study shows association rule using FP Tree and Conditional FP Tree. Main limitation of this study is number of sliding windows and size of these windows. Using this algorithm database size, number of transactions and hence time is higher. To overcome these limitations, present study has been planned.

In present study new algorithm is proposed which is more efficient then other algorithm which is available in literature. Main aim of present study is to produce single mega transaction dataset from two different datasets of “Closing Price” and “Trade Volume” of selected companies listed in National Stock Exchange.

The objectives are to get idea of bulk selling or buying pattern with the help of following:

1) Closing Price of a company of a day from Closing price of another company on same day

2) Closing Price of a company of a day from Closing price and volume of another company on same day

3) Closing Price of a company from of next day from Closing price of same company on previous day

For all these three objectives researcher has made different algorithm and program. The detail of logic of the algorithm is given below with example.
The logic is given in form of flow chart as below:

1. Make mxn matrix
2. Obtain separate matrices for each company containing daily changes in closing price / volume
3. Convert daily increase / decrease data in form of codes and reframe above matrices
4. Decide Minimum Threshold
5. Obtain Support
6. Calculate Confidence
4.2 Notations

Let assume closing price of $i^{th}$ company on $j^{th}$ day is denoted by $C_{ij}$. Similarly volume of $i^{th}$ company on $j^{th}$ day is denoted by $V_{ij}$.

Let $X: [x_{ij}]$ be an $m \times n$ matrix representing closing price of $i^{th}$ company on $j^{th}$ day.

$$
X = \begin{bmatrix}
    1 & X_{11} & X_{12} & \ldots & X_{1j} \\
    2 & X_{21} & X_{22} & \ldots & X_{2j} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    i & X_{i1} & X_{i2} & \ldots & X_{ij}
\end{bmatrix}
$$

After making matrix as per above format, calculate rate of increase or decrease (inter day changes) in share price of company $i$ on day $j$ as compared to day $j-1$. It means rate of increase or decrease of $i^{th}$ company denoted by $D_{ij}$ is obtained by difference of closing price of days $(j-1)$ and $j$ divided by closing price of day $(j-1)$ multiply by 100.

Let denote this new matrix containing inter day changes by $D: [D_{ij}]$. It present inter day changes in price / volume of $i^{th}$ company. Thus $D_{11}$ represents change in price / volume of first company from day 1 to day 2. Let us introduce same in form of equation.

$$
D = [D_{ij}] = \frac{(C_{ij} - C_{ij-1})}{C_{ij-1}} \times 100
$$

$$
D = \begin{bmatrix}
    1 & D_{11} & D_{12} & \ldots & D_{1,j-1} \\
    2 & D_{21} & D_{22} & \ldots & D_{2,j-1} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    i & D_{i1} & D_{i2} & \ldots & D_{i,j-1}
\end{bmatrix}
$$
After obtaining matrix D, the percent increase / decrease is grouped in following manner.

- If closing price or volume increase up to 2.5% code = A
- If closing price or volume increase above 2.5% code = B
- If closing price or volume decrease up to 2.5% code = C
- If closing price or volume decrease above 2.5% code = D

In notation, above codes are given as follows:

A: \( 0 \leq C_p \leq 2.5 \)

B: \( C_p \geq 2.5 \)

C: \( -2.5 \leq C_p \leq 0 \)

D: \( C_p \leq -2.5 \)

Let introduce matrix \( Y \): \([Y_{i(j-1)}]\) containing information of matrix D in form of codes as given above. Thus entries in matrix \( Y \) are only A, B, C & D codes.

\[
Y = \begin{bmatrix}
Y_{11} & Y_{12} & \ldots & Y_{1(j-1)} \\
Y_{21} & Y_{11} & \ldots & Y_{2(j-1)} \\
\vdots & \vdots & \ddots & \vdots \\
Y_{i1} & Y_{i2} & \ldots & Y_{ij-1}
\end{bmatrix}
\]
After obtaining matrix Y, next step is to obtaining support and confidence of the logic.

4.2.1 Support and Confidence

**Definition:** Support of an association rule is defined as the percentage/fraction of records that contain to the total number of records in the database.

\[
\text{Support (XY)} = \frac{\text{Support count of XY}}{\text{Total Number of Transaction in Y}}
\]

For present logic, support of an event is counted as number of times that event occurs. In present example, support is obtained using matrix algebra. Support of each notation is obtained by getting summation of each row of that matrix.

**Definition:** Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain to the total number of records that contain X, where if the percentage exceeds the threshold of confidence an interesting association rule can be generated.

\[
\text{Confidence (X|Y)} = \frac{\text{Support (XY)}}{\text{Support(X)}}
\]

After obtaining support of each notations, check whether they satisfies the minimum threshold decided by researcher or not. Select records of notations (matrix rows) from each company which satisfies minimum threshold.
In present logic, confidence is obtained as number of times co-occurrence of notations (whose supports are satisfying minimum threshold) out of all possible occurrences of notations.

Above logic is explained with small example here for better understanding. For example let assume that $X: [x_{ij}]$ is defined as follows:

$$
\begin{array}{cccc}
\text{Company/Day} & 1 & 2 & 3 & 4 \\
1 & 186.2 & 188.3 & 189.2 & 181.2 \\
2 & 59.3 & 64.3 & 58.9 & 62.6 \\
3 & 190.2 & 188.4 & 180.3 & 168.2 \\
4 & 168.4 & 164.8 & 170.4 & 160.2 \\
\end{array}
$$

$$
\begin{array}{cccc}
\text{Company/Day} & 2 & 3 & 4 \\
1 & 1.128 & 0.478 & -4.228 \\
2 & 8.432 & -8.398 & 6.282 \\
3 & -0.946 & -4.299 & -6.711 \\
4 & -2.138 & 3.398 & -5.986 \\
\end{array}
$$

For easy understand and quick calculation, inter day percentage changes is coded as follows.

A: $0 \leq C_p \leq 2.5$

B: $C_p \geq 2.5$

C: $-2.5 \leq C_p \leq 0$

D: $C_p \leq -2.5$
Define matrix Y: \([Y_{ij}]\) consists notations (codes) as above. It contains only entries of codes “A, B, C and D”. It looks like as below.

\[
Y = \begin{array}{c}
\text{Company/Day} \\
1 & A & A & D \\
2 & B & D & B \\
3 & C & D & D \\
4 & C & B & D \\
\end{array}
\]

After preparing matrix of notations, prepare new “k” number of m x n matrix where, k is number of companies, m is number of notations (in our case m = 4) and n is number of days. In each matrix, for all row give binary code “0” and “1” which represents “Absence” and “Presence” of corresponding notation in that row. E.g. if we consider matrix \(y\) as given above to make two new m x n matrices for company – 1 and company – 2, they look like as follows.

\[
\begin{array}{c}
\text{Notation/Day} \\
A & 1 & 1 & 0 \\
B & 0 & 0 & 0 \\
C & 0 & 0 & 0 \\
D & 0 & 0 & 1 \\
\end{array}
\]

Matrix for Company – 1 =

\[
\begin{array}{c}
\text{Notation/Day} \\
A & 0 & 0 & 0 \\
B & 1 & 0 & 1 \\
C & 0 & 0 & 0 \\
D & 0 & 1 & 0 \\
\end{array}
\]

Matrix for Company – 2 =
4.2.2 Obtaining Support

As discussed above, support of an event is counted as number of times that event occurs. Support of each notation is obtained by getting summation of each row of that matrix.

After obtaining all matrices of each selected companies as above, calculate row sums for all matrices. In first matrix, row totals (supports) will be 2, 0, 0 and 1 for Notations A, B, C and D respectively. In similar way, for second matrix it will be 0, 2, 0 and 1.

As a researcher we have to decide a “cut off” which is called “Minimum Thresholds”. Find out row totals which satisfied “Minimum Thresholds”.

Now, if we decide the minimum threshold by 2 (two) then only first row of first matrix and second row of second matrix will be selected for further analysis.

<table>
<thead>
<tr>
<th>Notation/Day</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Matrix for Company – 1 =

<table>
<thead>
<tr>
<th>Notation/Day</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Matrix for Company – 2 =

After obtaining support for each notations of company – 1 and – 2, confidence is to be obtained to check validity of prediction.
4.2.3 Obtaining Confidence

As per definition, confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain to the total number of records that contain X, where if the percentage exceeds the threshold of confidence an interesting association rule can be generated.

After obtaining support of each notations, check whether they satisfies the minimum threshold decided by researcher or not. Select records of notations (matrix rows) from each company which satisfies minimum threshold.

In present logic, confidence is obtained as number of times co-occurrence of notations (whose supports are satisfying minimum threshold) out of all possible occurrences of notations.

In above example, notation “A” from company – 1 and notation “B” from company – 2 are satisfying minimum threshold of “2”. It means that majority of times, in company-1; closing price is increasing up to 2.5% every day. Similarly, in company-2, majority time’s closing price is increasing between 2.5% to 5%.

Now to obtain confidence, select only those rows from both matrices which satisfies minimum threshold. Sum of each row of first matrix with every row of second matrix is obtained to get confidence.

Matrix for Company – 1 =

<table>
<thead>
<tr>
<th>Notation/Day</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Matrix for Company – 2 =

<table>
<thead>
<tr>
<th>Notation/Day</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
The addition of above two rows gives answer [2 1 1]. Here values 2 occur once out of three entries (days). Hence the confidence of “AB” is 1/3 = 33.33%. It can be concluded that 33.33 times out of 100, price of company – 1 is increasing by 0-2.5% with price of company – 2 by 2.5-5%.

The above logic is given in form of algorithm as follows:

### 4.3 Extended Frequent Inter-Transaction Association Rule Mining Algorithm

**Input:** transaction database D, minimum support minisup, maximum span maxspan.

**Output:** all frequent inter-transaction patterns FP.

**Method:**

1. Scan D to find all frequent patterns and prepare Company list A, B, C
2. Calculate support for company list A,B,C
3. Display support as symbolic representation of grade in up and down strategies for company list A,B,C
4. Calculate confidence by matrix multiplication

   for each x into CompanyListA[grade A,B,C,D]

   for each y into CompanyListB[grade A,B,C,D]

   for each z into CompanyListC[grade A,B,C,D]

   sum $\leftarrow$ CompanyListA[x] + CompanyListB[y] + CompanyListC[z]
prepare confidence 1stAfterMultiply[ ]

end for

end for

end for

5 Possible combination

for i \rightarrow 0 \text{ to } 1stAfterMultiply[ ].count

for j \rightarrow 0 \text{ to } 1stAfterMultiply[ ].count

keep if confidence \geq \text{ threshold value}

end for

end for

for i \rightarrow 0 \text{ to FinalConfidenceList}

Convert confidence in %

end for

6 for each confidence in final confidence list

Display result as possible confidence

end for

7 Output FP

Above algorithm is used for bulk selling or buying (transaction) pattern as follows:
1) Closing Price of a company of a day from Closing price of another company on same day
2) Closing Price of a company of a day from Closing price and volume of another company on same day
3) Closing Price of a company from of next day from Closing price of same company on previous day

Using above algorithm, a program has been made with good user interface. This program has options to select objective to analyse. Present program is made for comparison of three companies at a time. One option for selection of sector (group of companies e.g. Auto, Banking, IT, Oil, Parma) is added to select proper option. On the basis of cut off provided by user, it provides supports of all companies for each factor combinations. At the end it provides final confidence with interpretation and time taken by program to run this output (separately for support and for whole output).

4.4 Programming Algorithm of EFITARM

For the programme development researcher used

IDE: Visual Studio 2012
Database: SQL Server 2008
Programming Language: C#
.Net Framework: 4.5
Application Architecture components: WPF

Start Execution Function Name: Calculation()

Local Variable Name: arraylistCompanyA, arraylistCompanyB, arraylistCompanyC, currentTime

1) //Get Data from Database of CompanyA

DataSet ds =
Common.ReturnDS(GetGradeByCompanyId(Convert.ToInt32(cmbCompany1.SelectedValue)));

//Convert that Data to List based on grade for Create Matrix

for (int i = 1; i < count; i++)
{
    if (Convert.ToString(ds.Tables[0].Rows[i]["Grades"]) == "A")
        lstA.Add(1);
    else
        lstA.Add(0);
    if (Convert.ToString(ds.Tables[0].Rows[i]["Grades"]) == "B")
        lstB.Add(1);
    else
        lstB.Add(0);
    if (Convert.ToString(ds.Tables[0].Rows[i]["Grades"]) == "C")
        lstC.Add(1);
    else
        lstC.Add(0);
if (Convert.ToString(ds.Tables[0].Rows[i]["Grades"]) == "D")
    lstD.Add(1);
else
    lstD.Add(0);
}

2) //Create main List arraylistCompanyA and add data to this list in matrix form for Further Operation and lstSupportCompanyA for Support Display

for (int i = 0; i < arraylistCompanyA.Count; i++)
{
    Combination temp = arraylistCompanyA[i];
    Support objSupport = new Support();
    Foreach (var id in temp.ar)
    {
        if (id > 0)
            objSupport.supp++;
    }
    lstSupportCompanyA.Add(objSupport);
}

if (string.IsNullOrEmpty(txtCutOffValue.Text))
    return null;
if (temp.ar.Sum() < Convert.ToInt32(txtCutOffValue.Text))
{

arraylistCompanyACopy.Remove(temp);
}
}

3) //Display Support

grdSupportA.ItemsSource = null;
grdSupportA.Items.Clear();

lblCompanyA.Content = "Company " + cmbCompany1.Text;

lstSupportCompanyA[0].grade = "0-2.5% ↑";

lstSupportCompanyA[1].grade = "2.51-5% ↑";

lstSupportCompanyA[2].grade = "0-2.5% ↓";

lstSupportCompanyA[3].grade = "2.51%-5% ↓";
grdSupportA.ItemsSource = lstSupportCompanyA;

4) //Repeat Above Step For Company 2 and Company 3 so we get Three list

arraylistCompanyA, arraylistCompanyB, arraylistCompanyC

5) //Create List lstAfterMultiply for arraylistCompanyA, arraylistCompanyB,

arraylistCompanyC for Confidence calculation

// Calculate Confidence by Multiply Matrix

foreach (Combination first in arraylistCompanyA)
{
    foreach (Combination second in arraylistCompanyB)
    {
        foreach (Combination third in arraylistCompanyC)
{  
    int sum = 0;
    for (int j = 0; j < third.ar.Count; j++)
    {
        sum = first.ar[j] + second.ar[j] + third.ar[j];
    }

    Confidence obj1 = new Confidence();
    obj1.val = sum;
    obj1.combination = Convert.ToString(first.PosssibleCombination + second.PosssibleCombination + third.PosssibleCombination);
    lstAfterMutiply[a].Add(obj1);
    a++;
    }

}  

//Possible Combination
List<Confidence> lst3 = new List<Confidence>();  
for (int i = 0; i < lstAfterMutiply.Count; i++)
{
    Confidence objcom = new Confidence();
    objcom.val = 0;

    for (int j = 0; j < lstAfterMutiply[i].Count; j++)
    {  

}
if (lstAfterMutiply[i][j].val == 3)
{
    objcom.val = objcom.val + 1;
    objcom.combination = lstAfterMutiply[i][j].combination;
    lst3.Add(objcom);
}

lst3 = lst3.Distinct().ToList();
for (int i = 0; i < lst3.Count; i++)
{
    Confidence obj = new Confidence();
    obj.val = (lst3[i].val / Convert.ToDouble(Days - 1)) * 100;
    obj.val = Math.Round(obj.val, 2);
    obj.combination = lst3[i].combination;
    confidencelst.Add(obj);
}

6) //Display Confidence

List<Confidence> lsttempc = new List<Confidence>();
foreach (Confidence obj1 in confidencelst)
{

79
obj1.Percentage = obj1.val + "\%";

lsttempc.Add(obj1);

grdConfidence.ItemsSource = lsttempc;

After the running above programme screen shot of out put is given below

4.5 EFITARM Snap Shots of Four Different Objectives.

For the programme development researcher used

IDE: Visual Studio 2012

Database: SQL Server 2008

Programming Language: C#

.Net Framework: 4.5

Application Architecture components: WPF

Computer configuration (Where researcher implement the programme)

Operating System: Windows 8

Processor: Intel(R) Core(TM) i3-3120M CPU @ 2.50 GHz

RAM: 4.00 GB

System type: 64-bit Operating System, x64-based processor
4.5.1 Closing price of one company with two other companies on same day

![Image](image_url)

**Figure 4.1:** Logic - 1 Predicting price of Company - 3 on day 1 using prices of companies - 1 & 2 on same day
4.5.2 Closing price of one company on previous with two other companies on next day

![Figure 4.2: Logic - 2 Predicting price of Company - 3 on day 2 using prices of companies - 1 & 2 on day 1](image)

Revised: If the price of Company Bajaj Auto goes up to 2.5% and Company Hero Moto goes up to 1.5%, TATA Motors goes up to 2.5% (Next Day)

Total number of record: 1244
4.5.3 Closing price and volume of one company on a day with closing price of another company on same day

Figure 4.3: Logic - 3 Predicting price of Company - 2 on day 1 using prices and volume of company - 1 on same day
4.5.4 Closing price and volume of one company on previous day with closing price of another company on next day

Figure 4.4: Logic - 4 Predicting price of Company - 2 on day 2 using prices and volume of company - 1 on day – 1
4.6 FITI Snap Shots of Four Different Objectives.

4.6.1 Closing price of one company with two other companies on same day

Figure 4.5: Logic - 1 Predicting price of Company - 3 on day 1 using prices of companies - 1 & 2 on same day
4.6.2 Closing price of one company on previous with two other companies on next day

![Image of a software interface for predicting stock prices]

Figure 4.6: Logic - 2 Predicting price of Company - 3 on day 2 using prices of companies - 1 & 2 on day – 1
4.6.3 Closing price and volume of one company on a day with closing price of another company on same day

Figure 4.7: Logic - 3 Predicting price of Company - 2 on day 1 using prices and volume of company - 1 on same day
4.6.4 Closing price and volume of one company on previous day with closing price of another company on next day

Figure 4.8: Logic - 4 Predicting price of Company - 2 on day 2 using prices and volume of company - 1 on day – 1
4.7 Utility

4.7.1 Import Data

4.7.1.1 Add Group

Figure 4.9: Add sector

4.7.1.2 Add Price

Figure 4.10: Add price of the company
4.7.1.3 Add Volume

![Figure 4.11: Add volume of the company](image1)

4.7.1.4 Import Data

![Figure 4.12: Import data successfully message](image2)
4.7.2 Export Data

Figure 4.13: Generate report

4.7.2.1 Report Generate

Figure 4.14: Report generate in pdf formate
4.7.3 Print

Figure 4.15: Print report

4.7.4 Truncate Data

Figure 4.16: Delete the data from database
4.8 Performance Analysis

Figure 4.17: Comparison of average time taken by two logics for different threshold for logic – 1

Mean time taken by EFITARM and FITI logics for logic – 1 are given in above figure with different threshold values. For every threshold value, average time taken by EFITARM is significantly low as compared to FITI logic (p-value<0.01).
Figure 4.18: Comparison of average time taken by two logics with different number of days for logic – 1

Mean time taken by EFITARM and FITI logics for logic – 1 are given in above figure with different number of days. Up to 200 days of data, average time taken by both logics are almost equal but for higher number of days (>200), it is significantly low for EFITARM as compared to FITI logic (p-value<0.01). As number of days increase, difference of time taken by two logics is significantly increases.
Mean time taken by EFITARM and FITI logics for logic – 2 are given in above figure with different threshold values. For every threshold value, average time taken by EFITARM is significantly low as compared to FITI logic (p-value<0.01).
Figure 4.20: Comparison of average time taken by two logics with different number of days for logic – 2

Mean time taken by EFITARM and FITI logics for logic – 2 are given in above figure with different number of days. Up to 150 days of data, average time taken by both logics are almost equal but for higher number of days (>150), it is significantly low for EFITARM as compared to FITI logic (p-value<0.01). As number of days increase, difference of time taken by two logics is significantly increases.
Figure 4.21: Comparison of average time taken by two logics for different threshold for logic – 3

Mean time taken by EFITARM and FITI logics for logic – 3 are given in above figure with different threshold values. For every threshold value, average time taken by EFITARM is significantly low as compared to FITI logic (p-value<0.01).
Figure 4.22: Comparison of average time taken by two logics with different number of days for logic – 3

Mean time taken by EFITARM and FITI logics for logic – 3 are given in above figure with different number of days. Up to 150 days of data, average time taken by both logics are almost equal but for higher number of days (>150), it is significantly low for EFITARM as compared to FITI logic (p-value<0.01). As number of days increase, difference of time taken by two logics is significantly increases.
Figure 4.23: Comparison of average time taken by two logics for different threshold for logic – 4

Mean time taken by EFITARM and FITI logics for logic – 4 are given in above figure with different threshold values. For every threshold value, average time taken by EFITARM is significantly low as compared to FITI logic (p-value<0.01).
Figure 4.24: Comparison of average time taken by two logics with different number of days for logic – 4

Mean time taken by EFITARM and FITI logics for logic – 4 are given in above figure with different number of days. Up to 150 days of data, average time taken by both logics are almost equal but for higher number of days (>150), it is significantly low for EFITARM as compared to FITI logic (p-value<0.01). As number of days increase, difference of time taken by two logics is significantly increases.
Overall comparison of four logics with EFITARM algorithm on the basis of mean time taken to run the output at different threshold levels is given in above figure. Average time taken by logics – 1 & 2 and logics 3 & 4 are almost same. Mean time taken by logics – 1 & 2 is significantly lower than logics 3 & 4 (p-value <0.01).

Figure 4.25: Comparison of average time taken by EFITARM between four logics for different threshold values
Overall comparison of four logics with FITI algorithm on the basis of mean time taken to run the output at different threshold levels is given in above figure. Average time taken by logics – 1 & 2 and logics 3 & 4 are almost same. Mean time taken by logics – 1 & 2 is significantly lower than logics 3 & 4 (p-value <0.01).
Figure 4.27: Comparison of average time taken by EFITARM between four logics for different number of days

Mean time taken by EFITARM algorithm for four logics is compared in above figure for different number of days. Up to 250 days, no significant difference is seen between four logics in mean time taken by algorithm whereas for higher number of days (>250), it shows significant difference between logics - 1 & 2 with logics 3 & 4 (p-value <0.01). As number of days increase difference of mean time taken by EFITARM algorithm between logics is also significantly increase.
Mean time taken by FITI algorithm for four logics is compared in above figure for different number of days. Up to 150 days, no significant difference is seen between four logics in mean time taken by algorithm whereas for higher number of days (>150), it shows significant difference between logics 1 & 2 with logics 3 & 4 (p-value <0.01). As number of days increase difference of mean time taken by FITI algorithm between logics is also significantly increase.