Impact Factor 3.025

ISSN 2349-638x

Refereed And Indexed Journal

AAYUSHI INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (AIIRJ)

UGC Approved Monthly Journal

VOL-IV ISSUE-X Oct. 2017

Address
· Vikram Nagar, Boudhi Chouk, Latur.
· Tq. Latur, Dis. Latur 413512 (MS.)
· (+91) 9922455749, (+91) 8999250451

Email
· aiirjpramod@gmail.com
· aayushijournal@gmail.com

Website
· www.aiirjournal.com

CHIEF EDITOR – PRAMOD PRAKASHRAO TANDALE

Verification of Match Fixing in Cricket using Sports Data Mining and Algorithmic Design Notation

K.Ananthapadmanabha
Research Scholar,
PP COMP.SCI.ENG.0251
Computer Science & Engineering
Rayalaseema University, Kurnool,
siridevikap@yahoo.co.in

Dr.K.Udayakumar
Principal and Professor,
Department of Computer Science and Engineering,
Adarsha Institute of Technology,
India Bangalore, India

Abstract

Sports Data Mining is the method of extracting unseen patterns from the sports data. Nowadays it is used in professional sports the world over. Cricket is the second most popular sport in the global sport arena. Cricket has been part of major controversies like doping, ball tampering and latest one is that of match fixing. To verify match fixing in cricket, we are proposing algorithmic design notation to verify underperformance by participants and favours received by participants as evidences for match fixing.

Key Words: Sport Data Mining, Cricket, Match Fixing, Algorithmic Design Notation.

Introduction

Match fixing in cricket is a ground reality today. With more and more cricket match being played worldwide, frequency of occurrences of match fixing is also increasing. Match fixing in cricket is possible due to illegal collaboration between players, umpires, officials, bookies and middlemen. To verify match fixing in cricket we need to provide evidence for match fixing like:

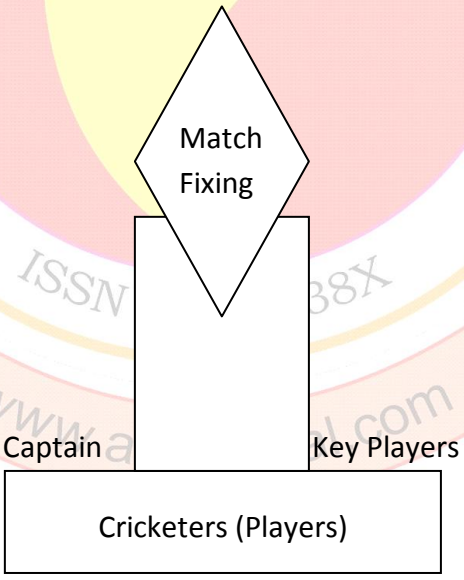


Fig.: Recurring Relationship Representation between Captain and Key Players Involved in Match Fixing

- Under performance by match participants
- Favours received by participants for under performance
- Criminal Network Analysis

In this paper we propose algorithm to prove under performance by match participants and related favours received by the participants for under performance.

Sports Data Mining

In today's world, sports is not only played for entertainment, it has moved beyond entertainment and now it is a multi trillion industry with many sports based enterprises investing multiple billions for their operations. This industry today has huge volumes of sports data across all domains of sports. This data can be with respect to individual player performance, team performance, tournament details and game details. All these data can be used for professional purposes like team selection, captain decision making process, for coaching and managerial decision process. It can also be used for trend analysis, sponsorship, sports management, talent recognition details, prediction of match outcomes, controversies and crisis in sports like doping and match fixing. It may help in fine tuning fitness level of players and team. Individual team decision making can use this data for competitive advantage against their opponents in understanding teams strength and weakness analysis, venue details etc.

Sports data mining deals with sports data in all domain of sports like football, cricket, volleyball, hockey etc., expertise available in the related sports domain like commentators, umpires, official details, Software packages help in analyzing sports data and the latest research happening in these fields. Data stored in sports data warehouses have huge volume of data, representing hidden relationship which when mined can provide competitive advantages. Sports data may be in the form of comments and reviews stored in social media like twitters, facebook to name a few which can be analyzed using data mining techniques like opinion mining or sentiment analysis, background knowledge of sports can be analyzed using Ontology based mining for purposes like Ontology mapping, expertise matching, opinion spam detection etc. Review mining can be conducted for verifying reviews stored in sports reviews on the social media by both viewers and domain experts views, emotions on a particular topic, event or game. Sport data mining is the method of extracting unseen patterns from sports data in professional sports and today it deals with art of winning an unfair game.

Methodology

We are proposing the following algorithms for verifying match fixing in cricket.

1.Matchfixing()

This algorithm checks whether any of the team involved in cricket match have fixed the match. Important attributes considered are team name, toss result, toss decision. If team A wins the toss and invites B for batting then we invoke function Batting_performance(B) and Bowling_performance(A). Otherwise if team A decides to bat first than we invoke Batting_performance(A) and Bowling_performance(B).

If team B wins the toss and invites A for batting then we invoke function Batting_performance(A) and Bowling_performance(B). Otherwise if team B decides to bat first than we invoke Batting_performance(B) and Bowling_performance(A).

2.Batting_performance()

The algorithm batting_performance(team_name) is used to verify batting performance of each player of the team. Important attributes considered here are playerid[i], runsscored[i], careerstrikerate[i], numberofballsfaced[i] and we calculate playersmatchstrikerate[i] and averagerunsscored[i].

3.Bowling_performance()

In algorithm bowling_performance(team_name) we try to verify bowling performance of players of a given team. The input for this algorithm is bowler score book details. Important attributes used are playerid[i], playertype[i], numberofoversbowled[i], numberofrunsgiven[i], numberofwicketstaken[i], playercareerbowlingstrikerate[i].

If player_type[i] = "Bowler" or player_type[i] = "Allrounder"

then calculate player[i] matchstrikerate[i] and we invoke the function bowlerunderperformance(teamname).

4.Batting_underperformance()

The algorithm batting underperformance(team_name) accepts team name as input with related batting score book details and batting threshold expected performance. This algorithm returns 1 if batting under performance of any player is true otherwise it returns 0.

5.Bowling_underperformance()

The algorithm bowling underperformance(team_name) is used to verify under performance of any bowler in a given team. The input for this algorithm are team_name, player id, bowler bowling details. This algorithm returns 1 if any bowler in this team is under performing in this team otherwise it returns 0.

6.Favours_received()

To prove match fixing only providing evidence for under performance by players is insufficient, we need to provide evidence for related favours received by participants. To provide this proof. We are proposing algorithm Favoursreceived(team_name, player_id). The objective of this algorithm is to check whether any player who has under performed has received any favours from bookies. Important attributes considered for favours received are income of player before match, income of player after match and income threshold value. If (income_after_match – income_before_match) exceeds income threshold value then we agree that player has received favours otherwise we say no favours are received.

List Of Algorithms Proposed

ALGORITHM Matchfixing( )
// **PURPOSE:** To check matchfixing
// **INPUT:** Match score book
// **OUTPUT:** Returns 1 if match fixing is true else returns 0

Step 1: For I = 1 to 2 do
Begin
Step 2: Accept Toss_result, Team_name
Step 3: If Toss_result = “Won” and If ((Team_name = “A”) and (Toss_result = “Won”)) Then
Step 4: Accept Team A decision
Step 5: If Team A decision = “Batting” then
Step 6: Call Batting_performance(A)
Call Bowling_performance(B)
else
Step 7: Call Batting_performance(B)
Call Bowling_performance(A)
endif
endif
endif
Step 8: If Toss_result = “B” Then
Step 9: Accept TeamBDecision
Step 10: If TeamBDecision = “Batting” Then
Step 11: Call Batting_performance(B)
Call Bowling_performance(A)
Else

Step 12: Call Batting_performance(A)
Call Bowling_performance(B)
endif
endif
Step 13: Call favours_received(A)
Step 14: Call favours_received(B)
Step 15: If ((Team_type = "A") and (under_performance = 1) and (favours_received = 1)) Then
Match_fixing_of_A = 1
Else
Step 16: Match_fixing_of_A = 0
Return match_fixing_of_A
Endif

Step 17: If ((Team_type = "B") and (under_performance = 1) and (favours_received = 1)) Then
Match_fixing_of_B = 1
Else
Step 18: Match_fixing_of_B = 0
Endif
Step 19: If (Match_fixing_of_A = 1) Or (Match_fixing_of_B = 1)) Then
Match_fixing = 1
Else
Step 20: Match_fixing = 0
Endif
Step 21: If ((Match_fixing = 1) and (Match_fixing_of_A = 1)) Then
Print "Match is fixed by teams"
Endif
Step 22: If ((Match_fixing = 1) and (Match_fixing_of_B = 1)) Then
Print "Match is fixed by Team B"
Endif
Step 23: If (Match_fixing = 0) Then
Print "Match is Not Fixed"
Print "Match is a Fair Game"
Endif
Step 24: return match_fixing
End

ALGORITHM Batting_performance(x)
// **PURPOSE:** To know batting performance of given team x can be A or B teams
// **INPUT:** Match score book with batting details
// **OUTPUT:** To know batting performance of team x
Step 1: For I ← 1 to 11 do
Begin
Step 2: Accept playerid [1]
Step 3: Accept run_scored [1]
Step 4: Accept carreer_strike_rate [1]
Step 5: Accept match_strike_rate [1]
Step 6: Accept number_of_balls_faced [1]

Step 7: Calculate Average runs scored [1]
Step 8: Call Batting under_performance(x)
End

ALGORITHM Bowling_performance(z)
// **PURPOSE:** To verify bowling performance of given team Z
// **INPUT:** Bowling score book details
// **OUTPUT:** To know individual players bowling performance

Step 1: For 1 ← 1 to 11 do
Begin
Step 2: Accept player_id [1], player_type
Step 3: If player_type = "Bowler"
Step 4: Accept Number of over bowled, Number of runs given
Step 5: Accept Number of wickets taken
Step 6: Accept Career bowling strike rate
Step 7: Accept match strike rate
Step 8: Accept threshold [1]
Step 9: Call Bowling under_performance (Z)
endif
End

ALGORITHM Batting_underperformance(p)
// **PURPOSE:** To check whether any player of a team has underperformed during batting
// **INPUT:** Batting score book details
// **OUTPUT:** Return 1 if batting underperformance is true else return 0.

Step 1: Flag_up = 0
Step 2: For I ← 1 to 11 do
Begin
Step 3: If (Careerstrikerate – Matchstrikerate >= threshold)
Flag_up = 1
Endif
Step 4: return flag_up, player_id [1], teamname
End

ALGORITHM Bowling_underperformance(q)
// **PURPOSE:** To verify bowling underperformance by members of a team
// **INPUT:** Bowling score book details
// **OUTPUT:** Return flag Blower_up = 0 if no underperformance else return 1 if underperfornace by any blower

Step 1: Flag_up = 0
Step 2: If (Careerstrikerateofbowling – Matchstrikerateofbowler > threshold) then
Flag_bowler_up = 1
Endif
Step 3: return flag_bowler_up
End

ALGORITHM Favours_received(x)
// **PURPOSE:** To check whether any form of favours received by player of a team
// **INPUT:** Player id, income before and after match
// **OUTPUT:** Returns 1 if favour is received else return 1

Step 1: For I ← 1 to 11 do
Begin
Step 2: Accept Incomebeforematch [1], Incomeaftermatch [1]
Step 3: Accept threshold
Step 4: If ((Incomeaftermatch [1] – Incomebeforematch [1]) > threshold
Flag_FR[1] = 1
Else
Step 5: Flag_FR [1] = 0
Step 6: If flag_FR [1] = 1
Step 7: return playerid [1], flag_FR [1]
Endif
Endif
End

ALGORITHM Avgrunsscored(x)
// **PURPOSE:** To calculate average runs scored
// **INPUT:** player performance details
// **OUTPUT:** Average runs scored by a player
Step 1: Accept teamname
Step 2: For I ← 1 to 11 do
Begin
Step 3: Avgrunsscored [1] = runsscored [1] / numberofballsfaced [1]
Step 4: return Avgrunsscored
End

Criminal Network Analysis

In a criminal network, we need to identify name of criminals, role played by criminals in the network flow of measurable, qualitative, quantitative goods and information and association among these objects. A network is composed of captain vertex, groups and subgroups inside the network. If captain of the network has a low profile then finding the real captain of the network is very difficult to identify.

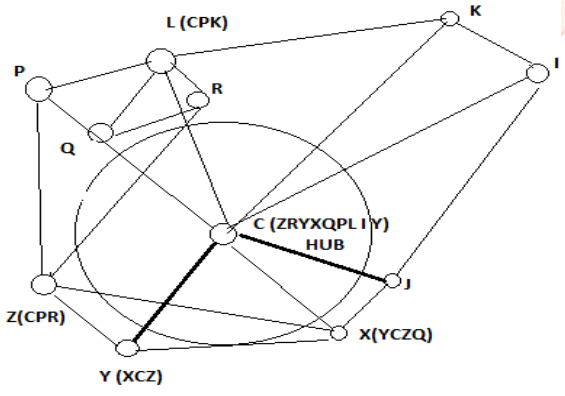
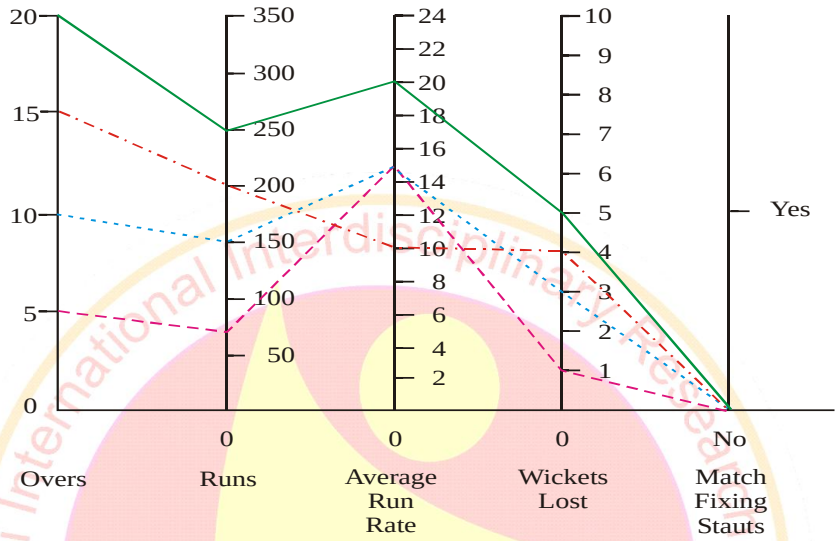


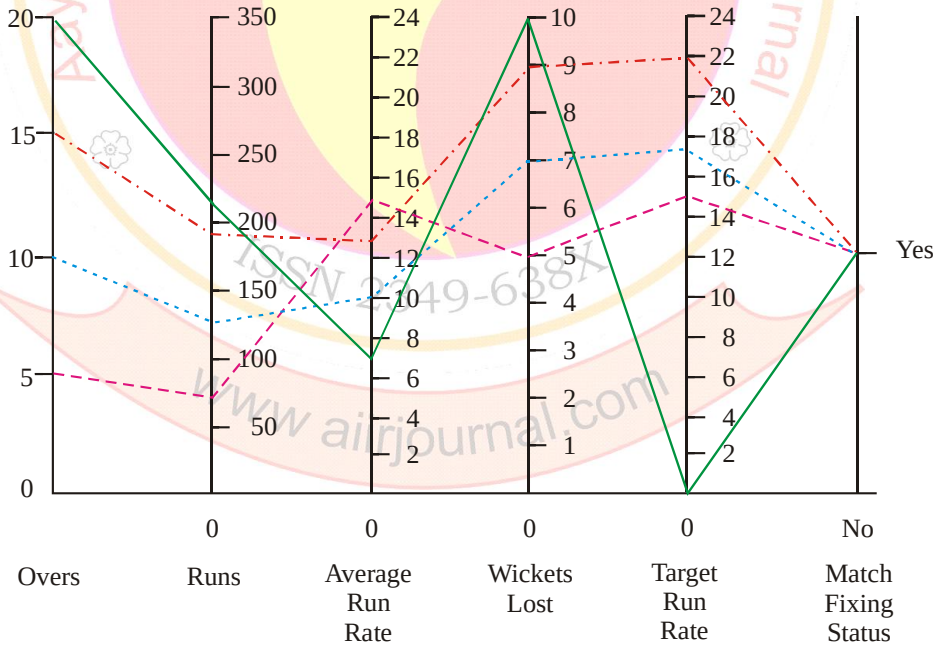
Fig.: Criminal Network

Here, entity extraction technique helps in identifying patterns from audio, video clipping, documents, images and email. This helps in identifying name of the person, his address and personal features. By implementing computer forensics, by using algorithm, data, data structures, program control flow, internal documentations by using comment statements, use of variable names helps in extending criminal investigation.Quality of criminal investigation based on computer forensic depends on high quality clean input data, with less noise, no missing data. Appropriate Extract, Transform Load (ETL) methods need to be identified and implemented.

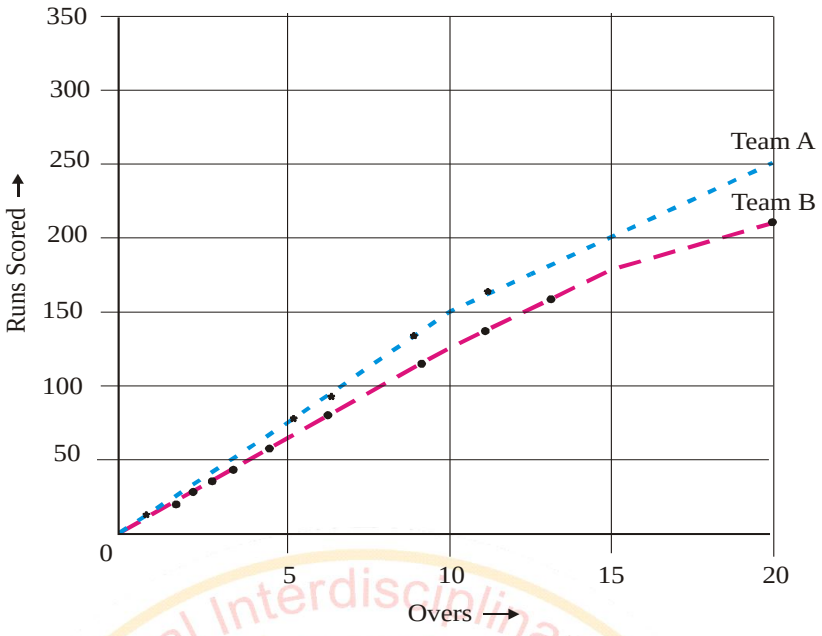
Results



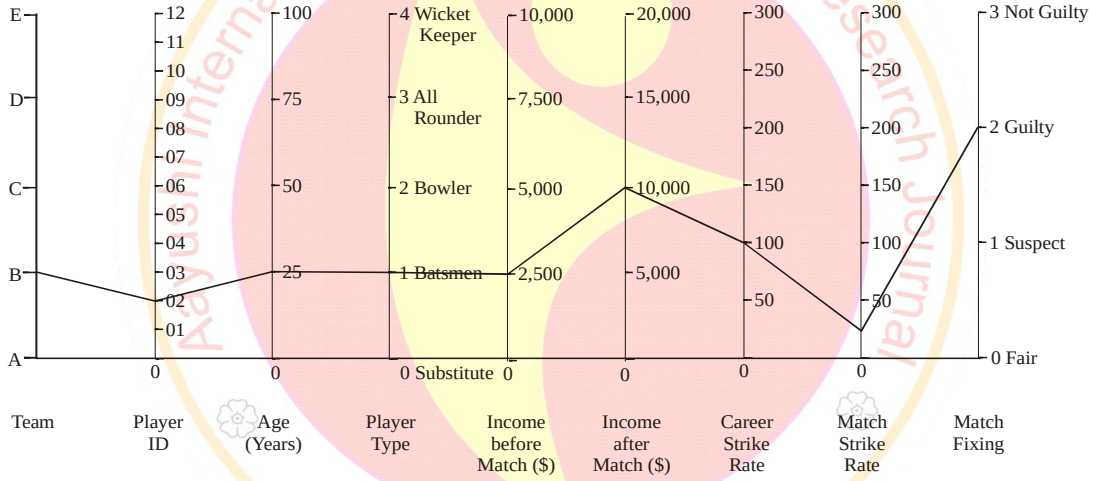
Graph 1: Iceberg Diagram for Team A (Target Setter)



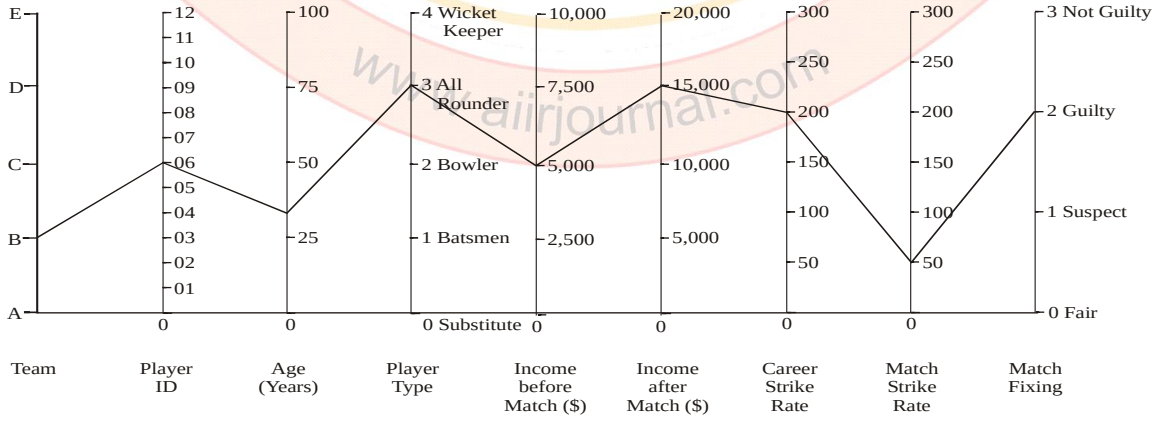
Graph 2: Iceberg Diagram for Team B (Target Chaser)



Graph 3: Line Graph for Comparison between Team A and Team B Performance



Graph 4: Iceberg Diagram for Representing Player ID 02 involved in Match Fixing



Graph 5: Iceberg Diagram for Representing Player ID 06 involved in Match Fixing

Match indicates how target chaser team B is a match fixer. In this match team B after winning the toss invites team A to bat first. Team A sets a target of 250 in 20 overs. While chasing, team B in the first 5 overs (1-5) scores 75 runs at an average of 15 runs per over but losses 5 wickets. In the next 5 overs (6-10) it scores only 50 runs at an average of 10 runs per over by losing 2 wickets. In the next 5 overs (11-15) it scores 65 runs at an average of 13 runs per over, losing 2 wickets. In the last 5 overs (16-20) it scores 20 runs at an average of 4 runs per over.

In match, Team B player with player ID 02 who is a batsmen with career strike rate of 100 has scored in the match at a strike rate of 25. His income before match was $2,500 and his income after match is $10,000. This clearly indicates his involvement in match fixing. Also another player from team B with a players ID 06 who is an all-rounder with a career strike rate of 200 has a match strike rate of 50 in this match. His income before match was $5,000 and his income after match is $15,000. This clearly indicates his involvement in match fixing.

Conclusion

Today, with popularization of cricket match fixing is a ground reality. Sports data mining and algorithmic design notation are suitable for solving such problems. Player under performance and favours received along with criminal network analysis helps in providing proof for match fixing in cricket.

References

1. Jiawei Han and Micheline Kambe "Data Mining Concepts and Techniques", 2nd Edition, Morgan Kaufmann publishers An imprint of Elsevier.
2. K.Udayakumar and K.Ananthapadmanabha, KL Cluster Nearest Neighbor Outlier Prediction Algorithm for Match Fixing in Cricket, International Journal of Advances in Electronics and Computer Science, Special Issue, Sep.-2016, pp. 79-81, ISSN: 2393-2835.
3. R.V.Hauck *et.al*, "Using Coplink to Analyze Criminal Justice Data", Computer, March, 2002, 30-37.
4. G.K.Gupta, "Introduction to Data Mining with Case Studies", PHI Learning Private Limited.
5. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Pearson Publications
6. Alex Berson, Stephan J Smith "Data Warehousing, Data Mining and OLAP", Tata McGraw Hill Edition publication.
7. K.Udayakumar and K.Ananthapadmanabha, Algorithmic Design Notation for Match Fixing in Cricket Using Outlier Analysis, Higher Education Conclave 2016, p. 31, 2016, ISBN No. 978-81-8281-575-9.
8. Pang-Ning Tan, Michael Steinbach and Vipin kumar "Introduction to Data Mining", Pearson Education publication
9. K.Udayakumar and K.Ananthapadmanabha, Forest Fire Model Proposal for Match Fixing in Cricket Based on Criminal Network Analysis, International Journal of Engineering Research, Volume No.5 Issue: Special 4, pp: 790-991, May 2016, doi: 10.17950/ijer/v5i4/016, ISSN: 2319-6890 (online), 2347-5013(print).