**CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY (CUET) DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CHITTAGONG – 4349**

f

**(Thesis Proposal)**

**Application for the Approval of B.Sc. Engineering Project**

**(Computer Science & Engineering)**

**Date:** 08-10-2018

**1. Name of the student :** Ratul Roy

**Student ID Session**

**:** 1404010

**:** 2014-2015

**2. Present address :**  Shaheed Tareq Huda Hall, CUET.

**3. Name of the Supervisor :** Prof. Dr. Mohammad Shamsul Arefin

**Designation :** Professor & Head of the Department,

Department of Computer Science & Engineering, Chittagong University of Engineering &Technology.

**4. Name of the Department :**

**Program :**

**5. Date of First Enrolment**

**In the Program :**

**6. Tentative title :**

Computer Science & Engineering (CSE) B.Sc. Engineering.

March 18, 2015.

**Proposing Optimum Team Line-Up Based on Predictive Analysis of Previous Match Commentary**

**7. Introduction**

Predicting an individual athlete’s performance based upon his/ her past record can be critical in the selection of team members in international competitions. This process is highly subjective

and usually requires much expertise and negotiative decision making. In this study, we explore the analysis of related commentary and user comments in such decision situations. We take the case of team selection in international contests in cricket.

In a team selection committee environment multiple members evaluate each player’s performance and vote for inclusion/exclusion from the team. These selection committee members provide rankings for cricketers. Negotiations are then conducted to produce an agreement among the selection committee members as to which cricketer should be finally recommended to be selected to the team. We simulate this process by forecasting a player’s performance using predictive analysis.

Cricket is a popular game played by a few countries. There are two versions of the game – Test Cricket which is played over five days and one-day cricket, which is obviously played over a day. One-day cricket was introduced in the English domestic season of 1963 due to the growing demand for a shorter and more dramatic form of cricket to stem the decline in attendance. One-day, single-innings matches often took place before this, but the innovation was the limiting of each side’s innings to an agreed number of lovers (nowadays usually 50). The inaugural 1975 World Cup was a great success. The abbreviations ODI (One-day International) or sometimes LOI (Limited Overs International) are used for international matches of this type. Frequent nail-biting finishes and the impossibility of either side opting to play for a draw have seen ODI cricket gain many supporters.

In this project, we’ll try to rank team players based on their past performances in limited number of recent matches, from the commentary and comments. For commentary we’ll use the archives of cricinfo[3], cricbuzz[4] etc websites.

**8. Related Work**

We have found that only a few studies have been published on rating players, forming teams and analyzing commentaries.

Duckworth and Lewis [1] have developed innovative rain interruption rules that are extensively used in one-day cricket matches. Their methods differ from previous approaches in that they take into account the available run scoring resources (overs and wickets) the two teams have left. Overall, the more unused run-scoring resources a team at their disposal at the end of an interrupted innings the more runs they would score if not interrupted.

S. R. Iyer & R. Sharda[2] did excellent work on rating players and player selection. They’ve explored the use of neural networks to rate players and select specific players for a competition.

**9. Objectives**

The thesis will be carried out to achieve following goals:

1. To develop a framework to rank players based on their past performances
2. To analyze sports commentary and evaluate players based on it
3. To suggest best players for next

**10. Methodology**

In this thesis experiment, we’ll collect commentary data from websites and analyze them to score players on some simple rules.

The system architecture of the framework comprises 5 basic modules; Website Access Module, Data Scraping Module, Data Storing Module, Player Ranking Module and Recommending Module. The system architecture is showed in Figure 1.

In the Website access module first, we have to establish connection to get access from Website. That connection will allow us to get access Website API[6][7]. The Website API will give the pathway to get access tokens for collecting data from Website.

Website are deeply concern about bandwidth. So they limit api requests which can be extended by paying a minimum amount. By using the access token which one we have gotten from access module, in the data crawler module we can through data through a crawler.

In Data storing module, we will store our data separately after light filtering if there exists any possibility. It will make our data more reliable. Along with collected data, this module also handles storage of important information for efficient retrieval purpose.

In Data filtering module, we will filter and extract data. We will use text segmentation to probe through the commentary and find player specific sentences. Then through filtering we’ll use these sentences to assign points to each player via specific rules. This process is beneficiary for the top eleven player combination.

After finding useful commentary that can help us rate players, we’ll follow some rules to give rankings among the players. This is the Ranking module’s job. The ranking module have to consider optimum numbers of bowlers and batsman along with a wicket-keeper. Just plainly ranking would not satisfy a balanced team needs.

So we’ll have to rank each player both as a bowler and as a batsman – separately. Then fulfilling a minimum quota, we’ll suggest top eleven for the team.

Website Access Data Filtering



Developer

API

Textual Data

Recommending

Data Crawler

Access token debugger

Rest client library

Filtering

Rating

Data Crawling

Ranking Module

PHP Scraper/

API

Access token

Calculate Rating

Rate The Players

HTML Scraping

Rank Players

by Rating and Quota

Data Storing

Filter

Usable

Commentary

Database

Suggest Top eleven

Figure 1: System Architecture

In the ranking module, this will be done through matching tag which we will extract from his profile.

The database initialization and processing module consists of some sub-modules: Parsing through HTML texts, retrieving commentry, data crawling and rating, separating usable commentry, storing data into database, text segmentation, suggesting op eleven. The relationships among this sub module are showed elaborately in Figure 2.

Websites

Database

Scraper/

API

No

No

Filtered?

Have access token?

Yes

Yes

Find Rating

Ranking

Scrape Data

Collect Commentry

Final Player Combination

Call on Custom Rules

Figure 2: Data flow diagram

**11. Required Resources**

Resources required to complete this task are listed below:

1. Personal Computer

2. Operating System (Linux)

3. Apache Server (XAMPP)

4. IDE (jet brains)

5. Text Editor (sublime text)

5. Hbase

**12. Cost Estimation**

The costs that will occur to implement our proposed system are estimated below:

|  |  |
| --- | --- |
| **Materials** | **Cost (BDT)** |
| Internet | 2000 |
| Paper | 400 |
| Printing | 800 |
| Binding | 150 |
| Typing | 150 |
| Miscellaneous | 500 |
| **Total** | 4000 |

**13. References**

[1] F. C. Duckworth and A. J. Lewis, “A fair method for resetting the target in interrupted one-day cricket matches”, J. Oper. Res. Soc. , 49, (1998), 220-227.

[2] S. R. Iyer, and R. Sharda, “Prediction of athletes performance using neural networks: An application in cricket team selection”, Expert Systems with Applications, 36, 5510 (2009)

[3] ESPN Cricinfo Website, <http://espncricinfo.com/>

[4] Cricbuzz website, <https://www.cricbuzz.com/>

[5] CricketAPI, <https://github.com/dynamitechetan/CricketAPI>

[6] Cricbuzz for Python, <https://github.com/codophobia/pycricbuzz#cricbuzz-for-python>

**14. CSE Undergraduate Student (CUGS) Committee reference**

**Meeting No: Resolution No: Date:**

**15. Number of Under-Graduate Student(s) working with the**

**Supervisor at Present**: 12

Signature of the Student(s)

Signature of the Supervisor

Signature of the Head of the Department