

Cricketer's Tournament-Wise Performance Prediction and Squad Selection Using Machine Learning and Multi-Objective Optimization

Project & Thesis-II

CSE 4250

A thesis Report

Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Noorun Nashfin	160204030
Devopriya Tirtho	160204033
Shafin Rahman	160204040
Kazi Tunaz Zina	160204052

Supervised by

Prof. Dr. Md. Shahriar Mahbub



Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

July 03,2021

CANDIDATES' DECLARATION

Dr. Md. Shahriar Mahbub, Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh, has supervised our study, and the thesis provided in this paper is the result of that investigation. According to the Department's course curriculum for the Bachelor of Science in Computer Science and Engineering degree, the work was spread out across two final year courses, **CSE4100: Project and Thesis I** and **CSE4250: Project and Thesis II**. It is further stated that no part of this thesis, or any part of it, has been submitted to any other institution for the granting of a degree, certificate, or other qualification.

It is also declared that neither this Thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications

Noorun Nashfin [16.02.04.030]

Devopriya Tirtho [16.02.04.033]

Shafin Rahman [16.02.04.040]

Kazi Tunaz Zina [16.02.04.052]

CERTIFICATION

This thesis titled, “Cricketer’s Tournament-Wise Performance Prediction and Squad Selection Using Machine Learning and Multi-Objective Optimization”, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in July 2021.

Group Members:

Noorun Nashfin [16.02.04.030]

Devopriya Tirtho [16.02.04.033]

Shafin Rahman [16.02.04.040]

Kazi Tunaz Zina [16.02.04.052]

ACKNOWLEDGEMENT

The first and most important thing we can say is that we are thankful to Almighty Allah for providing us with the excellent health and well-being that we needed to complete this thesis. The next step is to convey our sincere appreciation to our esteemed supervisor, Professor Dr. Md. Shahriar Mahbub, for his invaluable advice and support during the preparation and production of our undergraduate thesis. His guidance and encouragement have been invaluable to us over the years, and he always encourages us to learn new things and experiment with new techniques. We are very thankful that he agreed to serve as our thesis supervisor and mentor throughout our undergraduate project. We would also like to sincerely thank Prof. Dr. Mohammad Shafiul Alam, the department's head, as well as all of our respected faculty members at the Department of Computer Science and Engineering at the Ahsanullah University of Science and Technology for their assistance in providing us with some useful resources and valuable advice. We would have been unable to finish our research project without their assistance. Lastly, we would want to express our gratitude to the individuals who have helped us get this far, our parents, for their unwavering support, blessings, and unconditional love, which have enabled us to make significant strides. We would like to express our heartfelt gratitude to our friends for always being there for us and assisting us with a variety of issues and psychological assistance.

ABSTRACT

The sporting world receives significant investment as a result of sporting events, which provide both excellent entertainment and significant revenue to the sport's industry. These occurrences occur on a regular basis in every region of the globe and take on various shapes. Franchise-based cricket tournaments are becoming more and more popular every day. These events attract players from all around the world who come to compete in them. As computer technology has blessed us with the ability to mimic the human brain, it has also given us the ability to analyze historical data in order to predict the future. We take use of this chance to forecast player performance, and based on the forecasted performance, we attempt to construct some optimum teams that may be used to assist team owners in forming squads for competitions. We learn how to implement the prediction process using machine learning methods in this article, and we see how multi-objective optimization approaches may help us build optimum squads in a more precise manner by utilizing the predicted performances.

Contents

CANDIDATES' DECLARATION	ii
CERTIFICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
List of Figures	iii
List of Tables	iv
1 Introduction	1
1.1 Overview	1
1.2 Problem Description	2
1.3 Motivation	4
2 State of the Art	6
3 Background Study	19
3.1 Naive Bayes Classifier	19
3.2 Support Vector Machine Classifier	20
3.3 Decision Tree Classifier	21
3.4 K-Nearest Neighbors Classifier	21
3.5 Random Forest Classifier	23
3.6 NSGA-II Algorithm :	24
3.7 SPEA2 Algorithm :	25
4 Methodology	30
4.1 Methodology of Machine Learning Algorithm Implementation Part	31
4.1.1 Data Collection	31
4.1.2 Data Cleaning	34
4.1.3 Calculation of Batting Features	35
4.1.4 Calculation of Bowling Features	38

4.1.5	Calculation of Fielding Features of Wicket-Keepers	41
4.1.6	Calculation of Travelling Features	42
4.1.7	Rating of Features' Values	42
4.1.8	Calculation of Derived Features	53
4.1.9	Method of Prediction Using Machine Learning Algorithm	56
4.2	Methodology of Multi-Objective Optimization Implementation Part	57
4.2.1	Data Collection	58
4.2.2	Data Preprocessing	59
4.2.3	Problem Formulation	64
4.2.4	Objective Formulation	66
4.2.5	Constraint Formulation	67
4.2.6	Repair Function Formulation	67
4.2.7	Selection of Multi-objective Optimization Algorithm	70
5	Result Analysis	73
5.1	Result Analysis of Machine Learning Phase	74
5.1.1	Prediction of Performance Using Naive Bayes Classifier Algorithm .	74
5.1.2	Prediction of Performance Using Support Vector Machine Classifier Algorithm	76
5.1.3	Prediction of Performance Using K-Nearest Neighbors Classifier Al- gorithm	78
5.1.4	Prediction of Performance Using Decision Tree Classifier Algorithm	81
5.1.5	Prediction of Performance Using Random Forest Classifier Algorithm	83
5.1.6	Comparison of results among all the Classifiers	85
5.2	Result Analysis of Multiobjective Optimization Phase	87
5.2.1	General Experimental Settings	87
5.2.2	Squad Selection using SPEA2	93
5.2.3	Performance Evaluation between NSGA-II and SPEA2	98
6	Future Work	100
6.1	Limitations	100
6.2	Possible Future Approach	101
7	Conclusion	103
	References	105
A	Resources and Data sets	109

List of Figures

3.1	K-Nearest algorithm	23
3.2	Random Forest algorithm	23
3.3	NSGA-II algorithm	25
4.1	Flowchart of Methodology	31
4.2	Unprepared Dataset of Bowlers	32
4.3	Traveling Dataset	33
4.4	Prepared Dataset for Research Work	34
4.5	Multi-objective Optimization Dataset	66
5.1	Accuracy Comparison for 80:20 Split	87
5.2	Different Views of 3D True Pareto Front for NSGA II.	89
5.3	Different Views of 3D True Pareto Front for SPEA2.	93
5.4	Multi-objective Optimization Dataset	98

List of Tables

1.1	Number of Players of Different Roles	3
2.1	Summary of literature review	15
4.1	Rating of Features of All Types of Batters	45
4.2	Bowler's Rating	50
4.3	Class-wise Run for Batters	59
4.4	Class-wise Wicket for Bowlers	60
4.5	Formation of Squad of 23 Players in Squad	68
4.6	Position of Top-order Batters	68
4.7	Position of Wicket-keepers	68
4.8	Position of Middle-order Batters	69
4.9	Position of All-rounders	69
4.10	Position of Bowlers	69
4.11	Comparison of different MOEAs	71
5.1	Naive Bayes Classifier Accuracy for Top Order Batters	75
5.2	Naive Bayes Classifier Accuracy for Middle Order Batters	75
5.3	Naive Bayes Classifier Accuracy for Lower Order Batters	76
5.4	Naive Bayes Classifier Accuracy for Bowlers	76
5.5	Support Vector Machine Classifier Accuracy for Top Order Batters	77
5.6	Support Vector Machine Classifier Accuracy for Middle Order Batters	77
5.7	Support Vector Machine Classifier Accuracy for Lower Order Batters	78
5.8	Support Vector Machine Classifier Accuracy for Bowlers	78
5.9	K-Nearest Neighbors Classifier Accuracy for Top Order Batters	79
5.10	K-Nearest Neighbors Classifier Accuracy for Middle Order Batters	79
5.11	K-Nearest Neighbors Classifier Accuracy for Lower Order Batters	80
5.12	K-Nearest Neighbors Classifier Accuracy for Bowlers	80
5.13	Decision Tree Classifier Accuracy for Top Order Batters	81
5.14	Decision Tree Classifier Accuracy for Middle Order Batters	82
5.15	Decision Tree Classifier Accuracy for Lower Order Batters	82
5.16	Decision Tree Classifier Accuracy for Bowlers	83
5.17	Random Forest Classifier Accuracy for Top Order Batters	84

5.18 Random Forest Classifier Accuracy for Middle Order Batters	84
5.19 Random Forest Classifier Accuracy for Lower Order Batters	85
5.20 Random Forest Classifier Accuracy for Lower Order Bowlers	85
5.21 Comparison of All Classifiers for Prediction Class of Runs and Wickets	86
5.22 General parameter settings for NSGA-II and SPEA2	88
5.23 Batting Oriented Team Formation	90
5.24 Bowling Oriented Team Formation	91
5.25 Cost-Effective Team Formation	92
5.26 Batting Oriented Team Formation	95
5.27 Bowling Oriented Team Formation	96
5.28 Cost-Effective Team Formation	97
5.29 Mean and Standard Deviation for Hypervolume Values of NSGA-II and SPEA2	98
5.30 Mann-Whitney U-test	99

Chapter 1

Introduction

1.1 Overview

Cricket is a game in which two teams compete against each other on a field known as a cricket ground. The field consists of a pitch of three stumps grounded with bails at either end of the pitch. When a cricket match is organized, the management of each team announces a squad of about twenty three to twenty five players, with eleven players from each team participating in the match. Since the game's inception, cricket has been recognized as a gentleman's game. The game begins with a coin flip between the captains of the two teams to see if the winning captain will bowl or bat first. The batting side attempts to score as many runs as possible in the first half of the game in a finite number of overs. The bowling team attempts to bowl them out in the shortest period of time possible. The first half of the game stops after a certain number of overs are completed or when the batting side is bowled out. Later in the game, the bowling team enters the field to bat and score at least one more run to win the game. The other side attempts to keep them as far away from the aimed run as possible. Cricket is a thrilling sport in which each team competes to win. When a tournament is held, more than two teams compete against one another for the silverware. Since winning a match is dependent on the members of a squad, the team should be chosen in such a way that all of the top performers have a chance to play. Since a cricket competition takes several days to schedule further matches, a team must be formed such that if a player is wounded in the middle of the tournament, no problems arise. Again, the purpose of squad selection is to choose a team of appropriate players who can compete against teams of varying approaches. As a result, a certain player's picks are fairly relevant in terms of winning a match. Cricket tournaments for twenty overs are now more common than any other form of cricket tournament. Franchise-based twenty-overs tournaments are held in almost every cricketing country, with a significant number of local and international players participating. These competitions schedule a series of matches in

a variety of venues over a limited period of time, requiring teams to fly to participate. These franchise-based tournaments go by different names; in Australia, the tournament is known as Big Bash. In Bangladesh, the league is known as the Bangladesh Premier League, while in India, it is known as the Indian Premier League. In Pakistan and the West Indies, the leagues are known as the Pakistan Premier League and the Caribbean Premier League, respectively. When computer technology evolves, people continue to integrate various computer science approaches into every area of life to solve various problems. The performance of the players is the most significant consideration in the formation of a team for a tournament. The performance of a player can be calculated approximately by learning the patterns of a player in a machine and applying different algorithms. Machine learning is a method that uses a machine to study and then attempts to predict based on the learning. Machine learning is a sub-field of Artificial Intelligence that attempts to learn from data feeds and function accordingly. Machine Learning is used in many areas of our daily lives, such as healthcare, computer vision, predicting the outcomes of specific events, and so on. Several algorithms have already been developed to support the purpose of Machine Learning. To solve different problems, they use Naive Bayes, Support Vector Machines, Random Forest, Decision Tree, Linear Regression and Logistic Regression. Automation in different fields of operation may be accomplished with the assistance of Machine Learning. The continuation of the outcome of machine learning is enormous, and the most optimized outcomes are attractive from the results. Another strategy, known as optimization, is available to solve this problem. In the functional area, optimization is extremely important. When working with a vast amount of data and analyzing each data point, the assistance of computer science is needed. When we need the most optimized solutions from a set of options, the implementation of optimization techniques will help. Optimization may be performed by taking into account one or more objectives while keeping various constraints in mind to provide optimal outcomes. In this report, we will explain the use of machine learning to forecast cricket results, and we will use the Multi-objective Optimization Technique to find the most optimal squads for a specific tournament.

1.2 Problem Description

The formation of a strong team seems to be a significant term in all sports around the world. Other games, such as football, soccer, baseball, and basketball, require a large number of well-performing, healthy players to form a squad to compete in a competition. Since the sports universe is all about competition, team management is constantly on the lookout for the best talent and recruiting them to their squads. In franchise-based competitions, players are not required to participate with a certain team if they receive a stronger bid from another team. In today's cricket, a franchise-based twenty overs competition will first hold a draft or

auction to allow all team owners to form a team that they believe will win the tournament for them. The team manager begins bidding for the player they wish to sign from a list of local and international talent, each with a set base price. This process is based on a significant amount of analysis of a player's previous success, the playing conditions of the grounds where the player will do well or not, and other different considerations such as income, cultures, motivations, and so on. So, in order to choose a player, management must think and study extensively, and they must also consider their budget during the bidding process. We consider this procedure to be a challenge that can be solved using different computer science methodologies. There have been a few studies conducted before us to solve this sort of dilemma. For the first time, we combine the structure of machine learning processes and multi-objective optimization techniques with a slew of related techniques to solve the problem of automated squad creation. We use the Indian Premier League tournament as a model for the formulation of the problem and attempt to solve it using the methodologies we suggest. Each franchise in the Indian Premier League is given a set sum of credit, which they use to create a team with a certain number of local and foreign players. Each squad consists of 25 players, with a limit of 8 non-Indians. The administration wants to choose their pre-selected players at the auction based on the coaches' and staff's preparation. The approach we would suggest consists of forecasting the outcome of the players using machine learning algorithms and the previous playing data of the players in the tournament. We use a multi-objective optimization strategy to shape optimized squads after the prediction process for each player for the entire tournament. The following problem is a multi-objective optimization problem in which we take the expected batting performance, bowling performance, and base price of the players as objectives to create a squad of 23 participants with the constraint of choosing eight international players for each squad. The key goal of our thesis-work is to find the best squad for a tournament in which the performance of the players is expected for the tournament ahead of time and the squad is formatted based on the performance and expense of the players. For a competition, the squad consists of twenty-three to twenty-five players, depending on the team's retained player numbers. Our aim is to form a squad of twenty-three participants, with no more than eight international players. Here is the chart of the number of players of different roles:

Table 1.1: Number of Players of Different Roles

Playing Role	Number in Squad
Top Order Batter	4
Middle Order Batter	3
All-rounder	5
Wicket-Keeper	3
Bowler	8

Twenty-three players are chosen from a pool of 180 to form a squad. Since a player's success is determined by his or her continuity, form, and performance against a specific opponent and in a specific location. Our aim is to study all facets in order to forecast results. Following estimation, the base price levied on a player is enabled with the expected batting and bowling results, which we consider to be several objectives for forming an optimum team. We shape optimum squads of various prices while keeping the playing quota limit in mind. The task of team management and coaching personnel comes into play here to select an appropriate squad for them from the list of qualifying squads that may compete in the tournament in order to achieve a successful result in advance.

1.3 Motivation

The International Cricket Council governs the game of cricket, and organizes world cups for both twenty-overs and fifty-overs cricket every four years. For a long time, the fifty-over world cup tournament, in which many countries compete, has been the only global cricketing rivalry. The value of cricket has changed dramatically with the introduction of a franchise-based shorter version of the game. Cricket is no longer just a venue for entertainment; the monetization of cricket events requires sponsors to spend heavily, making it a great source of commercial paradise. From the article (Barma & Sultana, 2020), Cricket's sponsorship presence may be objectively measured. Benson and Hedges, Prudential, and Wills became correlated with the international cricket world cup in the 1990s. The commitment values of the sponsors grow and become massive over time. Cricketing tournaments are now a source of financial jewels. Various franchise-based tournaments are held in the modern world. The Indian Premier League is one of the most influential, and in terms of competing teams, it ranks among the best. When the tournament auctions take place, each team is given a sum of eighty-five crores of Indian rupees, which they must use wisely to acquire twenty-three to twenty-five players. Bidding wars erupt from time to time, when many franchises need a single player to play with them. Since the monetization of cricket is enormous, this area has fewer studies in terms of having teams optimally under a limited budget. Our mission, and hence our job, is entirely focused on investigating this area in order to provide an integrated and structured analysis in which team owners can sit and analyze the predicted performance for the upcoming event and pick what best suits them from a list of tailored squads. We have many solutions in mind for this type of challenge, including machine learning and deep learning. We use machine learning approaches because the data does not require as much as a deep learning process. As we focus on players who have previously played in the IPL, we must study different facets of cricketing sectors in order to accurately forecast a player's performance. The reason for working on this is to correctly predict results, which requires analyzing different aspects of the game that affect

the players.. Prediction of results is not our only goal; in order to participate in the competition, a squad must be formed, and a squad of eleven members is insufficient. For a long tournament with a limited budget, the whole squad must be created, which is where multi-objective optimization techniques come in to view our problems and solve the problem by resulting in any optimal squads that could participate in the game if selected.

Chapter 2

State of the Art

We have analyzed some papers which are related to our topic. The short description of our reviewed papers are given below:

D. Thenmozhi, P. Mirunalini, S. M. Jaisakthi et al [1] proposed a research to predict the outcome of a current cricket match, taking on account over-by over, using the information and data provided from each over. In every over, the statistics provided is the amount of runs scored, the total of fours and sixes hit, the numbers of wickets taken, and the number of extras. From the data collected, different models were needed for this study. Match results depend on several stages of the game. 2-Overs, 5-Overs, 8-Overs, 12-Overs, 16-Overs, and 20-Overs are the various stages of the match.. These models are used to forecast the outcome of a current match. The author made predictions for all IPL matches. This is accomplished by conducting match prediction using machine learning techniques. The Random Forest machine learning algorithm was the best method for anticipating match winners since it had the highest accuracy across all teams. The top models of this technique have an overall accuracy of 75.5 percent.

Shubhra Singh , Parmeet Kaur et al [2] have developed a prediction tool and data visualization that makes use of an open-source distributed non-relational database called HBase, which is distributed and non-relational to keep records of data about IPL (Indian Premier League) cricket matches. Later this information is utilized to create a visual representation of the previous performance of players. The data is also utilized to evaluate the result of a match via the application of different machine learning techniques. The suggested technology may be useful for teamwork during player auctions, particularly in the process of deciding the best team. For data storage the program utilizes open source , HBase, a distributed and non-relational database. HBase is rapidly being used to store tables, which contains huge amount of data .It enables automated and customizable table shredding for system scalability. The following are the features of the given work:

- To compile statistical data about players based on their many qualities.
- To forecast a team's success based on every player data.
- To accurately predict the result of Indian Premier League matches.

The paper discusses the issue of forecasting the result of an IPL cricket match as well as a player profile method that may be very beneficial for team captains on bidding day. The trials made use of the data from 644 matches. Numerous factors, including fate and player skill, are used to determine the outcome of a match. The key feature of the suggested method is that it approaches the issue as a vibrant one, and that it makes use of a relevant non-relational data, HBase, to ensure the stability of the software program.

Saptarshi Banerjee , Arnabi Mitra , Debayan Ganguly , Ritajit Majumdar and Kingshuk Chatterjee et al [3] proposed a paper that (i) for batsman and bowlers, chose terms of functional but also established various derived features based on evolutionary features, (ii) Heuristics have been developed for categorizing batters into three groups: opener, middle order batsman, and finisher and (iii) developed algorithms for determining the relative rating of batsmen and bowlers, analyzing individual player's individual performances as well as his or her previous experience , (iv) proposed 2 greedy methods for team selection in which the team's total credit point and the number of individuals for each group are constant. The authors examined conventional and derived characteristics in this study and evaluated them. While the suggested ranking system and algorithm may be used to pick the best potential squad, it can also be used to find the best alternative player in the event that each of the targeted players is unavailable. The forthcoming prospect of this paper is to amalgamate two-higher level clusters of allrounder in batting and balling. The squad selection process may also involve some more budget in which all-rounders are prioritized above batsmen and bowlers. It is possible to investigate the trade-off between including an allrounder in the squad and including a batsmen or a bowler who has a higher credit point.

Parag Shah , Mitesh Shah et al [4] proposed a paper in which they discovered a new statistical measure FORM that evaluates a player's form. They used a method called exponential decaying averages (EDMA). All scoring is taken into account in this measurement, however as you move back in history, every scoring is lowered by a specific amount. This implies that the most current value gets prioritized, while previous values are offered less weight. FORM is calculated using a basic logic consisting of short term EDMA and long term EDMA. The performance of the team relies on the overall form of the players and this is very important . It may be used to calculate the overall score of the whole team. In the present situation, runs scored in innings in which a batsman is not out are included in the batsman's total runs scored. However, those innings are excluded in the overall number of innings when computing the average. This over rates the batsman's ability to perform on the field. Although

not out runs are counted as out runs, the batsman's performance is undervalued. To address this flaw in the current average, the writers proposed a new rationale for the average. Here two rules are applied:

1. When not out runs are greater than average then calculating his total runs, take into account not out runs.
2. When not out runs are less than average it will take the average point from the player's entire runs. Overestimation and underestimation will be solved from this method.

Sricharan Shah, Partha Jyoti Hazarika and Jiten Hazarika et al [4], have proposed a paper where the Factor Analysis algorithm was used to evaluate the performance of cricket players. In this paper the datasets of 95 batsmen and 95 bowlers ;85 batsmen and 85 bowlers from the IPL9, 2016 (20 overs) and the ICC World Cup, 2015 (50 overs) were taken into consideration. It is shown by the results of this research that batting skill outweighs bowling skills, which is consistent with the observation of a journal written on the very same game. The five most important batting statistics (average batting performance (ABP), highest individual score (HS), no of fours (4s), and number of sixes (6s), strike rate (SR)) and three bowling statistics, including bowling average, bowler's economy rate, and bowling strike rate, were taken into consideration to analyze players' batting and bowling performances in both the IPL9, 2016 and the World Cup, 2015. According to this article, it is discovered that for both twenty-overs and fifty-overs matches, the 5 dimensions were classified as factor 1 (batting), whereas the 3 dimensions were classified as factor 2 (i.e., bowling). The amount of variation described by factor 1 (batting) is significantly greater than the amount of the variance described by factor 2. (bowling). The conclusion may be reached as a consequence that this batting skill is more important than bowling ability. As a result, the bowler's performance has become one of the key variables that may influence the outcome of a match.

Faez Ahmed, Abhilash Jindal, and Kalyanmoy Deb et al [5] proposed a multi-objective method that uses the NSGA-II algorithm to improve a team's bowling ,batting efficiency and identify players in a team. A decision-making method for ultimate selection of the squad is also suggested, which makes use of the findings obtained from the trade-off front. A research paper using a group of players who were auctioned off in the fourth edition of the Indian Premier League has been conducted, with the current T-twenty statistical information of the players serving as performance measure for each player. In this study, the authors have shown applicability of multi-objective optimization technique to biased team building issues from a group of players utilizing accessible data. So they have taken cricket for this confirmation. Every player costs a unique integer number and is uniquely identified integer value. A team is expressed as a chromosome composed of eleven real variables, each of which corresponds to a player's badge. The overall bowling and batting performance

of every other team member is assessed as fitness. In addition, the total combined budget limitation is indicated as a barrier for the overall team's cost. Another restriction is that no 2 players on the squad may be the same, thus there can't be any duplication of members within the same team. The IPL's unique conditions are also kept in mind, such as the fact that no more than four foreign players are allowed in the team. An examination of the acquired trade-off resolution has shown that it results in a chosen team that was discovered to have greater bowling and batting average than the team that wins of the most recent Indian Premier League event.

Prof. A. R. Babhulgaonkar, Ganesh Karale, Anup Kushwah, Ashutosh Kshirsagar et al [6] proposed a paper where the performance of a cricketer was predicted using a multivariate linear regression model. Multiple linear regression analysis, which is reliant on many independent factors. The performance of a player is anticipated depending on a variety of parameters including the venue, pitch condition, strike rate, and so on. Predicting the result of the batsman and the economy of the bowler for a certain match is computed on the basis of this information. Individuals' scores and economy are taken into consideration while selecting the eleven players who will participate in a match depending on their performance. Multiple linear regression with gradient descent method is used to determine the performance of cricket players in this research study, to improve cricket player performance. Any cricket board may use this prediction to pick the top eleven players to create an optimal squad.

Kalpdrum Passi, Niravkumar Pandey et al [7] proposed a paper in which the author tries to estimate the performance of players, such as how many runs each batsman will score. Multiple classifier issues are being focused: the number of runs and the number of wickets are both being categorized in distinct regions for both issues. The writers of this article use trained machine learning methods to evaluate players' performance in One Day International (ODI) matches by examining their traits and statistics. They do this by predicting the performance of batsmen and bowlers individually, including how many wickets a bowler and how many runs a batsman will get in a given match. Because player statistics such as strike rate, average and so on are not accessible for each game, the statisticians computed these features from every inning lists applying aggregate functions and statistical equations, which they then used to rank the players. Random Forest was the most efficient classification algorithm for both samples, estimating runs scored by a batsman with an accuracy of 90.74 percent and wickets gained by a bowler with a precision of 92.25 percent.

Deep Prakash, Deep Prakash, Vasantha Lakshmi et al [8] presents a Deep Mayo Predictor model that is consisting of three sections, one of which is reliant on a variety of factors acquired through a thorough examination of T20 cricket. The models are built utilizing data analytics techniques from the area of machine learning. This article describes a rating system for the batsmen and bowlers participating in the IPL IX. This rating serves as the foundation

for all three models that the authors utilized in their research. In the Prediction phase, the training data is assembled by eliminating upto IPL 8 players' info. The five characteristics that contribute to batting and bowling performances are used in this situation. Batting and bowling rankings of these players are computed using all these characteristics and the weights generated for these parameters using the Random Forest Algorithm. This model is identical to the one before it. The change is that, in addition to including the players' current form, the training set is modified with every new match played in 9 th season , and the data gets into the training set for upcoming matches. The training data for Prediction Model 3 is created by collecting all of the players' data up to IPL 8. The characteristics taken into account are each one of the five previously established features for batting and bowling performances. These characteristic values are calculated from the average of each of the 11 players in the 9 th season . As a result, each team receives a total of ten distinct characteristics. This is done for the competitor teams, and the disparities between the two teams are utilized as the training vectors for the different teams. The outcomes of the matches make up the target vector, which is 1 when the 1st team wins and 0 if the other team wins. This training data is used to train a Support Vector Machine model, which is then used to determine IPL 9 regarding the data from the training data. In the article, the findings acquired from PM1, PM2, and PM3 separately, as well as the results gained from the composite Mayo approach, which simply reflects the majority of votes between the 3 other models, are presented. The proposed predictor correctly guessed the result of 39 out of 56 matches which is a moderate prediction accuracy.

Vidit Kanungo, Tulasi B et el [9] proposed a paper about which gives the decision makers a clear idea for choosing players for their team. This paper analysed toss related data and goes into data visualization breadth. This article discusses data visualization methods, as well as Toss-related analysis like plotting. This paper investigates data visualization methods, as well as Toss-related evaluation like plotting, for the data that has been gathered. When compared to numbers and words, data presented in a visual format is more effortlessly understood. A visual presentation of information is achieved by representing data in the form of charts and graphs. Data visualization and predictive analysis are two key aspects of information visualization and predictive analysis. NumPy is used to perform numerical computations on the datasets that have been provided. The primary visualization for players is Matplotlib. When it comes to Toss-related data, along with team and player insights, the Seaborn program is utilized as the basic visualization. A number of new functionalities are being added. The authors pay extra attention to player performance, mainly to batsmen, and also mark the analysis. This paper analyses data visualization methods, as well as Toss-related analysis such as plotting, for the data that has been gathered. This paper provides selectors with a solid decision of batsmen from the Mumbai Indians and Kings XI Punjab, since both teams performed well under pressure throughout all of the seasons from 2008 to

2018. Team Management can pick the proper players and teams for the bidding by taking into account all of this visualization and toss-related research. Within a certain budget, a decent and powerful cricket team may be created that has the best chance of winning.

Md. Jakir Hossain, Md. Abul Kashem, Md. Saiful Islam and Marium-E-Jannat et al [10] have proposed a paper utilizing statistical data and genetic algorithms to choose an optimum cricket team. The performance of all Bangladeshi players in international matches and national league over the last two years is used to predict individual player's performance and identify the top thirty players based on statistics. The 30 players are then subjected to a genetic algorithm in predicting the final Bangladeshi national cricket team of 14 players. This suggested model's projected squad is only valid for one-day international (ODI) matches. The authors of this paper presented a methodology for cricket squad selection that takes into account both genetic method and statistical analysis. The top 30 Bangladeshi players are chosen based on statistical data. A specific set of batsmen, bowlers, allrounders, and wicket keepers are included in the team of 30 players. Traditional computer methods, on the other hand, make it impossible to evaluate all of these options and choose the best team. As a result, a genetic algorithm is used to pick the final Bangladesh national cricket team from these 30 players. The initial version for genetic algorithms is picked from a statistically determined group of 30 participants. In a genetic algorithm, each potential solution is referred to with a chromosome, and each player in a chromosome is referred to as a gene. In a genetic algorithm, the steps of parent selection, mutation, crossover and survivor selection are carried out in order to choose the final squad fitness evaluation values and fitness functions of each player. The writers computed each squad's fitness value and ran GA on our system multiple times. The writers then choose the team with the highest fittest number as their ultimate recommended team.

Sandesh Bananki Jayantha, Akas Anthonyb, Gududuru Abhilashab, Noorni Shaikb and Gowri Srinivasaa et al [11] have proposed a paper where SVM model with linear and non-linear poly and RBF kernels is used in combination with a supervised learning technique to predict the result of a match against a specific side by categorizing the players at various levels in the sequence of play including both teams. When various groups of players at a same rank are compared, the order of groups that leads to winning percentage is determined. Developing a system that suggests players for particular roles in teams based on previous performances is being considered, according to the suggested research. This was accomplished by the author by grouping all of the players using k-means clustering and then identifying the five closest players using the k nearest neighbor (KNN) classifier to identify the comparable players. It determines a player's rating index by analyzing the game and extracting data from the players' performance in a specific event. By collecting data about the cricket game and players from different sources and incorporating them into a framework, the author has developed a framework that predicts the result of matches, conducts team

analysis, suggests player roles, and makes team recommendations. The suggested model has the following characteristics: 1. Extract and store unstructured data about matches and players from sports websites. 2. Statistics are used to evaluate player performance metrics in order to rank the players. 3. Using historical data from previous matches, create a method to anticipate the result of the match based on the players on both sides. 4. Presents a study of the chosen team's squad structure, which is important in achieving victory. 5. Finding a collection of comparable players in the database and recommending a preferable position for a particular player is the goal of this function. Consequently, their data set is not linearly separable, according to the findings. As a result, the author intends to utilize the SVM combined with the RBF technique to forecast the outcome of a match. It was decided to utilize player performance metrics to cluster all of the players using k-means clustering, and comparable players were identified using k-nearest neighbor classifiers in this player recommendation system.

Faez Ahmed, Kalyanmoy Deb and Abhilash Jindal et al [12] used the NSGA-II algorithm to propose an optimized team of 11 players who were used as variables. They used a multi-objective approach. This paper optimizes the entire bowling and batting strength of players. The decision making process also includes other cricketing criteria like fielding performance. The final team was formed using a multi-criteria decision making approach. For this they used the data from the obtained trade-off front. Their test case contains - Indian Premier League (IPL) and Twenty20 cricket game. They used a set of players who were auctioned in IPL (4-th edition) and considered their current statistics. The league franchises can use the proposed technique to form a team according to their preference and budget. The paper also shows how helpful this analysis is in selecting players from an auction specially in a dynamic environment. According to the authors the bowling performance of a team is not affected by wicket-keepers. They started with 129 players. Their procedure introduced a team which is high in terms of performance. The proposed team also cost less compared to the winning team of IPL (4th edition). After a satisfactory result they also added other criteria to get an even better team.

Madan Gopal Jhavar, Vikram Pudi et al [13] used team composition perspective to measure the result of a cricket match. They worked on One Day International (ODI) cricket matches. They used a supervised learning approach. They came up with the conclusion that, in predicting the winner of a match, a very crucial feature is the relative strength in between the competing teams. It is really important to model an individual's performance (both batting and bowling) to model an entire team's strength. To model a player, the authors used both their recent performance and their career statistics. The authors also used some player independent features. K-Nearest Neighbor (kNN) algorithm gives more accurate results than any other algorithms according to their studies. The authors estimated the bowling and batting performance of twenty two players who played the match to measure

the result of ODI cricket matches. For this purpose they used players' career statistics. This paper also used two other base features. They are the venue of the match and toss decision. This research work introduced some novel approaches to measure the performance of the players and players' contribution. Their approach was dynamic in solving the prediction problem which is the novelty of their contribution.

S Anjali, V Aswini, M Abirami et al [14] proposed a paper on twitter accounts having a large amount of data particularly they worked on the tweets related to cricket. They used that data to analyze the winning performance of a team. The authors analyzed the fetched data by using reducing algorithms and map - concepts of big data. The tweets of people around the globe, consists of their opinions, emotions and views. This research includes the tweets of people related to cricket as their data. The data is huge containing positive - negative comments. There might also be some unwanted data. Flume, one of the big data tools was used to retrieve the data in this paper. Hdfs server was used to store the data in the form of data nodes after extracting it. Since this paper used live data, it opened up the way towards getting more accurate results for research. This procedure can be applied to all sports like baseball, football etc. Incorporating a hybrid programme can also be taken into consideration.

Maral Haghighat, Hamid Rastegari and Nasim Nourafza et al [15] reviewed the data mining techniques in their proposed paper for predicting the result in sports. The present study reviewed researches on data mining techniques which tries to predict sport results and estimates characteristics of each system. The 2nd part of the paper includes the data collection, 3rd part contains the feature selection techniques, 4th part includes the classification methods and 5th part includes the result and advantages and disadvantages of the chosen systems. As the popularity of sports is increasing rapidly in the current world, many organizations expend huge funds to have a better result in matches. So it is very crucial for different sports organizations to predict the result of games. For this purpose, data mining is a great tool. In recent years a variety of data mining methods have been used for predicting the match results such as - decision trees, fuzzy methods, Bayesian method, SVM, ANN, logistic regression. Two challenges came up while evaluating the result. 1st for reliable predictions further research is needed. 2nd sports website lacks in providing accurate statistics. This leads the researchers towards an unclear prediction. This paper tries to solve these challenges. For example they proposed that the use of data mining techniques and machine learning can improve the accuracy of prediction. Hybrid algorithms can also be used for prediction to get a good result. The paper also suggests collecting a dataset by taking help of a group of experts who belong to the sports world.

Shanthi Muthuswamy and Sarah S. Lam et al [16] explored an approach which is based on neural networks. They used (RBFN) radial basis function and (BPN) backpropagation network and predicted the Indian cricket team bowlers performance. Here runs gained a more

effective prediction model than wickets. Therefore, for the wickets scenario a classification method was approached which used the above mentioned two network paradigms. For the classification and prediction purposes they used two approaches - RBFN and BPN. The authors tried to measure the bowler's performance in the cricket team of India who played against 7 different international teams. (BPN) backpropagation network and (RBFN) radial basis function network was used to calculate the number of each bowler's run. The above described network models were used to classify the number of wickets which were taken into 2 groups. Here they tried to find the most suitable network structure and for that they experimented with a single output model as well as a multiple output scenario. The given factors are the number of overs to be bowled, the bowler ID and the team ID. Now this approach would measure the runs that the bowlers would give. The approach will also predict the number of wickets which would be taken in one of the following 2 categories - 1st category is between zero to two wickets and the 2nd category is three or more wickets. This study has a limitation that it is limited to the 8 Indian bowlers who played since 2000 in the ODI matches against 7 different countries.

1. To compile statistical data about players based on their many qualities.
2. To forecast a team's success based on every player data.
3. To accurately predict the result of Indian Premier League matches.

Table 2.1: Summary of literature review

Ref.No.	Method	Dataset	Accuracy & limitations
[1]	Gaussian Naive Bayes, Support Vector Machine, KNearest Neighbor and Random Forest.	Data was extracted from the website, www.cricsheet.org 1 which consists of ball-by-ball information about the IPL matches.	The accuracies obtained are 75%, 80%, 55%, 75%, 80%, 80%, 75% and 84% for the teams CSK,RR, DD, RCB, MI, SRH, KXIP and KKR respectively.
[2]	KNN(k-Nearest Neighbors) algorithm with k=4 has been used for predicting the winner of a match between 2 IPL teams. Further, KNN was compared with other machine learning algorithms; namely, decision tree, logistic regression, random forest and Support Vector Machine.	Collection of data was performed using the BeautifulSoup[8]library of the Python programming language from the website www.cricbuzz.com	KNN has the highest accuracy among all the algorithms used.Supervised KNN with k=4 gives accuracy up to 71%.
[3]	Greedy algorithm	Database contains data of all the players and their performance in the last eleven seasons of IPL.	The ranking obtained by heuristic scheme is in acceptance with the known player rankings in IPL.The greedy algorithm for team selection has a shortcoming that it may select some high ranking players with some very low ranking ones.
[4]	Exponentially decaying average (EDMA)	Dataset contains data of players in last 50 innings from www.cricinfo.com website	When short term EDMA is more than long term EDMA it shows that batsman is scoring well in recent matches.So,the FORM will be more than 100 and paper says that the batsman is in good form and when it is less than 100, he is not in good form.

Ref.No.	Method	Dataset	Accuracy & limitations
[17]	factor analysis	data has been collected from ICC WorldCup tournament, 2015 and IPL session 9, 2016 which are freely available in the website: www.icc-cricket.com and www.espncriinfo.com .	Batting capability dominates over bowling capability. So, in 50-overs matches, the performance of the bowler is one of the significant factors which may change the scenario of the matches.
[5]	Non-dominated sorting genetic algorithm (NSGA II)	Data used for this work has a pool of 129 players from IPL 4th edition.	An analysis of the obtained trade-off solution has been shown to result in a preferred team that has been found to have better batting and bowling averages than the winning team of the last IPL tournament.
[6]	Multiple linear regression with gradient descent algorithm.	Historical data of Indian cricket players consisting of bowler, batsman and all-rounder was collected from various cricket sites and stored into database.	Using historical data of cricket player it is possible to successfully predict the performance of the cricket player
[7]	Naïve bayes, random forest, multiclass SVM and decision tree classifiers	Data was obtained from www.cricinfo.com using scraping tools, parsehub and import.io	Random Forest predicts runs with the highest accuracy of 90.74% and predicts wickets with highest accuracy of 92.25%.
[8]	Composite Mayo model, Support Vector Machine (SVM)	Data was obtained from www.cricinfo.com using scraping tools, parsehub and import.io	Mayo model is able to predict the outcomes with high accuracy getting 39 out of 56 matches.

Ref.No.	Method	Dataset	Accuracy & limitations
[9]	Data visualization techniques	Data consists of the ball by ball details for a total of 696 matches from 2008-2018 .	A good and strong cricket team can be formed within a given budget, which will have the highest chance of winning.
[10]	Mapping on statistical data.GENETIC ALGORITHM has been used.	Statistical data is used to predict the performance of each player based on their last 2 years national and international Performances. Data is scraped from espn-cricinfo.com .	Authors have calculated fitness value for each squad and run GA several times on the system. Then chose the best fittest valued squad as our final predicted squad.
[11]	SVM, k nearest neighbor (KNN) classifier	SThe 2011 cricket World Cup (CWC) and the scorecards provided in Howstat.com has been considered.	SVM with RFB kernel yields the accuracy of 75, precision of 83.5 and recall rate of 62.5. So the use of SVM with the RBF kernel for game outcome prediction was recommended.
[12]	NSGA-II,multi-criteria decision making (MCDM) methods,multi-objective genetic algorithm and multiple criteria decision making aids.	Data is taken from the public domain sources, compiled and stored in a website http://www.iitk.ac.in/kangal/cricket .	In a comparison to the winning team of the IPL 4-th edition (played in April-May, 2011), the teams obtained by this paper are theoretically better in both bowling and batting performances and importantly also cost less to hire the whole team.

Ref.No.	Method	Dataset	Accuracy & limitations
[13]	Nearest Neighbor, supervised learning algorithms	Dataset includes all the matches played between 2010 and 2014 which have been scrapped from the cricinfo website.	The paper addresses the problem of predicting the outcome of an ODI cricket match using the statistics of 366 matches. The novelty of the approach lies in addressing the problem as a dynamic one, and using the participating players as the key feature in predicting the winner of the match.
[14]	Reactive – business intelligence, Reactive – big data BI, Proactive – big analytics,	The source of data for this project is twitter. Then these data are retrieved with the help of big data tools like flume	The live data consideration for analysis enables the authors to come out with more accurate results.
[15]	Data mining techniques - ANN, decision trees, Bayesian method, logistic regression, SVM, and fuzzy methods.	Data was collected from valid sports websites.	This paper evaluated available literature and detected two major challenges. First, low prediction accuracy. Second, lack of a general and comprehensive set of Statistics. The paper also suggest a number of solutions to eliminate such challenges.
[16]	Neural network approach using back-propagation network (BPN) and radial basis function network (RBFN)	Data was extracted from various cricket websites.	An accuracy of 87.10% with BPN and 91.43% with RBFN was achieved.

Chapter 3

Background Study

3.1 Naive Bayes Classifier

The Naive Bayes classifier [18] is a part of the probabilistic classifier which is based on Bayes theorem. In this classifier, the features are assumed to be independent of each other. Naive Bayes classifiers can be scaled easily. Maximum-likelihood training can be accomplished by calculating a confined expression, which takes just linear time, instead of by expensive iterative approximation, which is the method of choice for many other types of classification algorithms. Simple Bayes models and independence Bayes models are two names for naïve Bayes models that are commonly used in the statistical and computer science field. Naive Bayes is a basic methodology for building classifiers: models that represent class instances, which are expressed by vectors of feature values, and where the class labels are selected from a finite set of possible choices. There is no one technique to train such classifiers, but rather a variety of algorithms that are all based on the same underlying principle: most naive Bayes classifiers assume that ; given the class parameter ; the presence of features are not dependent on each other. Using the example of an apple, if this is reddish, round, and around 15 cm, it is regarded to be an apple. Naive bayes considers each of these characteristics, contributing independently to likelihood that such a fruit is an apple, irrespective of whether the colouring has any correlation between them, roundness, and diameter characteristics or whether there is any correlation between the colouring, roundness, and diameter characteristics. In a supervised learning scenario, naive Bayes classifiers can be learned relatively quickly for certain types of probability models. In many practical uses, parameterization in naive Bayes models is performed using the approach by maximum likelihood. A naive design and oversimplified assumptions have led to the success of naive Bayes classifiers in a variety of complex real-world settings, despite their shortcomings. Naive Bayes has the advantage of requiring a limited range of training data for estimating the parameters required for classification. The Bayes Theorem states that if B is a data item and A is a class label,

then If we assume that B belongs in class A, therefore

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (3.1)$$

3.2 Support Vector Machine Classifier

Support Vector Machine (SVM) [18], a supervised machine learning algorithm which can perform classification, regression, outlier detection. It analyzes data for regression and classification problems. It was introduced by Isabelle Guyon, Vladimir Vapnik and Bernhard Boser. SVM depends on statistical learning models. It is a robust prediction model. It performs accurately than other models and avoids overfitting. SVM can predict classification problem and numerical problem. The actual data is converted into an accurate format. If nonlinear mapping is used, a linear more accurate hyperplane is searched by SVM. It adds another dimension which separates the instances of each class. With the help of an accurate mapping and a high enough dimension, instances from two classes can easily be divided. We can write a separating hyperplane as:

$$P \cdot L + q = 0 \quad (3.2)$$

Here P is a weight vector, $P = p_1, p_2, p_3, \dots, p_n$, the attribute number is represented by n . Here q , a scalar, is often referred to as a bias. Let us input two attributes $M1$ and $M2$, then the training tuples are 2-D,

($L = (l_1, l_2)$), where l_1 and l_2 are the values of attributes $M1$ and $M2$, respectively. So, any points which are above the separating hyperplane belong to Class $M1$:

$$P \cdot L + q > 0 \quad (3.3)$$

And any points which are below the separating hyperplane belong to Class $M2$:

$$P \cdot L + q < 0 \quad (3.4)$$

When enough information about the data is unavailable, SVM's perform really well. Performs well with semi structured, unstructured data. For example trees, text and images. The actual strength of SVM is the kernel trick.

3.3 Decision Tree Classifier

This classifier generates trees for training tuples that are class labeled. A decision tree [18] is a tree model like a flowchart. The inside nodes of the tree indicate test attribute, branch indicates test's result. Every leaf node has a class label. The tree head is referred to as the root node. To identify an instance A, the characteristics of the instance are differentiated with the decision tree, which is started at the initial node and ended at the bottom nodes. The ID3 decision tree approach was first discussed by Ross Quinlan. Next, C4.5 which is a successor to the prior ID3 was introduced as there was few deficiency of the proposed ID3 like over-fitting problem. C4.5 can deal with missing values, costs of training data, control discrete and continuous attributes. A decision tree induction contains all of the training instances at its root node. Following that, the tuples are recursively partitioned depending on the properties selected. Heuristic procedure is used in deciding the splitting indicator. No further splitting is needed if all the training instances are owned by the same class. Expected data for classifying a tuple in the training set D

$$Info(D) = - \sum_{i=1}^m P_i \log_2(p_i) \quad (3.5)$$

After the dividing, information is required

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (3.6)$$

Next, information gain

$$Gain(A) = Info(D) - Info_A(D) \quad (3.7)$$

A split info is used to calculate the gain ratio,

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right) \quad (3.8)$$

Next,

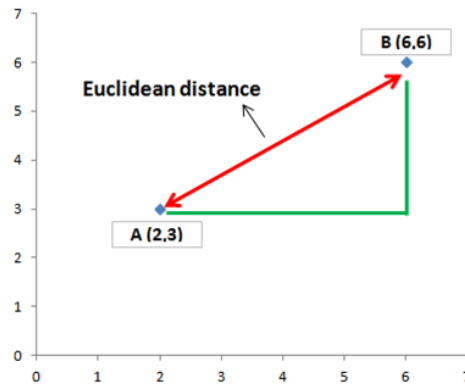
$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \quad (3.9)$$

The attribute which gives the maximum gain ratio is the chosen dividing attribute.

3.4 K-Nearest Neighbors Classifier

K-nearest neighbors (kNN) [19], a supervised machine learning algorithm that solves both classification and regression tasks. This algorithm is similar to our real life. A particular

point's value is determined by other points that surround it. We can explain it in a way that when we have one best friend, spend time with him/her, we will share the same interests. It can be viewed as KNN where $k = 1$. On the other hand if we spend time with a group of 4 friends, each of them will have an effect on our interest. It can be viewed as KNN with $k=4$. A majority voting principle is used by KNN to determine the class of a point. If $k = 4$, the classes of 4 nearest points are checked. The majority class decides the predicted class. To determine the close data points euclidean distance (minkowski distance with $p=2$) is used. The following figure shows the calculation of distance between 2 points in a two dimensional space.



$$\text{Euclidean distance } (a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Figure 3.1: K-Nearest algorithm

Here, euclidean distance is the square root of $(16 + 9) = 5$. It is quite simple for 2 points in 2D space. Typically problems contain many samples containing many features.

3.5 Random Forest Classifier

Random forests [18], an ensemble learning method used for regression, classification and other tasks. Random Forest constructs a multitude of decision trees at training time. The output is determined by selecting the class that most of the tree chooses. Random forest can be seen as a collection of decision trees where every tree depends on a random vector. Random forest generates various results.

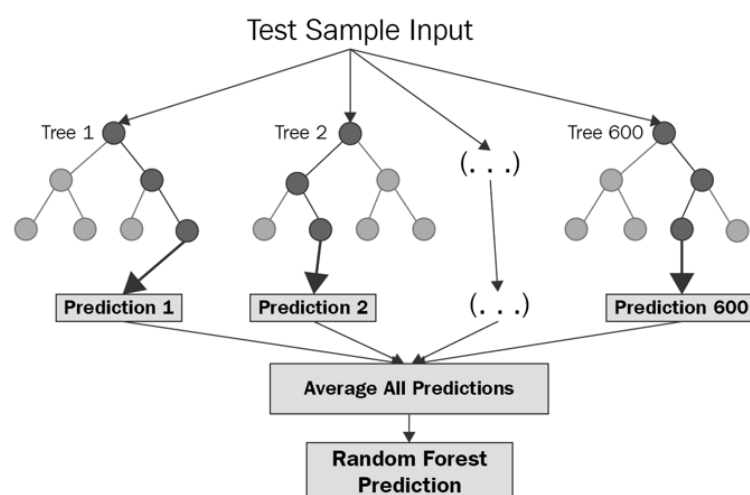


Figure 3.2: Random Forest algorithm

Here the forest is formed by decision trees. Random attributes are picked at random times to create a decision tree. Each node determines the splitting. The initial random forest method

was introduced by Tim Kam Ho. After that the extended algorithm was described by Breiman Leo. A dataset P of p instances is used for constructing the decision trees for this algorithm. Few data are replaced from dataset P for building a decision tree classifier. Split candidates are chosen randomly from the existing attributes. The tree is built using the CART approach (Classification and regression). CART is a non-parametric technique. The algorithm terminates when there is no more possible gain or some predefined terminating conditions are satisfied.

3.6 NSGA-II Algorithm :

It is theoretically possible that the presence of several objectives in a problem leads to the formation of a set of optimum solutions (often referred to as Pareto-optimal solutions), rather than the formation of a single optimum solution. If any more information is absent, it is impossible to determine which of these Pareto-optimal solutions is superior to the other. This necessitates the user's finding as many of the Pareto-optimal solutions as they possibly can. When dealing with multi-objective optimization problems, traditional methods (including multi-criterion decision-making methods) recommend that the multi-objective optimization problem be reduced to a single-objective optimization process by emphasizing only one particular Pareto-optimal solution at the same time. When a method such as this is used to find many solutions, it must be applied numerous times, with the aim of getting a different solution with each simulation run. From [20] it is clear that in the field of multi-objective optimization, NSGA-II is one of the most widely used algorithms because of its three unique characteristics: its fast non-dominated sorting approach, its fast populated distance estimation procedure, and its simple crowded comparison operator. It is one of the most popular algorithms because of its three special characteristics. NSGA-II can be broken down into the following steps which are mentioned in [20], which are approximately explained below.

- Step 1: Initialization of the population
 - Initialize the population using the problem range and constraint as a starting point.
- Step 2: Sort by non-dominant factor
 - This is a sorting method that is based on non-dominance criteria of the population that has been first created.
- Step 3: Crowding distance

- When the sorting is finished, the crowding distance value is assigned front-wise. Individuals in the population are chosen based on their rank and the distance between them.
- Step 4: Make a decision
 - It is decided who will be chosen by the use of a binary tournament selection with a crowded-comparison operator.
- Genetic Operators are the fifth step.
 - With simulated binary crossover and polynomial mutation, real programmed GA was achieved.
- Recombination and selection are the sixth and final steps.
 - The offspring population and the current generation population are combined, and the individuals of the following generation are set by selection. The new generation is filled by each front subsequently until the population size exceeds the current population size.

The procedure is found in [20] and the visual implementation is given below:

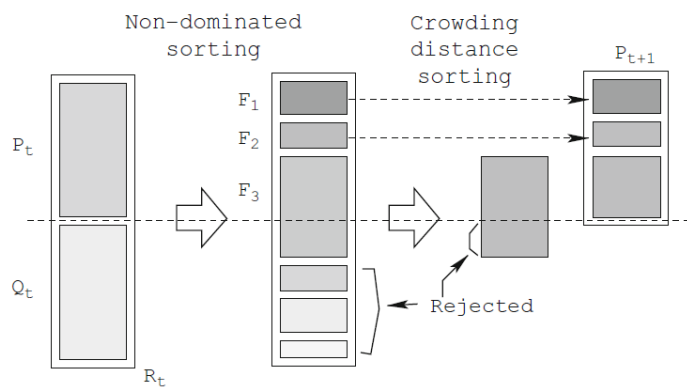


Figure 3.3: NSGA-II algorithm

3.7 SPEA2 Algorithm :

Strength Pareto Evolutionary Algorithm 2 is known as SPEA 2 which is an improved version of SPEA algorithm. The algorithm was proposed by Zitzler et al. (2001) [21]. SPEA2 makes use of a fine-grained fitness assignment mechanism, a density estimator, and an improved archive truncation method in order to maximize performance. The SPEA2 algorithm as in Zitzler et al. (2001) [21] is given below:

Algorithm 1: SPEA2 Algorithm

Input : N (population size)

N : archive size

T : maximum number of generations

Output: A (nondominated set)

1. Initialization: Generate an initial population P_0 and create the empty archive (external set) P_0 . Set $t = 0$.
 2. Fitness assignment: Calculate fitness values of individuals in P_t and P_t .
 3. Environmental selection: Copy all nondominated individuals in P_t and P_t to P_{t+1} . If size of P_{t+1} exceeds N then reduce P_{t+1} by means of the truncation operator, otherwise if size of P_{t+1} is less than N then fill P_{t+1} with dominated individuals in P_t and P_t .
 4. If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in P_{t+1} . Stop.
 5. Mating selection: Perform binary tournament selection with replacement on P_{t+1} in order to fill the mating pool.
 6. Variation: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to Step 2.
-

R. W. SAATY et al [22] explained the Analytic Hierarchy Process. Here AHP was explained as a procedure of measurement which has ratio scales. The author explained it with 2 examples. After that few central theoretical underpinnings are given. Next some proposals related to AHP are discussed. Special importance is given to the following topics -

- Relative and absolute measurement
- Departure from stability
- Measurement of departure from stability
- Rank preservation examples
- Relative measure in reversal

The proposed research was conducted in an expository manner to make it more accessible for people who are interested in this topic. The paper has three-fold purpose : 1st it introduces AHP with 2 hierarchically organized examples ; 2nd to explain the axioms and few underpinnings of the research ; 3rd the most relevant applications in the field of AHP. Mainly the paper describes the generalizations and uses of those measurements. Some of the fields regarding this work needs future studies to get a better vision on this topic

Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik et al [23] presented a training algorithm which aims to maximize the in-between decision boundary and training patterns. This approach can be applied to various range of classification functions like Radial Basis Functions, Perceptrons and Polynomials. The problem is complex and the effective parameters are automatically adjusted. The proposed methodology can be viewed as a linear combination of the supporting patterns. These subset of patterns are nearest to the decision boundary. In this research work a training algorithm is explained. This algorithm tunes the capability of the classification function. To do this the margin between class boundary and training examples are maximized [KM87], atypical examples are removed from the training dataset. The result of the classification function relies on the supporting patterns [Vap82]. The training instances are nearest to decision boundary and is a subset of the training data. It is showed that if the margin is maximized it minimizes the loss. This leads to many desirable results. The algorithms performance and efficiency is shown on handwritten digit recognition data. The paper did not use any task specified knowledge. Less than an hour training time was required in all experiments.

Yusliza Yusoff, Mohd Salihin Ngadiman, Azlan Mohd Zain et al [24] presented an overview on NSGA-II optimization techniques. It is a part of machining process parameters. Machining process parameters optimization includes several (MoGA) multi-objective optimization processes. Some of them are - Pareto-archived evolution strategy (PAES), micro genetic algorithm (Micro-GA), multi-objective genetic algorithm (MOGA), multi-objective genetic algorithm (MOGA), strength Pareto evolutionary algorithm (SPEA) etc. This research evaluates (NSGA-II) non dominated sorting genetic algorithm II applications. It is known as a MoGA technique which optimizes the process parameters in several machining operations. It is a well known algorithm and the approach is elite and fast sorting. Some parameters of process like rotational speed, cutting speed, feed rate etc are the appreciable contexts for optimizing the machining operations for maximizing or minimizing the performance. The single objective optimization technique gets dominated by other solutions while optimizing each objective. NSGA-II solves this problem. This is the most used algorithm in Optimizing the parameters of the machining process. This study reviews the uses of NSGA-II algorithm. This algorithm is a reliable and popular technique. Some techniques of machining performance predictions are Regression analysis, GP, fuzzy, ANN. NSGA-II can find sets of solutions depending on various combinations of appropriate variables.

H C S Rughooputh, R T F Ah King, K Deb et al [21] compared Strength Pareto Evolutionary Algorithm 2 (SPEA2) and The Fast and Elitist Nondominated Sorting Genetic Algorithm (NSGA-II). It was demonstrated on IEEE 30-bus system. The authors used normalized values. They used convergence metric as generational distance. They also used a diversity metric. An actual time of computation was also used. In this paper different indices of performance has been used to compare the above algorithms. A study on statistical comparison

was also carried using some tools of statistical comparison . This paper present the results for 2 classes.They are system having transmission losses and lossless system.Multi-objective optimization has two major goals.The 1st one is to find solutions which are close to the Pareto-optimal solutions. The 2nd one is to discover diverse solutions which are close to the Pareto-optimal solutions.Two performance metrics need to be used at least to compare the above two algorithms.In terms of both diversity and convergence, SPEA2 performed better rather than NSGA-II. But it has a computational time expense.As demonstrated, for this specific problem , the computational time difference is 1 magnitude higher for SPEA2.

Nyoman Gunantaraet el [25] contributed a paper which proposes an MOO settlement method. This method simplify the problem without requiring complex mathematical equations.Then the authors applied the settlement method to the ad hoc network.Though there are various kinds of optimization techniques, this paper explained only the MOO technique.The multi-objective optimization (MOO) tries to find the optimal solution which has various desired goals.In optimization, to simplify the problem , complicated equations are not required. That's why optimization uses multi-objective optimization.Vilfredo Pareto introduced multi-objective optimization.MOO has a vector of the objective function. Each of them is a function of the solution vector.MOO does not give a unique solution for all purposes. It rather gives several solutions.Scalarization and the Pareto method are the two classification of the solution of MOO problem.In case of separated performance indicators and desired solutions , the pareto technique is used. The solution is displayed as Pareto optimal front (POF).On the other hand, the scalarization method indicates performance , forming a scalar function. This paper concludes two reviews.

- Scalarization methods and Pareto methods are two MOO techniques which simplifies the problem by not requiring complex mathematical equations.The Pareto method describes POF in forms of non-dominated solutions and dominated solutions .Continuously updated algorithm gives the non-dominated solution.The scalarization method first determines the weights and turns multi-objective functions into a single solution.These weights are RS weights, equal weights and ROC weights.
- The pareto method gives a solution which indicates performance and forms MOO a compromise and separate solution.It is often viewed in the form of POF. On the other hand,the scalarization method produces a solution which is an indicator of performance. It is integrated in the fitness function.

Hardik H. Maheta, Vipul K. Dabhi el [26] proposed a paper on (SPEA2) , the renowned Multiobjective Evolutionary algorithm . They proposed enhancements to this algorithm. The proposed improvements enhance both diversity and convergence. To get a better convergence, in present study , Non-dominated solutions and Generational Crossover are used.

They are based on fitness of SPEA2. Solutions maintain their diversity by using the technique - K Nearest neighbor density estimation. ZDT family has widely used test problems. It is used to test the proposed algorithm. The described algorithm outperforms the SPEA2, gives better performance compared to other preeminent Multiobjective algorithms, based on the results collected from the simulation. This concept of Generational Crossover is generic. It can be used with other multiobjective evolutionary algorithms. A comparison is performed in this paper between the standard SPEA2 algorithm and the described modified SPEA2 algorithm. They also compared their proposed modified SPEA2 algorithm with other MOEAs like PAES, OMOPSO, MOEAD, DENSEA, FastPGA, IBEA, MOCcell, SMSEMOA, PESA2, CeILDE, NSGAII, AbySS, eMOEA, eNSGAI, SMPSO. The ZDT family has benchmark problem which was used to compare MOEAs like NSGAII and SPEA2 with the proposed algorithm. The proposed improved SPEA2 algorithm gives better results on all problems based on the experimental results. It is better than NSGAII and SPEA2 algorithms on all tested problems excluding ZDT4. It is a well diversified solution which is proved when the result of the IMP-SPEA algorithm is comparatively better than the above described algorithm. As shown MOEAs give good results in specific problems but the proposed algorithm gives good results for all problems. To conclude the above proposed algorithm is an agreeable and effective alternative to deal with the MOO problems.

Chapter 4

Methodology

A thesis study's methodology reflects the sequential work process. After conducting research in a specific area and evaluating the findings, viable methodologies for future study are selected. The works are completed in stages on a predetermined suitable surface. The dataset as well as the accuracies are computed and evaluated. The concurrent work procedure of our method will be highlighted in this section. We selected some practicable methods for our work consistent with past studies. The models will be constructed step by step, and the accuracy discovered will be compared to determine the best model for our process. The main goal of our thesis work is to create optimum squads based on the expected success of the players in the upcoming tournament and the auction prices of the players. Our analysis protocol is heavily reliant on two components. Machine learning techniques are used to forecast results, and multi-objective optimization techniques are used to formulate the optimum squad. Throughout the method, we use several acclaimed machine learning and multi-objective optimization algorithms to visualize our working outcomes, and after much study, we consider the final result to be optimal. Our first phase of study is focused on data from players who have previously competed in the Indian Premier League. Using the results, we were able to extract different facets of variables that have a significant impact on a player's results. The forecast can be performed specifically based on the dependency of certain attributes. We used Naive Bayes, Decision Tree, Logistic Regression, Support Vector Machine, and Random Forest classifiers to forecast results. These are several key machine learning algorithms for forecasting performance. For forecasting, any part of a player's playing attributes was examined, and forecasting of players' results was done using a suitable method of continuous analysis. Our subsequent process begins on the basis of the first steps of the report. In the later stages of the study, we want to employ a process in which the predicted success of the players is used to create an optimum squad. Multi-objective optimization methods are used to identify the squads with the highest price to value ratio when taking into account the tournament's different constraints. The optimization process must

be evaluated, and the workflow must be exactly completed using the methods we present. Furthermore, the approach we used went through a series of analyzing phases to ensure that the modified methodology was almost optimal for the type of analysis.

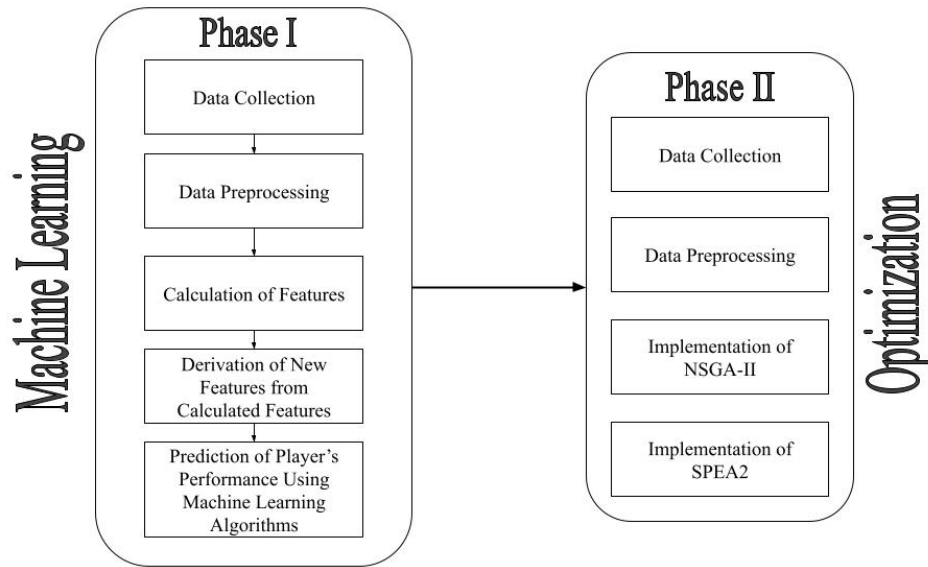


Figure 4.1: Flowchart of Methodology

4.1 Methodology of Machine Learning Algorithm Implementation Part

4.1.1 Data Collection

The first step in our research is to gather relevant data. Since we want to deal with the franchise-based competition known as the 'Indian Premier League,' the details we seek must be cricketing knowledge from players who have competed in the IPL. The tournament's first edition took place in April of 2008. We collected information about players who competed in editions from 2008 to 2019. We would reorganize the data from the accumulated data for players who are still active. Many players who have retired from cricket are not on the player list. We used several websites to gather data for the first steps of our study. These are:

- www.iplt20.com [27]
- www.howstat.com [28]
- www.cricmetric.com [29]

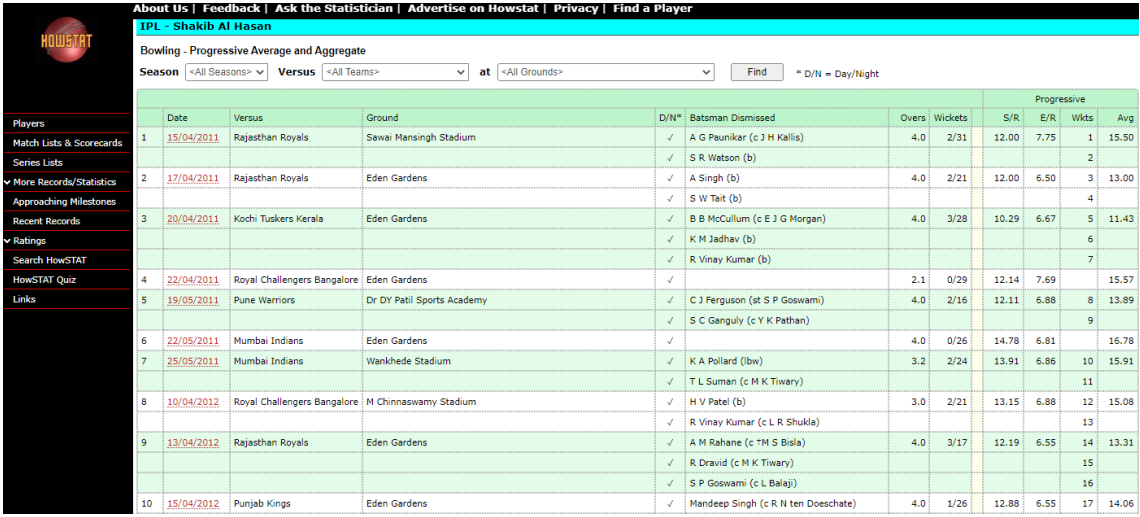
- www.espncricinfo.com [30]

4.1.1.1 Collection of Batters' Data

We searched different websites for suitable data with appropriate functionality in order to capture batters' data. We obtained the batters' information from www.howstat.com. We omitted players who mostly bowl and bat at the bottom of the order from our prediction of batter result's phase. From 2008 to the current season of the IPL, a total of 600 players have competed. We extracted 93 batters from the participants after removing former players and bowlers, and the all-rounders' batting data was also included.

4.1.1.2 Collection of Bowlers' Data

Bowlers are those players who bowl the most of their overs allotted to them, and all rounders are bowlers who also bat. We included all full-time bowlers and all-rounders' bowling data in the bowlers' data. 456 bowlers have competed in the IPL from its inception to the most recent season. Following the reinterpretation, we were able to retrieve the playing details of 97 bowlers. The data we have retrieved need various data cleaning techniques to make it suitable for our work. The visualization of the collection of bowlers data is shown below.



Date	Versus	Ground	D/N*	Batsman Dismissed	Overs	Wickets	S/R	E/R	Wkts	Avg
15/04/2011	Rajasthan Royals	Sawai Mansingh Stadium	✓	A G Paunikar (c J H Kallis)	4.0	2/31	12.00	7.75	1	15.50
			✓	S R Watson (b)					2	
17/04/2011	Rajasthan Royals	Eden Gardens	✓	A Singh (b)	4.0	2/21	12.00	6.50	3	13.00
			✓	S W Tait (b)					4	
20/04/2011	Kochi Tuskers Kerala	Eden Gardens	✓	B B McCullum (c E J G Morgan)	4.0	3/28	10.29	6.67	5	11.43
			✓	K M Jadhav (b)					6	
			✓	R Vinay Kumar (b)					7	
22/04/2011	Royal Challengers Bangalore	Eden Gardens	✓		2.1	0/29	12.14	7.69		15.57
19/05/2011	Pune Warriors	Dr DY Patil Sports Academy	✓	C J Ferguson (st S P Goswami)	4.0	2/16	12.11	6.88	8	13.89
			✓	S C Ganguly (c Y K Pathan)					9	
22/05/2011	Mumbai Indians	Eden Gardens	✓		4.0	0/26	14.78	6.81		16.78
25/05/2011	Mumbai Indians	Wankhede Stadium	✓	K A Pollard (lbw)	3.2	2/24	13.91	6.86	10	15.91
			✓	T L Suman (c M K Tiwary)					11	
10/04/2012	Royal Challengers Bangalore	M Chinnaswamy Stadium	✓	H V Patel (b)	3.0	2/21	13.15	6.88	12	15.08
			✓	R Vinay Kumar (c L R Shukla)					13	
13/04/2012	Rajasthan Royals	Eden Gardens	✓	A M Rahane (c M S Bisla)	4.0	3/17	12.19	6.55	14	13.31
			✓	R Dravid (c M K Tiwary)					15	
			✓	S P Goswami (c L Balaji)					16	
15/04/2012	Punjab Kings	Eden Gardens	✓	Mandeep Singh (c R N ten Doeschate)	4.0	1/26	12.88	6.55	17	14.06

Figure 4.2: Unprepared Dataset of Bowlers

4.1.1.3 Collection of Wicket-keepers' Data

Wicket-keepers are players who stand behind the wicket for dismissal actions such as catching and stumping. This kind of player bats as well. For our study, we collected data from 15(20) proper wicket-keepers. For the wicket-keepers, we must consider their dismissal ratio, which will be discussed later, as well as their batting performances.

4.1.1.4 Collection of Travelling Data

Cricket tournaments are usually held in different stadiums located in various cities. Currently, only eight teams compete in the Indian Premier League, and they face each other on a home-and-away basis. As a result, one team must fly to another seven teams' grounds to play them on a certain date. Due to the short duration of the tournament, players must often fly by air from one city to another. One thing is evident from Gray's writing [31] is that constant air travel affects one person's job rate. Traveling happens more often than usual during a tournament, which can have an effect on a player's results. That is why we concentrate on travel knowledge. We calculated the travel time and distance from one city to another based on the schedule of the players who play with the traveling team. We gathered travel information from www.airplanemanager.com [32] and prepared it in accordance with our work methods.

A	B	C	D	E	F
FROM_VENUE	TO_VENUE	venue1	venue2	Distance(km)	Time(hr)
Eden Gardens	ACA-VDCA Stadium	5	18	772	1.42
Eden Gardens	Barabati Stadium	5	19	350	2.6
Eden Gardens	Brabourne Stadium	5	20	1660	2.5
Eden Gardens	DY Patil Stadium	5	21	1660	2.5
Eden Gardens	Feroz Shah Kotla Ground	5	22	1320	2
Eden Gardens	Green Park Stadium	5	23	920	1.3
Eden Gardens	Holkar Stadium	5	17	1569	2.1
Eden Gardens	HPCA Stadium	5	24	1600	6.4
Eden Gardens	Jawaharlal Nehru Stadium	5	25	2835	4
Eden Gardens	JSCA International Cricket Stadium	5	10	320	0.92
Eden Gardens	M. A. Chidambaram Stadium	5	2	1380	1.92
Eden Gardens	M. Chinnaswamy Stadium	5	1	1540	2.33
Eden Gardens	Maharashtra Cricket Association Stadium	5	7	1580	2.66
Eden Gardens	PCA Stadium	5	3	1467	2.6
Eden Gardens	Raipur International Cricket Stadium/Shah	5	15	710	3
Eden Gardens	Rajiv Gandhi International Cricket Stadium	5	9	1182	2.2
Eden Gardens	Sardar Patel Stadium	5	14	1620	2.58
Eden Gardens	Saurashtra Cricket Association Stadium	5	16	1806	6.17
Eden Gardens	Sawai Mansingh Stadium	5	4	1358	2.17
Eden Gardens	VCA Stadium	5	26	980	1.92
Eden Gardens	Wankhede Stadium	5	6	1660	2.5

Figure 4.3: Traveling Dataset

4.1.2 Data Cleaning

Our analysis is heavily reliant on data obtained from different sources. We need our data to be properly cleaned and appropriate while we work on it to extract any new words that are not explicitly gettable and quantify those attributes with the aid of the attainable data. The data we obtain from various sources contains some missing values as well as some distorted values that cannot be easily derived. Data cleaning is the method by which we prepare our data for use in our work. If the data is incorrect, since we use machine learning algorithms first and then multi-objective optimization algorithms, our result would not be satisfactory. The data we obtain, a significant amount of data is nil for a certain player in a specific ground and against a specific opposition. This is because the player did not play on that field or against that opponent, so the performance attributes have zero values. These values were treated as missing values and were replaced with the class average of the respective attributes. Here is the visual of cleaned dataset we have used in our project.

Match	Date	Vs	Venue	How Dismissed	Out	Runs	Balls	TotalRuns
1	27/04/2008	Kings XI Punjab	Punjab Cricket Association Ground	b Yuvraj Singh	1	24	14	24
2	30/04/2008	Royal Challengers Bangalore	Arun Jaitley	b J H Kallis	1	2	4	26
3	4/5/2008	Mumbai Indians	Dr DY Patil Sports Ground	lbw b D J J B	1	28	23	54
4	8/5/2008	Chennai Super Kings	Arun Jaitley	not out	0	6	3	60
5	11/5/2008	Rajasthan Royals	Sawai Mansarovar	c M Rawat	1	13	15	73
6	19/05/2008	Royal Challengers Bangalore	M Chinnaswamy	c C L White	1	6	4	79
7	24/05/2008	Mumbai Indians	Arun Jaitley	not out	0	56	32	135
8	30/05/2008	Rajasthan Royals	Wankhede Stadium	c M Kaif b S	1	10	12	145
9	23/04/2009	Chennai Super Kings	Kingsmead	c A Flintoff	1	18	16	163
10	26/04/2009	Royal Challengers Bangalore	St George's	c J H Kallis b	1	12	9	175
11	28/04/2009	Rajasthan Royals	SuperSport	c & b S K Wa	1	4	5	179
12	30/04/2009	Deccan Chargers	SuperSport	lbw b P P Oj	1	41	30	220
13	2/5/2009	Chennai Super Kings	Wanderers	c M Muralith	1	52	31	272
14	8/5/2009	Mumbai Indians	Buffalo Park	not out	0	5	3	277
15	10/5/2009	Kolkata Knight Riders	Wanderers	not out	0	17	22	294
16	13/05/2009	Deccan Chargers	Kingsmead	not out	0	44	23	338
17	15/05/2009	Kings XI Punjab	Mangaung	c W A Mota	1	32	29	370
18	17/05/2009	Rajasthan Royals	Mangaung	not out	0	23	11	393
19	19/05/2009	Royal Challengers Bangalore	Wanderers	c R V Uthap	1	31	29	424
20	21/05/2009	Mumbai Indians	SuperSport	not out	0	0	1	424
21	22/05/2009	Deccan Chargers	SuperSport	b R J Harris	1	9	8	433
22	13/03/2010	Kings XI Punjab	Punjab Cricket Association Ground	c I K Pathan	1	20	18	453
23	15/03/2010	Rajasthan Royals	Barabati Stadium	not out	0	23	26	476
24	17/03/2010	Mumbai Indians	Arun Jaitley	st A P Tare	1	16	6	492
25	19/03/2010	Chennai Super Kings	Arun Jaitley	lbw b Joginc	1	19	14	511
26	21/03/2010	Deccan Chargers	Barabati Stadium	c & b A Sym	1	46	27	557
27	25/03/2010	Royal Challengers Bangalore	M Chinnaswamy	run out	1	17	19	574
28	29/03/2010	Kolkata Knight Riders	Arun Jaitley	b I Sharma	1	0	1	574
29	31/03/2010	Rajasthan Royals	Arun Jaitley	c sub b S W	1	69	38	643
30	4/4/2010	Royal Challengers Bangalore	Arun Jaitley	c C L White	1	6	4	649

Figure 4.4: Prepared Dataset for Research Work

4.1.3 Calculation of Batting Features

A batter is a player who enters the field to hit the ball with a cricket bat while attempting not to be thrown out. In a cricket match, each team has eleven batters coming to the crease. If the player is a batting specialist or not, he or she must bat when he or she comes to the field with the batting team. The type of pitch, playing position, and other factors all have an impact on a batter's output. A very good batter has good reflexes, a brilliant cricketing brain, and can adapt to any situation at any time during the game. When in a game, two members of the batting side are on the mound at any given time: the striker, who is approaching the latest delivery from the bowler, and the non-striker. When a batter is out, he or she is replaced by a teammate. This lasts to the end of the inning, which is usually when ten of the team's players are out, at which point the other team gets a chance to bat. Batting style can change depending on the situation. Batters in a test match try to face a lot of balls and score runs consistently, while in one-day and twenty-twenty cricket, the batting strategy varies. In a shorter format, batters attempt to score as many runs as possible as fast as possible by hitting fewer balls. As our research relies on Indian Premier League, we must consider the batting tactics of a twenty-twenty game. In cricket, there are certain common characteristics that are as important as scoring runs. These functionality will be discussed further below. We have considered batters of three different positions. These are:

- Top Order (1-3)
- Middle Order (4-5)
- Lower Order (6-7)

A batter's performance measurements are determined by the field on which he or she bats, the opposition against which he or she bats, the performance for a certain season of a competition, and the overall career performance [17]. Many of these considerations are taken into account when determining a few more characteristics. These four characteristics [7] are as follows:

- Consistency: The performance of a batter for his/her whole career
- Form: The performance of a batter for a particular season of tournament
- Venue-wise Performance: The performance of a batter in a particular venue
- Opposition-wise Performance: The performance of a batter against a particular opposition

These four derived characteristics would be explored in greater detail. For our analysis work to cover any part of a player's results, we consider the standard batting qualities in the following manner.

4.1.3.1 Batting Average

Batting average [13] is the metric which expresses the batting ability of a batter. The value of batting average is derived from the total runs a batter scored divided by the times the batter is out. Intuitively, the number is also easy to understand. This is the total number of runs scored per innings if all of the batter's innings were done (i.e. they were out both innings). If they did not end any of their innings (i.e. some innings were not out), this figure is an approximation of the unknown average number of runs they score each inning. This feature is derived by the following rule:

$$\text{Batting Average} = \frac{\text{Runs Scored}}{\text{Number of Times Out}}$$

The batting average is considered as one of the greatest performance metrics as it depends individually on a batter. The number of batting averages of a player differs in various formats of cricket. We have considered for a player:

- Average in whole career
- Average in each tournament proceeds
- Average in a Particular ground
- Average against a particular opposition

4.1.3.2 Batting Strike Rate

Batting strike rate is one of the most essential measures to measure how fast a batter scores. This reflects how quickly a player accelerates his performance in a match, that refers to scoring runs. To compute a batsman's strike rate [13], consider the following rule:

$$\text{Strike Rate} = \frac{\text{Runs Scored}}{\text{Balls Faced}} * 100$$

The number of batting strike rate refers to the number of runs scored by a batter per 100 balls faced. A batter's strike rate is less important in test matches than it is in limited overs cricket. Because there is a limit of overs in limited overs matches, it is regarded to score

runs more swiftly. Since our research work focuses on the data of 'Indian Premier League,' strike rate is really important. We have considered for a player:

- Strike Rate in whole career
- Strike Rate in each tournament proceeds
- Strike Rate in a Particular ground
- Strike Rate against a particular opposition

4.1.3.3 Centuries

The innings in which a batter scores more than or equal to 100 runs. We have considered for a player:

- Centuries in whole career
- Centuries in each tournament proceeds
- Centuries in a Particular ground
- Centuries against a particular opposition

4.1.3.4 Fifties

The innings in which the batter scores more than or equal to 50 runs but fewer than 100 runs. We have considered for a player:

- Fifties in whole career
- Fifties in each tournament proceeds
- Fifties in a Particular ground
- Fifties against a particular opposition

4.1.3.5 Zeroes

The innings in which the batter is out without scoring a single run. We have considered for a player:

- Zeroes in whole career

- Zeroes in each tournament proceeds
- Zeroes in a Particular ground
- Zeroes against a particular opposition

4.1.3.6 Highest Score

The most runs scored by a batter in any (single) inning in his career. We have considered for a player:

- Highest Score in whole career
- Highest Score in each tournament proceeds
- Highest Score in a Particular ground
- Highest Score against a particular opposition

4.1.4 Calculation of Bowling Features

In cricket, bowling is the act of releasing the ball with one's arm towards the batsman. There is a bowling regulation that states that the bowler must maintain the legal elbow angle. To throw a legal ball, the bowler ought not breach the bowling line's margin. If he or she steps over the line when throwing the ball, the ball is deemed a 'No-Ball.' The bowler is also not permitted to throw the ball straight to the batter over waist height. The wide line margin refers to the two margins at the crease on both sides of the batter. A bowler must throw the ball inside the margin to avoid being punished. The bowler can be penalized by both on-field and off-field umpires for throwing an illegal delivery. The result of a cricket match is determined equally by a team's batters and bowlers. Bowlers must take wickets as soon as possible so that the batting side cannot score many runs. Bowlers must also ensure that they do not give away many runs, which may put a higher target on them when the team's batters come into bat to chase the target. In the cricketing world, there are various measurements that are used to assess a player's bowling skill. In this study, we evaluate traditional characteristics as well as impose some derived factors that we believe can affect a bowler's performance. Because we are working on the 'Twenty-Twenty' format of cricket match, where the economy rate of a bowler indicates the number of runs scored by a batter in a single over of a bowler is more essential than the format of 'Test Match'. A bowler's major feature is the amount of wickets he has collected. Therefore, an economical bowler is also regarded favorably in terms of the game's format. Bowlers come in a variety of styles. Some of them throw the ball very fast, some make turns, and some swing the ball inside

or away from the batter to generate wicket-taking opportunities and to put pressure on the batter and the batting team. The performance of a bowler might vary depending on the ground and the opponents. Because the grounds contain pitches, certain pitches assist the batter in batting and others generate spins that make it difficult for the batter to play the ball. Pitches with grass assist fast bowlers in creating additional speed and swing to their delivery. Dusty pitches aid spin bowlers in turning the ball more than any other venue. There are several elements to consider that may have an effect on a bowler's performance. Many of these factors are used while deciding a few other traits. These four characteristics are as follows:

- Consistency: The performance of a bowler for his/her whole career
- Form: The performance of a bowler for a particular season of tournament
- Venue-wise Performance: The performance of a bowler in a particular venue
- Opposition-wise Performance: The performance of a bowler against a particular opposition

These four derived characteristics would be explored in greater detail. For our analysis work to cover any part of a player's results, we consider the standard bowler qualities in the following manner.

4.1.4.1 Bowling Average

In cricket, the bowling average [16] refers to the number of runs conceded against a single wicket. The lower the bowling average, the better the bowler. The bowling average is one of the primary indicators of a bowler's capability of bowling. In all formats of cricket, the metric is equally important for measuring a bowler's performance. We can derive the bowling average from the following formula:

$$\text{Bowling Average} = \frac{\text{Runs Conceded}}{\text{Wickets Taken}}$$

We have considered for a player:

- Bowling average in whole career
- Bowling average in each tournament proceeds
- Bowling average in a Particular ground
- Bowling average against a particular opposition

4.1.4.2 Bowling Strike Rate

Bowling strike rate [3] refers to how quickly a bowler takes wickets. This means how many balls a bowler bowls to take per wicket. This is also an important metric to sense the performance of a bowler. The value of bowling strike rate can be derived as follows:

$$\text{BowlingStrike Rate} = \frac{\text{Number of Balls Bowled}}{\text{Number of Wickets Taken}}$$

We have considered for a player:

- Bowling strike rate in whole career
- Bowling strike rate in each tournament proceeds
- Bowling strike rate in a Particular ground
- Bowling strike rate against a particular opposition

4.1.4.3 Three Wickets Haul

This number measures how many times a bowler takes three or more wickets in an inning. Traditionally, in the statistics of test matches and one-day matches, the haul of five wickets in an inning is considered, but because we are working on the twenty-twenty format, in which a player only bowls four over per inning, the three wicket haul is a significant criterion for measuring a player's performance. We have considered for a player:

- Three wickets haul average in whole career
- Three wickets haul in each tournament proceeds
- Three wickets haul in a Particular ground
- Three wickets haul against a particular opposition

4.1.4.4 Economy Rate

In limited overs cricket, the attribute named economy rate [3] has a greater influence for assuming the performance of a bowler. This refers to the number of runs conceded in an over. A bowler who has a lower economy rate is more crucial than one who has more. The value of economy rate can be derived as follows:

$$\text{Bowling Economy Rate} = \frac{\text{Number of Runs Conceded}}{\text{Number of Overs Bowled}}$$

- Economy Rate in whole career
- Economy Rate in each tournament proceeds
- Economy Rate in a Particular ground
- Economy Rate against a particular opposition

4.1.5 Calculation of Fielding Features of Wicket-Keepers

When a bowler from the bowling team bowls, the remaining 10 members of the team stay on the field. Fielders are all eleven players, including the bowler. Fielders attempt to catch the ball, throw the ball at the stumps to strike out a batter, and preserve runs by grabbing the ball as soon as possible. The fielding skills of wicket-keepers is taken into account in this study report. Wicket-keepers are fielders who stand or sit behind the stumps and remain alert in case a batter is run out or a catch is made. We do not attempt to anticipate a wicket-fielding keeper's performance; rather, we combine the fielding performance of a wicket-keeper with his/her batting performance to provide a comprehensive picture of the player himself/herself.

4.1.5.1 Catches

To catch the ball is a means of dismissing a batter in cricket. In order for a batter to be caught, the batsman must: hit the ball, from such a genuine delivery, with bat, as well as the ball be managed to catch by bowler or even a fielder before it touches the ground. It is common for wicket-keepers to get credit for a catch taken when the wicket is taken since this is referred to informally as being behind or at the wicket. The bowler's act of catching a ball with his hand is termed as "bowled" and "caught."

4.1.5.2 Stumping

In cricket, a batsman is dismissed when they are "stumped". Stumping is a technique of removing a batsman that entails the wicket-keeper trying to put down the wicket when the batter is out of the position. The batter is no longer in his area if he has gone past the popping crease, which is frequently the case in order to attempt to strike the ball. If the wicket-keeper is able to grab the ball which the batter misses and put the bails out of the stumps before returning to the margin of the popping crease, the batter is stumped out. It is one of the fielding team's responsibilities to appeal for the wicket by requesting the umpire. Normally, the appeal is sent to the square-leg umpire, who might be in the best possible position to provide a decision on the appeal.

4.1.5.3 Dismissal Rate

Dismissal rate refers to the number of catches and stumpings done in a single inning by a wicket-keeper. The formulation of dismissal rate as follows:

$$\text{Dismissal Rate} = \frac{(\text{Number of Catches} + \text{Number of Stumpings})}{\text{Number of Match Played}}$$

4.1.6 Calculation of Travelling Features

Cricket tournament fixtures span a number of days, where participating teams have the opportunity to play each other at different locations. The fields where cricket matches are played are placed in several locations across a country, necessitating the movement of the team members from one region to another when a tournament is in progress. Tournaments are held in a shorter period of time, which makes it possible for players to play a lot of matches over a period of time that is still quite brief. Due to this, traveling regularly from one location to another has a huge impact on a player's overall performance. We have included a working study that focuses on the traveling influence which may hinder a player's ability to perform.

4.1.6.1 Distance among Airports of Playing Venues'

In our research, we have taken into account the flight distance among venues which are situated in different cities. When consecutive matches are played in a single venue the value of flight distance is considered as zero.

4.1.6.2 Time for Air-Travel among Playing Venues'

Time for travelling is also a big issue which may affect a player's performance. Larger travelling time imposes fatigue to players which results in bad performance of a player. So, in our research along with the flight distance we have considered the flight time for any journey which had been made by a player during the tournament.

4.1.7 Rating of Features' Values

Rating is one of the techniques to classify data in various ranges. For our research work, we rate all the traditional features of batter and bowlers with respect to appropriate criteria to make a class of 1 to 5. As we have data which are labelled, so we make a proper dataset for our work with supervised classified features which may be manipulated by the classification

algorithms to make further predictions from the ratings. We consider ranging the features to bring balance among values of different features. In cricket, the values are not even at all, which may have spikes more often than other working sectors. So with proper ranging, putting a rating for each range makes the task easier for the further progress of our research work.'

4.1.7.1 Rating of Batting Features'

The batting features we have considered are given here:

- Number of Innings
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Batting Average
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Strike Rate
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Number of Hundred
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Number of Fifty

- In Whole Career
- In Each Tournament
- In a Particular Ground
- Against a Particular Opposition
- Number of Zero
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Highest Score
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition

All the features' values are ranged according to the playing position of a batter. We have considered three positions for a batter. These are:

- Top Order Batter (1st, 2nd and 3rd Batter)
- Middle Order Batter (4th and 5th Batter)
- Lower Order Batter (6th, 7th and Lower Order's Batter)

The values of each feature are ranged for each type of batter. For Top order batters, we have collected all the values of all features which we have mentioned above and then for each feature the values are sorted from lower to higher. We took the median value of each feature and range the values into 5 classes as follows:

$$\begin{aligned}
 \text{Range 1} &= \frac{(\text{Median-Lowest value of the feature})}{2} + \text{Lowest value of the feature} \\
 \text{Range 2} &= \frac{(\text{Median-Lowest value of the feature})}{2} + \text{Range 1's value} \\
 \text{Range 3} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 2's value} \\
 \text{Range 4} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 3's value} \\
 \text{Range 5} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 4's value}
 \end{aligned}$$

Then the ranges are sorted as follows:

$\text{Lowest value of the feature} \leq \text{Rating '1'} < \text{Range 1}$

$\text{Range 1} \leq \text{Rating '2'} < \text{Range 2}$

$\text{Range 2} \leq \text{Rating '3'} < \text{Range 3}$

$\text{Range 3} \leq \text{Rating '4'} < \text{Range 4}$

$\text{Range 4} \leq \text{Rating '5'} < \text{Range 5 or Higher}$

For middle order batters, and lower order batters the values are calculated according to the players' data of these positions respectively. The ratings are presented in a tabular form below:

Table 4.1: Rating of Features of All Types of Batters

Name of the Attribute	Name of the Derived Attribute	Top Order Batsman	Middle Order Batsman	Lower Order Batsman
No. of Innings:	Consistency	0 - 41: 1	0 - 41: 1	1 - 7: 1
		42 - 71: 2	42 - 71: 2	8 - 14: 2
		72 - 101: 3	72 - 101: 3	15 - 21: 3
		102 - 131: 4	102 - 131: 4	22- 96: 4
		≥ 132 : 5	≥ 132 : 5	≥ 97 : 5
	Form	0 - 3: 1	0 - 3: 1	0 - 3: 1
		4 - 8: 2	4 - 8: 2	4 - 8: 2
		9 - 10: 3	9 - 10: 3	9 - 10: 3
		11 - 13: 4	11 - 13: 4	11 - 13: 4
		≥ 14 : 5	≥ 14 : 5	≥ 14 : 5
	Opposition	0 - 7: 1	0 - 7: 1	1 - 2: 1
		8 - 12: 2	8 - 12: 2	3 - 4: 2
		13 - 18: 3	13 - 18: 3	5 - 6: 3
		19 - 24: 4	19 - 24: 4	7 - 16: 4
		≥ 25 : 5	≥ 25 : 5	≥ 17 : 5
	Venue	0 - 11: 1	0 - 11: 1	1 - 3: 1
		12 - 19: 2	12 - 19: 2	4 - 6: 2
		20 - 34: 3	20 - 34: 3	7 - 9: 3
		35 - 49: 4	35 - 49: 4	8 - 30: 4
		≥ 50 : 5	≥ 50 : 5	≥ 31 : 5
		0: 0	0: 0	0: 0

Table 4.1 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Top Order Batsman	Middle Order Batsman	Lower Order Batsman
		1 - 4: 1	1 - 3: 1	1 - 5: 1
		5 - 10: 2	4 - 6: 2	6 - 10: 2
		11 - 19: 3	7 - 14: 3	11 - 15: 3
		20 - 28: 4	15 - 22: 4	16 - 20: 4
		>= 29: 5	>= 23: 5	
	Form	0: 0	0: 0	0: 0
		1 - 2: 1	1 - 2: 1	1 - 2: 1
		3 - 4: 2	3 - 4: 2	3 - 4: 2
		5 - 6: 3	5 - 6: 3	5 - 6: 3
		7 - 8: 4	7 - 8: 4	7 - 8: 4
		>= 9: 5	>= 9: 5	
	Opposition	0: 0	0: 0	0: 0
		1 - 5: 1	1 - 2: 1	1: 1
		6 - 10: 2	3 - 4: 2	2: 2
		>= 11: 3	5: 3	3: 3
			>5: 4	4: 4
	Venue	0: 0	0: 0	0: 0
		1 - 8: 1	1 - 3: 1	1: 1
		9 - 16: 2	4 - 6: 2	2: 2
		>= 17: 3	>= 7: 3	3: 3
				4: 4
Hundred	Consistency	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		2 - 3: 2	2 - 3: 2	>= 2: 2
		>= 4: 3	>= 4: 3	
	Form	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		2: 2	2: 2	2: 2
		>= 3: 3	>= 3: 3	>= 3: 3
	Opposition	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		2: 2	>= 2: 2	>= 2: 2
		>= 3: 3		

Table 4.1 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Top Order Batsman	Middle Order Batsman	Lower Order Batsman
	Venue	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		>= 2: 2	>= 2: 2	>= 2: 2
Zero	Consistency	0: 0	0: 0	0: 0
		1 - 4: 1	1 - 4: 1	1 - 4: 1
		5 - 9: 2	5 - 9: 2	5 - 9: 2
		10 - 14: 3	10 - 14: 3	10 - 14: 3
		15 - 19: 3	15 - 19: 3	15 - 19: 3
		>= 20: 3	>= 20: 3	
	Form	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		2: 2	2: 2	2: 2
		3: 3	3: 3	3: 3
		4: 4	4: 4	4: 4
		>= 5: 5	>= 5: 5	
	Opposition	0: 0	0: 0	0: 0
		1: 1	1: 1	1: 1
		2: 2	2: 2	2: 2
		3: 3	3: 3	3: 3
		4: 4	4: 4	4: 4
		>= 5: 5	>= 5: 5	
Batting Average	Consistency	0.0 - 23.99: 1	0.0 - 9.99: 1	3.0 - 7.99: 1
		24.0 - 30.99: 2	10.0 - 19.99: 2	8.0 - 12.99: 2
		31.0 - 37.99: 3	20.0 - 27.99: 3	13.0 - 18.99: 3
		>= 38.0: 4	>= 28.0: 4	>= 19.0: 4
	Form	0.0 - 23.99: 1	0.0 - 23.99: 1	0.0 - 23.99: 1
		24.0 - 30.99: 2	24.0 - 30.99: 2	24.0 - 30.99: 2
		31.0 - 37.99: 3	31.0 - 37.99: 3	31.0 - 37.99: 3
		>= 38.0: 4	>= 38.0: 4	>= 38.0: 4
	Opposition	0.0 - 14.99: 1	0.0 - 7.99: 1	0.0 - 4.99: 1
		15.0 - 30.99: 2	8.0 - 15.99: 2	5.0 - 9.99: 2
		31.0 - 107.99: 3	16.0 - 54.99: 3	10.0 - 14.99: 3
		108.0 - 183.99: 4	55.0 - 93.99: 4	15.0 - 40.99: 4

Table 4.1 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Top Order Batsman	Middle Order Batsman	Lower Order Batsman
		$\geq 184.0: 5$	$\geq 94.0: 5$	$\geq 41.0: 5$
	Venue	0.0 - 12.99: 1	0.0 - 6.99: 1	0.0 - 3.99: 1
		13.0 - 26.99: 2	7.0 - 13.99: 2	4.0 - 7.99: 2
		27.0 - 62.99: 3	14.0 - 56.99: 3	8.0 - 11.99: 3
		63.0 - 98.99: 4	57.0 - 99.99: 4	12.0 - 51.99: 4
		$\geq 99.0: 5$	$\geq 100.0: 5$	$\geq 52.0: 5$
Batting Strike Rate	Consistency, Form, Venue	0.0 - 99.99: 1	0.0 - 65.99: 1	0.0 - 104.99: 1
		100.0 - 125.99: 2	66.0 - 130.99: 2	105.0 - 134.99: 2
		126.0 - 131.99: 3	131.0 - 183.99: 3	135.0 - 152.99: 3
		132.0 - 140.99: 4	184.0 - 236.99: 4	153.0 - 170.99: 4
		$\geq 141.0: 5$	$\geq 237.0: 5$	$\geq 171.0: 5$
	Opposition	0.0 - 65.99: 1	0.0 - 53.99: 1	0.0 - 45.99: 1
		66.0 - 131.99: 2	54.0 - 107.99: 2	46.0 - 79.99: 2
		131.0 - 179.99: 3	108.0 - 183.99: 3	80.0 - 113.99: 3
		180.0 - 228.99: 4	184.0 - 259.99: 4	114.0 - 224.99: 4
		$\geq 229.0: 5$	$\geq 260.0: 5$	$\geq 225.0: 5$
Highest Score	Consistency, Form, Venue, Opposition	0:0	0:0	0.0 - 7.99: 1
		1 - 16: 1	1 - 24: 1	8.0 - 15.99: 2
		17 - 33: 2	25 - 49: 2	16.0 - 39.99: 3
		34 - 104: 3	50 - 67: 3	40 - 63.99: 4
		105 - 175: 4	68 - 118: 4	$\geq 64.0: 5$
		$\geq 176: 5$	$\geq 119: 5$	
Run	Consistency, Form, Venue, Opposition	0 - 24: 1	0 - 24: 1	0 - 24: 1
		25 - 49: 2	25 - 49: 2	25 - 49: 2
		50 - 74: 3	50 - 74: 3	50 - 74: 3
		75 - 99: 4	75 - 99: 4	75 - 99: 4
		$\geq 100: 5$	$\geq 100: 5$	$\geq 100: 5$

4.1.7.2 Rating of Bowling Features'

The bowling features we have considered are given here:

- Number of Innings
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Bowling Average
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Bowling Strike Rate
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Three Wickets Haul
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition
- Economy Rate
 - In Whole Career
 - In Each Tournament
 - In a Particular Ground
 - Against a Particular Opposition

The values of each feature are ranged for bowlers. For bowlers, we have collected all the values of all features which we have mentioned above and then for each feature the values are sorted from lower to higher. We took the median value of each feature and range the values into 5 classes as follows:

$$\begin{aligned} \text{Range 1} &= \frac{(\text{Median-Lowest value of the feature})}{2} + \text{Lowest value of the feature} \\ \text{Range 2} &= \frac{(\text{Median-Lowest value of the feature})}{2} + \text{Range 1's value} \\ \text{Range 3} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 2's value} \\ \text{Range 4} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 3's value} \\ \text{Range 5} &= \frac{(\text{Highest value of the feature-Median})}{3} + \text{Range 4's value} \end{aligned}$$

Then the ranges are sorted as follows:

$$\text{Lowest value of the feature} \leq \text{Rating '1'} < \text{Range 1}$$

$$\text{Range 1} \leq \text{Rating '2'} < \text{Range 2}$$

$$\text{Range 2} \leq \text{Rating '3'} < \text{Range 3}$$

$$\text{Range 3} \leq \text{Rating '4'} < \text{Range 4}$$

$$\text{Range 4} \leq \text{Rating '5'} < \text{Range 5 or Higher}$$

The ratings are presented in a tabular form below:

Table 4.2: Bowler's Rating

Name of the Attribute	Name of the Derived Attribute	Rating
No. of Innings:	Consistency	0 - 16: 1
		17 - 31: 2
		32 - 75: 3
		76 - 119: 4
		>= 119: 5
	Form	0 - 3: 1
		4 - 8: 2
		9 - 10: 3
		11 - 13: 4
		>=14: 5
	Opposition	0 - 7: 1
		8 - 12: 2
		13 - 18: 3
		19 - 24: 4
		>=25: 5
		0 - 11: 1

Venue

Table 4.2 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Rating
		12 - 19: 2
		20 - 34: 3
		35 - 49: 4
		≥ 50 : 5
Overs:	Consistency	0.0 - 52.9: 1
		53.0 - 101.9: 2
		102.0 - 199.9: 3
		200.0 - 255.9: 4
		≥ 256.0 : 5
	Form	0.0 - 9.5: 1
		10.0 - 24.5: 2
		25.0 - 49.5: 3
		50.0 - 99.5: 4
		≥ 100.0 : 5
	Opposition	0.0 - 5.6: 1
		6.0 - 10.6: 2
		11.0 - 35.6: 3
		36.0 - 60.6: 4
		≥ 61.0 : 5
	Venue	0.0 - 5.6: 1
		6.0 - 10.6: 2
		11.0 - 65.6: 3
		66.0 - 131.6: 4
		≥ 131.0 : 5
Average:	Consistency	0.0 - 20.9: 1
		21.0 - 27.9: 2
		28.0 - 38.9: 3
		39.0 - 49.9: 4
		≥ 50.0 : 5
	Form	0.0 - 24.9: 1
		25.0 - 29.9: 2
		30.0 - 34.9: 3
		35.0 - 49.9: 4
		≥ 50.0 : 5
	Opposition	2.0 - 13.9: 1
		14.0 - 25.9: 2
		26.0 - 69.9: 3

Table 4.2 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Rating
		70.0 - 113.9: 4
		>=114.0: 5
	Venue	2.0 - 15.9: 1
		16.0 - 31.9: 2
		32.0 - 71.9: 3
		72.0 - 111.9: 4
		>=112.0: 5
Strike Rate :	Consistency	8.0 - 13.9: 1
		14.0 - 19.9: 2
		20.0 - 27.9: 3
		28.0 - 36.9: 4
		>=37.0: 5
	Form	0.0 - 29.9: 1
		30.0 - 39.9: 2
		40.0 - 49.9: 3
		50.0 - 59.9: 4
		>=60.0: 5
	Opposition	0.0 - 21.9: 1
		22.0 - 39.9: 2
		40.0 - 67.9: 3
		68.0 - 96.9: 4
		>=97.0: 5
	Venue	3.0 - 10.9: 1
		11.0 - 21.9: 2
		22.0 - 54.9: 3
		55.0 - 87.9: 4
		>=88.0: 5
3 Wickets+ :	Consistency	1 - 2: 2
		3 - 4: 3
		>=5: 5
	Form	1 - 2: 4
		>=3: 5
	Opposition	1 - 2: 4
		>=3: 5
	Venue	1 - 2: 4
		>=3: 5
		0.0 - 4.9: 5

Table 4.2 continued from previous page

Name of the Attribute	Name of the Derived Attribute	Rating
		5.0 - 6.9: 4
		7.0 - 8.9: 3
		9.0 - 10.9: 2
		>=11.0: 1
	Form	0.0 - 4.9: 5
		5.0 - 6.9: 4
		7.0 - 8.9: 3
		9.0 - 10.9: 2
		>=11.0: 1
	Opposition	0.0 - 4.9: 5
		5.0 - 6.9: 4
		7.0 - 8.9: 3
		9.0 - 10.9: 2
		>=11.0: 1
	Venue	0.0 - 4.9: 5
		5.0 - 6.9: 4
		7.0 - 8.9: 3
		9.0 - 10.9: 2
		>=11.0: 1
Wickets :	Consistency, Form, Venue, Opposition	0 - 1: 1
		2: 2
		3 : 3
		4: 4
		>=5: 5

4.1.8 Calculation of Derived Features

The features of any study is the content of the research. Using machine learning techniques, we made our way through the first step of our research by predicting the sports performance of a player. For these traits, we have included unique elements specifically for batter and bowler. They include two aspects which we've already discussed, as well as the four unique elements listed below, considering every circumstance that may be encountered through-

out a particular tournament. A single player's tournament's performance values of derived quantities fluctuate from tournament to tournament. To that end, we always bear in mind many facets, such as a player's previous performances and the caliber of the athlete. We may locate a player who is often underperforming in a given competition because of a variety of shortcomings which we in cricketing terminology call "Form" [4]. To the above two points, we may add the fact that there is a cricket player known as "A" who plays cricket for a long length of time and is a well regarded player. For this tournament kind of player, their performances in several matches will be somewhat below their typical standard, but after the third or fourth match they are bound to recover. Therefore, we regard a player's performance over the course of his or her whole career as the main criterion for assessing performance. More importantly, there are a few more considerations one should consider while picking a team's preferred sports opponent or ground. We may often hear that a player discusses his or her favorite opponent to whom he or she feels at home on the opposing squad. Even if a player does not think he/she is particularly effective against certain opposing teams, he/she may choose to view those opponents as the source of their struggles. Considering that there are both beneficial and unfavorable things that oppose us, we set out to research and formulate this idea. In addition, various factors which are associated with the venue must be considered. In cricket matches, the batsman and bowler occupy their usual positions on the pitch. Types of pitches include, but are not limited to, sinkers, cutters, splitters, and chinkers. Some pitches are green, and those are covered in grass. Some pitches are dusty, and some are named "Flat." Some pitches are labeled "Sporting," and these are different from all the others. A batter and bowler whose hitting and bowling style is strong on the favorable side of the pitch may do well in a particular sort of pitch. When determining how well someone will do, the sort of pitch that they deliver becomes an important component. Historically, grounds have traditionally been known for producing pitches of similar styles, which implies that a ground seldom alters the sorts of pitches it produces. There are very few pitch conditions data to go off of when determining how much or how little each pitch will affect a player.. As the data is not available greatly about match-wise pitch conditions and the data can be messy when we work on a large set of players, we consider grounds which may have an effect on a player's performance. For this project, we had to keep all of the aspects in mind, and thus we developed our new qualities. We have introduced four attributes and the idea of analysing in this way comes from Kalpdrum Passi , Niravkumar Pandey's paper [7]. The attributes are given here:

- Consistency
- Form
- Opponent-wise Performance
- Venue-wise Performance

These four attributes have been derived by taking a weighted average of the traditional criteria given above. To solve the problem, the weights were figured out by applying the Thomas L. Saaty's method of [22] analytic hierarchy technique.

4.1.8.1 Calculation of Consistency

Consistency refers to the attribute which derives a value that indicates how a player performs in his whole career. The value of the attribute is derived by the traditional attributes in multiplication of appropriate coefficients with those. We take the value of each feature for a batter/bowler in the view of his/her whole career. The formula of consistency for batter is:

$$\text{Consistency} = 4 * \text{Rating of Batting Average} + 3 * \text{Rating of No of Innings} + 2 * \text{Rating of Batting Strike Rate} + 0.3 * \text{Rating of Centuries} + 0.5 * \text{Rating of Fifties} - 0.2 * \text{Rating of Zeros}$$

The formula of consistency for bowler is:

$$\text{Consistency} = 0.4174 * \text{Rating of No of Overs} + 0.2634 * \text{Rating of No of Innings} + 0.1602 * \text{Rating of Bowling Strike Rate} + 0.0975 * \text{Rating of Bowling Average} + 0.0615 * \text{Rating of Three Wickets Haul}$$

4.1.8.2 Calculation of Form

Form of a player refers to the indicator of performance in a particular tournament. The value of form initializes in every season and with the progression of the tournament the values are updated accordingly. For a batter/bowler a good form value means his/her performance is good in that tournament. We consider tournament-wise data for batter and bowler to derive the value of form with proper coefficient weights assigned to each feature.

The formula of form for batter is:

$$\text{Form} = 4 * \text{Rating of Batting Average} + 3 * \text{Rating of No of Innings} + 2 * \text{Rating of Batting Strike Rate} + 0.3 * \text{Rating of Centuries} + 0.5 * \text{Rating of Fifties} - 0.2 * \text{Rating of Zeros}$$

The formula of form for bowler is:

$$\text{Form} = 0.3269 * \text{Rating of No of Overs} + 0.2846 * \text{Rating of No of Innings} + 0.1877 * \text{Rating of Bowling Strike Rate} + 0.1210 * \text{Rating of Bowling Average} + 0.0798 * \text{Rating of Three Wickets Haul}$$

4.1.8.3 Calculation of Venue-Wise Performance

The term venue-wise performance indicates how well a player performs in a particular venue. We take venue-wise data for all the traditional features for this derivation. The

formula of venue-wise performance for batter is:

$$\text{Venue} = 4 * \text{Average} + 3 * \text{Rating of No of Innings} + 2 * \text{Rating of Batting Strike Rate} + 0.3 * \text{Rating of Centuries} + 0.5 * \text{Rating of Fifties} + 0.2 * \text{Rating of Highest Score}$$

The formula of venue-wise performance for bowler is:

$$\text{Venue} = 0.3018 * \text{Rating of No of Overs} + 0.2783 * \text{Rating of No of Innings} + 0.1836 * \text{Rating of Bowling Strike Rate} + 0.1391 * \text{Rating of Bowling Average} + 0.0972 * \text{Rating of Three Wickets Haul}$$

4.1.8.4 Calculation of Opposition-Wise Performance

This feature has to do with how well a player performs against a specific opposition. When we process opposition-wise data for all the usual attributes, we end up with a different derivation.

The formula of opposition-wise performance for batter is:

$$\text{Opposition} = 4 * \text{Rating of Batting Average} + 3 * \text{Rating of No of Innings} + 2 * \text{Rating of Batting Strike Rate} + 0.3 * \text{Rating of Centuries} + 0.5 * \text{Rating of Fifties} - 0.2 * \text{Rating of Zeros}$$

The formula of opposition-wise performance for bowler is:

$$\text{Opposition} = 0.3177 * \text{Rating of No of Overs} + 0.3177 * \text{Rating of No of Innings} + 0.1933 * \text{Rating of Bowling Strike Rate} + 0.1465 * \text{Rating of Bowling Average} + 0.0943 * \text{Rating of Three Wickets Haul}$$

4.1.9 Method of Prediction Using Machine Learning Algorithm

Machine learning is all about utilizing computational power and techniques to mimic how humans learn. This is a component of artificial intelligence. Algorithms, or formulas, have long been used to define the learning process, but researchers have seen significant improvement in the creation of new algorithms in the last few years. With the help of sufficient computer power, these days machine learning methods are rather straightforward to apply to derive a trend. During our research, we apply machine learning approaches to forecast the performance of players for a particular tournament. The predicted result will go to the next phase where a fully optimal squad within a fixed budget will be constructed with the primary goal of providing the owner with the greatest possible value for money. There are a variety of machine learning methods that can support our work by giving us an approximation of how well the system will function. Various options were tried from these, and finally, the best result was obtained with the help of the algorithm that was discovered using this. As we have used balanced data, accuracy is the most appropriate measure for us to use when we evaluate the algorithm's performance. But while we have access to things like

precision, f1 score, and recall, we can also try to ensure that the result we choose comes from the best algorithm. In the field of study, we have relied on classification algorithms. In great depth, below is the description of the classification techniques we have used for the prediction of performance. In the machine learning phase, we have calculated four terms, namely Consistency, Form, Venue-wise Performance, Opposition-wise performance which have also done in [7] and [3] in a slightly different way. Apart from these features another two features are taken into action, and these are: Distance between venues of consequent matches and time taken to travel from one venue to another for consequent matches. So for input, we have given 8 features. These are:

- Opponent's Index
- Venue's Index
- Consistency
- Form
- Venue-wise Performance
- Opposition-wise Performance
- Distance between venues of consequent matches
- Time taken to travel from one venue to another for consequent matches

We have used five machine learning classification algorithms to forecast the runs and wickets of batters and bowlers respectively. The name of these five algorithms are:

- Naive Bayes Classifier
- Support Vector Machine Classifier
- K-Nearest Neighbors Classifier
- Decision Tree Classifier
- Random Forest Classifier

4.2 Methodology of Multi-Objective Optimization Implementation Part

MOO (multi-objective optimization) [25] is a branch of decision-making that deals with situations containing more than one objective function that must be maximized or minimized

at the same time. It is also known as vector optimization, multi-attribute optimization, multicriteria optimization, and Pareto optimization. MOO has been applied to many domains of research, including engineering, wherein optimal decisions must be made in the context of trade-offs involving two or more objectives that could be at odds with one another. Indeed, in many actual engineering applications, designers must choose between competing objectives—for example, optimizing performance while simultaneously decreasing fuel consumption and pollution emissions from a car. In these situations, a multiobjective optimization research should be carried out, which will yield many solutions that represent the trade-offs between the various objective functions. In our research, we used machine learning approaches to make predictions about the performances of participants. However, the forecast is insufficient for team management to form a squad for a tournament in which players are selected from a pool of candidates. Due to the tremendous amount of effort required, it is a difficult process. Team managers are allocated a set amount of money, and by allocating this money, the management hopes to create a team that is appropriately balanced and provides the best value for money. Multi-objective optimization techniques are used in this situation since we have objectives such as a tight budget and to make a well-balanced team for the entire competition based on the expected performance for the event. Here in this chapter we will discuss the methodology of the implementation of multi-objective optimization phase to our research work.

4.2.1 Data Collection

For doing some statistical research we always need an ample amount of data. Like the machine learning implementation phase, in this phase we also need some data to proceed our research work. The data we needed and collected are described hereby.

4.2.1.1 Collection of Dataset of Base-Price of Players'

The performance predictions for the players who will be utilized in this phase have been produced from the machine learning portion. Furthermore, as we previously stated, in a franchise-based tournament, each team is given a fixed amount of money to spend in order to assemble a squad that will compete in the tournament, and the base prices of the players are fixed in the sense that bidding will start at the base price and increase as the bidding progresses. The base prices of the players in the Indian Premier League are determined by the league's governing body. The basic price of the players will be taken into consideration when determining the pricing of the players. As a result, we have gathered the information on the base pricing of the players for the 2018 Indian Premier League (IPL), because in 2018, a mega auction was held in which 169 players were auctioned from a pool of 578 players. At

Table 4.3: Class-wise Run for Batters

Range	Class	Considered Runs for Estimation of Batting Performance
0-20	1	20
21-40	2	40
41-60	3	60
61-80	4	80
≥ 81	5	81

that edition of the IPL, 18 players were kept to their previous retained team, and as a result, they were not sold in the auction. For those players, we have taken into consideration the price at which they were sold in that edition of the tournament, as well as the basic price that was indicated for them at the time. We have acquired data from NDTV [33], Indian Express [34] and Mykhel's [35] website and have cleaned and preprocessed it further in order to use it in our work.

4.2.2 Data Preprocessing

In the optimization phase, we used the forecast of the performance of a player which was obtained by using machine learning techniques on the player's dataset. The predicted performance solely was not enough to visualize the performance of a player. We need to focus on some traditional cricketing features which seem to be indicators of a player's performance. So along with our predicted runs for batters, predicted wickets for bowlers, we merge the predicted runs or wickets of batters' and bowlers' with some traditional features to obtain batting performance and bowling performance for batters and bowlers respectively. For all-rounders, the batting and bowling performances are both calculated. For the wicketkeepers, along with the batting performance, his/her keeping performance is merged. Here, in these following sections, we will describe the estimation procedures of each category's player's performance.

4.2.2.1 Extraction of Predicted Performance from Machine Learning Phase

We have used machine learning techniques for the prediction of the class of runs and wickets for batters and bowlers respectively. As we have used classification algorithms, we have to make classes of runs and wickets for the prediction phase. The following table shows the class of runs for batters hereby: Here, in the table we can see that, the machine learning classification techniques try to predict the class of the run for a batter for each of the matches of a tournament, but for the estimation of the performance of a batter, we need a discrete amount of runs. So, for the prediction of runs, we consider the upper bound of each limit for the predicted runs of a batter then we add all the runs predicted for each match of

the upcoming tournament to see how many runs the batter will score in the tournament while playing against each team in a home and away manner. With the total predicted runs for the tournament, we have derived the value of batting performance with the help of some other traditional batting features which are described later. For the bowlers, we have similarly taken 5 classes of prediction where the number of wickets is a continuous range. For estimating the bowling performance, we need to have a discrete value of wickets which will be taken by the bowler at each match of the tournament. The summation of the wickets of every match is the predicted number of wickets which will be taken by the bowler in the upcoming tournament. Here, the table shows the range of wickets with classes and the considered number of wicket for the prediction of each class:

Table 4.4: Class-wise Wicket for Bowlers

Range	Class	Considered Wickets for Estimation of Bowling Performance
0-1	1	1
2	2	2
3	3	3
4	4	4
≥ 5	5	5

Here, in the table, we also consider the upper bound of each range as the prediction of the discrete number of wickets for each match. The total number of predicted wickets were predicted by the use of machine learning algorithm, along with the predicted number of wickets, some traditional bowling features are taken into action to evaluate the bowling performance of the bowler for the future tournament. For batters and bowlers, we do not only focus on the predicted runs and wickets as we had the prediction in classes, so we have taken approximated runs and wickets for each player. As the prediction is only an approximation, to make it more feasible and practical we merge the predicted results with various cricketing features to make the objectives more robust. We do not pass the predicted runs and wickets directly to the optimization phase, rather we make two features namely batting performance and bowling performance with the help of some other features along with the predicted results, so the batting and bowling performance values are the indicator of what type or level of performance we expect from that particular player in the future tournament, which we consider as objectives in our multi-objective optimization phase where, the cost of the player is also considered as an objective to make the problem a three objective optimization problem under some specified constraints, which will be discussed later.

4.2.2.2 Estimation of Batters' Performance

For the estimation of batters' performance we have considered each type of batters' performances separately. These types are:

- Top-Order Batter

The batters who come to bat at number 1st to 3rd position are considered as top order batters. We make a proper dataset of top order batters, where the prediction of runs came from the first phase of our work, which is the machine learning section. As the prediction is one kind of approximation, we cannot solely depend on the approximation for forecasting the future performance of the batter. So, we focus on the batting average and batting strike rate of that batter too. Three features that we considered are:

- Predicted Run for the Batter 50
- The Strike Rate of the Batter 20
- The Batting Average of the Batter 30

For all types of batter, we considered our forecast of the runs that he/she would score in the upcoming tournament as 50% important. The other two features which are the batting strike rate and batting average vary from one position to another position's batter. As, top order batters come into play first, and their role is to put a strong foundation of an inning, so we have put more touch on batting average rather than the batting strike rate which is mentioned in [3]. So the estimation of batting performance of a top order batter is estimated by the following formula:

$$\text{Top-Order Batter Performance} = \text{Predicted Runs} * 50 + \text{Carreer Strike Rate} * 20 + \text{Career Batting Average} * 30$$

The values of the batting performance are normalized among 0 to 1 by dividing all the batting performance values of all top-order batters by the highest value of batting performance from top-order batters.

- Middle-Order Batter

As the middle order batters come into play at position 4th and 5th, their task is to play sensibly as well as fastly. So we need to focus on the batting strike rate and batting average equally for these positioned batters which is also mentioned in [3]. So, the performance estimation follows this rule:

$$\text{Middle-Order Batter Performance} = \text{Predicted Runs} * 50 + \text{Carreer Strike Rate} * 25$$

+ Career Batting Average*25

The values of the batting performance are normalized among 0 to 1 by dividing all the batting performance values of all middle-order batters by the highest value of batting performance from middle-order batters.

- Lower-Order Batter

The lower order batters come into bat at 6th and 7th position. In a twenty-twenty game where each team only plays twenty overs, the lower order batters, mostly cannot play as many balls as the other batters play. And, as they come into bat at last, their task is to focus on getting early runs where not to give much focus on the fact that they might get out early. Sometimes, the situation changes and lower-order batters also need to focus on not getting out early. But the case is rare, that is mentioned in [3]. So, the formula of estimation of batting performance for a lower order batter is:

$$\text{Lower-Order Batter Performance} = \text{Predicted Runs} * 50 + \text{Carreer Strike Rate} * 30 + \text{Career Batting Average} * 20$$

The values of the batting performance are normalized among 0 to 1 by dividing all the batting performance values of all lower-order batters by the highest value of batting performance from lower-order batters. The bowling performance of the batters are assigned to 0 as the batters are not significant at taking wickets, while they may bowl and take wickets in matches in real life but for our research work, we consider the bowling performance of the batters to be zero (0).

4.2.2.3 Estimation of Bowlers' Performance

In our research work, all the players performance estimation is done for the whole tournament comprising fourteen (14) matches. Bowlers' performance is evaluated by considering the predicted number of wickets for the tournament and two important features. One of these is the economy rate and the other one is the bowling average of the bowler. As we have mentioned earlier, the economy rate is the feature which refers to the number of runs given in a single over of six (6) balls. The other feature is bowling average, this feature indicates how many runs are scored against per wicket taken by the bowler. As the prediction of the number of wickets taken by the bowler is an approximation, we should not completely depend on this feature. So, we have taken into consideration the other two features to calculate the bowling performance's value of a bowler. The estimation is done by the formula which is given below:

$$\text{Bowler Performance} = \text{Predicted Wickets} * 50 + \text{Career Economy Rate} * 30 + \text{Career Bowling Average} * 20$$

The values of the bowling performance are normalized among 0 to 1 by dividing all the bowling performance values of all bowlers by the highest value of bowling performance from bowlers. The batting performance of the bowlers are assigned to 0 as the bowlers are not significant at scoring runs, while they score some runs in matches in real life but for our research work, we consider the batting performance of the bowlers to be zero (0).

4.2.2.4 Estimation of All-Rounders' Performance

Allrounders are those players who can bat and bowl as well. In previous sections, we have mentioned that, for batters the bowling performance is considered to be zero and for bowlers the batting performance is considered to be zero. As all rounders bats and bowls, we consider both the batting and bowling performance for this type of player. The batting performance is estimated by the batter's formula and the bowling performance is estimated by the above bowler's formula. The all-rounder's performance is estimated by the following formula :

$$\text{Allrounder Performance} = \text{Batting Performance} * 50 + \text{Bowling performance} * 50$$

The values of the batting and bowling performance are normalized among 0 to 1 by dividing all the batting and bowling performance values of all all-rounders by the highest value of batting and bowling performance respectively from all-rounders.

4.2.2.5 Estimation of Wicket-Keepers' Performance

Wicket-keepers are the players who stand behind the wicket to take catches and try to fall the ball from the stump to make out the batter while the batter's foot is out of his/her popping crease or safe margin. Traditionally a wicket-keeper can bat as well. As we predict runs for wicket-keepers as well, we try to bolster the performance value of wicket keepers by merging the fielding or keeping capability of the keepers. We take dismissal rate as a considerable feature, which indicates how many catches a wicket-keeper takes and how many run-out or stumping he/she can do in a match. So, the performance of the wicket-keepers are evaluated by the following formula:

$$\text{Wicket Keeper Performance} = \text{Batting Performance} * 70 + \text{Dismissal Rate} * 30 + \text{Career Bowling Average} * 20$$

The values of the wicket-keepers' batting performance are normalized among 0 to 1 by dividing all the batting performance values of all wicket-keepers by the highest value of batting performance respectively from wicket-keepers.

4.2.3 Problem Formulation

The methodology we followed in the first phase of our research was to predict runs for batters and wickets for bowlers match by match for an upcoming tournament of Indian Premier League. In this section, while we worked on the second phase of the research, we tried to make an optimal squad where the batting performance, bowling performance of the a team would be maximized while the cost should be minimized. From the first phase, we have obtained predictions of runs and wickets of players. In this phase, we made two features namely: Batting performance and bowling performance. The problem every team management faces at the beginning of the tournament is to make a team which would be champion in the tournament. The player selection is a tough process as it needs tremendous computation to think of every player, and make a proper combination where both the batting and bowling ability of the team balances. This is the issue, most of the team managers encounter that one team may be good in batting but there is a lack in bowling. While the opposite can also happen. Our approach is to automate the process. We neither rely on the predicted performance of a player nor make a team focusing on the previous performance of players. Firstly, we estimate how a player would perform in the upcoming tournament. Then the list of players along with their asking price are gone to the multi-objective optimization phase to make a squad for that tournament. In the IPL, an ideal squad is made of 23 players. We have predicted performance and estimated the performance of 152 players. From them we have categorised the players based on their capability and their playing position. So, the squad of 23 members from 22 top-order batters, 18 middle-order batters, 38 all-rounders where the all the lower-order batters are present with some middle and top-order batters, 15 wicket-keepers and 59 bowlers are made by using the following combination:

$$\text{Squad} = \binom{22}{4} C_{TO} \binom{18}{3} C_{MO} \binom{15}{3} C_{WK} \binom{38}{8} C_{AL} \binom{57}{8} C_B \quad (4.1)$$

Here,

$TO = \text{Top} - \text{orderBatter}$

$MO = \text{Middle} - \text{orderBatter}$

$WK = \text{Wicket} - \text{keeper}$

$AL = \text{All} - \text{rounder}$

$B = \text{Bowler}$

To solve the problem, we have used multi-objective optimization algorithms to form squads from where team management can observe and can choose squads according to their choice. Here, we are saying the team management can choose squads from the resultant squads after applying multi-objective optimization techniques because, traditionally some home-grounds of some teams help bowlers and some help batters to score runs. So, from the resultant squads the team managers can choose which type of squad they want to form, whether the squad would be good at batting or it would be good at bowling or they want a balanced squad where the batting and bowling capability of the team are the same. The resultant teams would be of various budgets, so the owner can choose any squad from the various budget ranges to serve their purpose.

4.2.3.1 Preparation of Dataset for Multi-Objective Optimization

In this part of our research work, we discuss the dataset which is used by various multi-objective optimization algorithms to make squads for an upcoming tournament. The dataset is made of players where the top 0 to 21 indexed players are top-order batters, then the 38 to 54 indexed players are middle-order batter, after them the 71 to 92 indexed players are lower-order batters. These three types of players have some value of batting performance, but the bowling performance values of these players are set to 0, and the base prices of the players are given in the dataset. The price which we are calling the base price, is the price, when the player was sold in an auction of any previous version of the tournament of Indian Premier League, the authority put an opening price of the player, where the bidding starts. After research, we have found that the average deflection rate of price is 2.5 crores for each player in an IPL auction. The bidding process is uncertain [36], so we take the base price of the players and try to make a squad of 23 players where the deflection of prices is considered, so that from the resultant squads the members of team management can bid as per their wish. The 55 to 92 indexed players are the all-rounders, who have both batting and bowling performance values along with their prices. The next type of players are the wicket-keepers who are indexed from 22 to 35, where the batting performance is available for this kind of player and the bowling performance is set to zero along with their respective prices. The final type of players are bowlers who are indexed from 93 to 151. The bowling

performance values are only available for them whereas the batting performance value is set to zero for them and the price of the players are attached with the respective player. Here is a glimpse of the dataset, which is used by the multi-objective optimization methods:

	A	B	C	D	E	F	G	H	I	J
1	Index	Foreign	Player_Name	Cost	Bowling Type	Order	Batting Perform	Bowling Perform	IndexRole	Performance
2	0	1	Faf du Plessis	1.5	-	Top	0.509743547	0	0	0.375010463
3	1	0	Murali Vijay	2	-	Top	0.438140827	0	0	0.322333447
4	2	0	Suresh Raina	2	-	Top	0.907263785	0	0	0.667459969
5	3	0	Prithvi Shaw	0.2	-	Top	0.422044621	0	0	0.310491716
6	4	0	Shikhar Dhawan	2	-	Top	0.554138599	0	0	0.407671218
7	5	0	Shreyas Iyer	2	-	Top	0.508182968	0	0	0.373862369
8	6	1	Chris Lynn	2	-	Top	0.689824886	0	0	0.507493526
9	7	0	Nitish Rana	0.2	-	Top	0.663704895	0	0	0.488277451
10	8	0	Shubman Gill	0.2	-	Top	0.683740205	0	0	0.503017112
11	9	1	Evin Lewis	1.5	-	Top	0.48566054	0	0	0.357292966
12	10	0	Rohit Sharma	2	-	Top	0.443989057	0	0	0.326635899
13	11	0	Suryakumar Yadav	0.3	-	Top	0.663783712	0	0	0.488335436
14	12	1	Chris Gayle	2	-	Top	0.718722742	0	0	0.528753232
15	13	0	Karun Nair	0.5	-	Top	0.74604776	0	0	0.548855827
16	14	0	Mayank Agarwal	0.2	-	Top	0.439953621	0	0	0.323667091
17	15	0	Virat Kohli	2	-	Top	0.360050583	0	0	0.58858487
18	16	0	Ajinkya Rahane	2	-	Top	0.484802309	0	0	0.356661579
19	17	0	Rahul Tripathi	0.2	-	Top	0.487144053	0	0	0.358384364
20	18	1	Steven Smith	2	-	Top	0.358101173	0	0	0.557723337
21	19	1	David Warner	2	-	Top	0.782065429	0	0	0.57535347
22	20	1	Kane Williamson	1.5	-	Top	0.581157105	0	0	0.501116773
23	21	1	Martin Gupthill	0.75	-	Top	0.595160279	0	0	0.437850235
24	22	1	Sam Billings	1	-	Lower	0.394075055	0	2	0.3218968236
25	23	1	Nicholas Pooran	0.5	-	Middle	0.521364692	0	1	0.4338846144
26	24	0	Sanju Samson	1	-	Top	0.619548045	0	0	0.5184074871
27	25	0	Robin Uthappa	2	-	Top	0.573392744	0	0	0.5094643168

Figure 4.5: Multi-objective Optimization Dataset

4.2.4 Objective Formulation

As, our problem is a multi-objective optimization problem, so we have introduced three objectives. These objectives are:

- Batting Performance
- Bowling Performance
- Cost

The first objective is the batting performance of the squad. We have previously mentioned how the batting performance of a player is estimated. Here, we are on the verge of making a squad of 23 players, from them except the bowlers, all the players have some value of batting performance. We need to maximize the value of the batting performance of the entire squad. The objective is:

$$f_1 = \max_{i=P_1, R_2, \dots, P_{23}} \text{Batting Performance (i)} \quad (4.2)$$

The second objective is the bowling performance. Here, the bowlers and the all-rounders have some value of bowling performance. The entire bowling performance of the squad is calculated by the summation of all the values of bowling performance of players. We need

to maximize the value of the bowling performance of the whole squad. So, the objective is to:

$$f_2 = \max_{i=P_1, R_2, \dots, P_{23}} \text{Bowling Performance (i)} \quad (4.3)$$

The Third objective is the cost of the squad. As we have said earlier, in Indian Premier League, the players are picked for a team after winning in an auction. So, the cost is important here, to form a squad. We know that in the IPL, no team can spend not more than 80 crores of Indian Rupees to form a squad in an auction. So, the cost is a huge factor in terms of forming a squad. We have studied the mega auctions of the IPL, which have taken in place in 2018. In that auction, the highest deviation from asking price of a player was 10.5 crores and least deviation from asking price of a player was 0 Rupee. The average deviation was 2.5 crores from the asking price of a player. In auction, there are some special players, who are chosen by various franchises but the highest bidder gets the player. So the process of bidding is uncertain, and our work does not cover the prediction of prices of players. So, we have taken the base prices of players and tried to minimize the cost of forming an optimal squad. So, the final and third objective is:

$$f_3 = \min_{i=P_1, R_2, \dots, P_{23}} \text{Cost of Player (i)} \quad (4.4)$$

4.2.5 Constraint Formulation

Constraint in a multi-objective optimization problem is that the term should be taken care of while generating a new population. As we are working with making an optimal squad for Indian Premier League, there is a rule of forming the squad that no more than 8 foreign players (the players who are not citizens of India) can present in a squad. So, a squad having more than 8 foreign players is infeasible. So, the constraint is:

$$g_1 \Rightarrow \sum_{P_i} \text{Number of Foreign Players (i)} = 8 \quad (4.5)$$

4.2.6 Repair Function Formulation

When we need to manipulate a new population at the time of taking the initial solution, generating offspring and the generation of offspring population, the process can be done by writing a block of code which is known as a repair function. In our problem, we needed to implement a repair function too. We have a total of 152 players and the players are indexed serially according to their playing type such as top-order batter, middle-order batter, wicket-keeper, all-rounder where all the lower-order batters are present with some top and middle-order batters and bowler. When multi-objective algorithms such as NSGA-II and

SPEA2 run there is a chance of picking the same player multiple times. As we worked to make an optimal squad, no duplication of players can happen. To solve the problem, we have considered the squad as an array of 23 sizes. Here, the array is formatted with the combination of players of their playing styles.

Table 4.5: Formation of Squad of 23 Players in Squad

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
TP	TP	TP	TP	WK	WK	WK	MD	MD	MD	AR	AR	AR	AR	AR	B	B	B	B	B	B	B	B

Here,

- TP= Top-order Batter
- WK= Wicker-keeper
- MD= Middle-order Batter
- AR= All-rounder
- B= Bowler

So, we can divide the array according to various types of players into five distinct divisions. These are:

- Top-order batters

Table 4.6: Position of Top-order Batters

1	2	3	4
Top-Order Batter	Top-Order Batter	Top-Order Batter	Top-Order Batter

As we have total 22 top-order batters in our list of 152 players, four players have to be chosen among them. If there happens to be any duplicacy player's index, the last duplicate elected player should be omitted from the divided array of only 4 players and a player from the list of to-order batters has to be chosen while keeping in focus that the player did not appear before in the generation of new population.

- Wicket-Keepers

Table 4.7: Position of Wicket-keepers

5	6	7
Wicket-Keeper	Wicket-Keeper	Wicket-Keeper

There are 15 wicket-keepers in our full list of players. If the last elected wicket-keeper's index appears before, the player has to be unpicked and a new wicket-keeper would be selected randomly from 22 to 36 indexed players who are wicket-keepers.

- Middle-Order Batters

Table 4.8: Position of Middle-order Batters

8	9	10
Middle-order batter	Middle-order batter	Middle-order batter

The same procedure continues for middle-order batters, if any duplicates can be found the duplicacy would be removed by picking 38 to 54 indexed players from the whole list of players, the process continues until no duplicacy is found.

- All-rounders

Table 4.9: Position of All-rounders

11	12	13	14	15
All-Rounder	All-Rounder	All-Rounder	All-Rounder	All-Rounder

If any duplicate can be found in the sub-array of all-rounders, the duplicacy is removed from 55 to 92 indexed players who serve the purpose of all-rounder.

- Bowlers

Table 4.10: Position of Bowlers

16	17	18	19	20	21	22	23
Bowler	Bowler	Bowler	Bowler	Bowler	Bowler	Bowler	Bowler

Here is the sub-array of bowlers, where the duplicacy is solved by electing random players from 93 to 151 indexed players who are bowlers while keeping focus on the fact that the duplicacy solving proceeds until no duplicate can be found. We, have written the repair function times and the places are:

- At the Creation of Initial Solution
- At the Creation of Offspring
- At the Creation of Offspring Population

4.2.7 Selection of Multi-objective Optimization Algorithm

In this phase of implementing multi-objective optimization to our problem, we try to make an optimal squad of 23 players (standard size of a squad of the participating team in IPL) where we have considered some objectives as well as some constraints. Before diving into the solution of the problem domain, we need to specify the multi-objective optimization algorithms, which we will be going to use for our research work. In An Improved Spea2 Multi Objective Algorithm With Non Dominated Elitism And Generational Crossover [26] we have found a table where the differences among various multi-objective algorithms are shown. The table is shown here:

Table 4.11: Comparison of different MOEAs

Algorithm	Advantages	Disadvantages
VEGA(Vector Evaluated Genetic Algorithm)[[37]].	First MOEA.	Dominance of objective weight.
WBGA (Weight Based Genetic Algorithm) [[38]].	Simple extension of single objective optimization.	Difficulties to find solutions in Non-convex Pareto-front.
MOGA (MultiObjective Genetic Algorithm)[[39]].	Simple extension of Single objective optimization.	Slow convergence rate for large number of objectives. Niche count calculation is computationally expensive.
PAES (Pareto Archived Evolution Strategy) [[40]].	Local search with Random mutation hill climbing strategy.	Performance depends on cell sizes.
PESA (Pareto Envelope based Selection Algorithm)[[41]].	Easy to implement and Computationally efficient .	Performance depends on Cell sizes, Prior information needed about objective space.
PESAll(Region based Selection in Evolutionary Multi-objective Optimization) [[42]].	Extension of PESA, Region based selection.	Performance depends on Cell sizes, Prior information needed about objective space.
NSGA (Non-dominated Sorting Genetic Algorithm) [[43]]	Fast Convergence.	Uses user-defined niche size parameters.
NSGA II (Non-dominated Sorting Genetic Algorithm)[[20]].	Timely, efficient, Well tested, better than NSGA.	Slow convergence rate in the later stage of the algorithm and when the number of objectives increases.
SPEA(Strength Pareto Evolutionary Algorithm)[[44]]	No user-defined parameter for clustering.	Extreme points are not preserved.
SPEA2 (Improved SPEA) [[44]].	Extreme points are preserved, Well spread solutions.	Fitness and density estimation is computationally expensive.
IBEA(indicator Based Evolutionary Algorithm) [[45]]	Well tested,Preference based convergence and diversified solutions.	Difficult to implement, Reference point selection is critical, a problem of local optimal solution.

For our research work, we have chosen NSGA-II and SPEA2 as two multi-objective optimization algorithms. A research paper of Faez Ahmed, Abhilash Jindal, Kalyanmoy Deb [5] used NSGA-II for predicting the team in an IPL tournament before. In our research work, we implemented both of the algorithms and lastly evaluated the performances of the algorithms

to specify the best one which served our purpose best.

Chapter 5

Result Analysis

We divide the section into two parts. The first one is the result analysis of our machine learning phase, which is the first phase of our research. The latter part is the result analysis of the multi-objective optimization phase, which is the last part of our research. In the first section, we will talk about the outcomes of machine learning approaches that have been obtained as a result of putting our recommended approaches into action. A comparison between our modelling accuracies and the other modelling accuracies that have been previously implemented will also be presented. We will try to predict runs and wickets for batters and bowlers respectively using the techniques given in the first section of Chapter 3 and see how far we can get. We separated the data into two groups: the training set and the testing set. The training set is then utilized to train our model, which is the last step. After that, we utilized the testing set to assess the accuracy of the models that we had put in place. This investigation was conducted using five distinct classifiers. These models are trained and tested on our processed dataset, and a comparison of the accuracies of the models is presented in this section. In the second section, we will discuss the results of our multi-objective optimization part. The results of the first phase have entered into the second phase, where with the results of predicted runs and wickets of players, we tried to make a squad, where there were three objectives, such as batting performance, bowling performance and cost. We tried to make a squad of 23 players, where the predicted performance for the upcoming tournament came from the machine learning portion. In this phase, we implement two algorithms of multi-objective optimizations and we will discover the analysis of the results here in the second phase of the result analysis chapter. We will also show the comparison between our work and the existing works for the justification of our claim that our model is performing better with the model, which we have proposed.

5.1 Result Analysis of Machine Learning Phase

The output of the machine learning phase is runs for batters and wickets for bowlers respectively one by one for 14 matches. The output of runs or wickets are classified into five classes from 1 to 5. As we have balanced data for the implementation of the machine learning phase, we use accuracy as an evaluation metric to evaluate the performance of our proposed model. For every machine learning classifier algorithm, we have split the data into training and test samples in accordance to four setups. These four setups of training-testing samples ratio are given hereby:

- 60% training samples-40% testing samples
- 70% training samples-30% testing samples
- 80% training samples-20% testing samples
- 90% training samples-10% testing samples

For prediction, we have divided our players into two main types and these are 'batters' and 'bowlers'. The forecast of classes of runs is done for batters of three kinds. These are:

- Top-order Batter
- Middle-order Batter
- Lower-order Batter

For the prediction of the class of wickets, we consider the players who balls and the result is evaluated for each machine learning classification algorithm. In the following sections, the accuracy scores of every model of every set-up are discussed.

5.1.1 Prediction of Performance Using Naive Bayes Classifier Algorithm

In this section, we used Naive Bayes Classifier for the prediction of runs for batters and wickets for bowlers. The evaluation of the model for both batter and bowler is shown in the following section.

We have selected 10 random players, and averaged the accuracy values to achieve the final average value. Here, in the following table the training and testing accuracies for predicting the class of runs for top-order batters is shown for all the variations:

Table 5.1: Naive Bayes Classifier Accuracy for Top Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	85.567	Training Accuracy	86.900	Training Accuracy	87.024	Training Accuracy	86.432
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	88.449	Accuracy	89.892	Accuracy	91.039	Accuracy	90.322

In the above-mentioned table, we can see that when we used Naive Bayes Classifier for prediction of runs of top-order batters we achieved highest accuracy of 91.039% when the training sample was about 80% and the testing sample was 20%.

Here, in the following table the training and testing accuracies for predicting the class of runs for middle-order batters is shown for all the variations:

Table 5.2: Naive Bayes Classifier Accuracy for Middle Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	95.649	Training Accuracy	94.825	Training Accuracy	96.857	Training Accuracy	94.268
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	94.295	Accuracy	93.383	Accuracy	95.169	Accuracy	92.580

For middle-order batters, 95.169% accuracy is achieved for 80:20 ratio of training: testing samples.

Here, in the following table the training and testing accuracies for predicting the class of runs for lower-order batters is shown for all the variations:

Table 5.3: Naive Bayes Classifier Accuracy for Lower Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	91.214	Training Accuracy	90.233	Training Accuracy	89.954	Training Accuracy	92.258
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	89.920	Accuracy	88.692	Accuracy	86.979	Accuracy	90.476

The highest accuracy is 90.476% for 90:10 ratio of training: testing samples. As we consider the ideal ratio is 90:10 for training: testing samples, we got on average 91.127% accuracy for all kinds of batters for Naive Bayes Classifier.

Here, in the following table the training and testing accuracies for predicting the class of wickets for bowlers is shown for all the variations:

Table 5.4: Naive Bayes Classifier Accuracy for Bowlers

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	93.159	Training Accuracy	92.425	Training Accuracy	94.370	Training Accuracy	94.541
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	91.447	Accuracy	90.906	Accuracy	91.430	Accuracy	92.145

For bowler, at the time of prediction of class of wickets with the Naïve Bayes Classifier, we have got 92.145% accuracy.

5.1.2 Prediction of Performance Using Support Vector Machine Classifier Algorithm

This section explains the accuracy score of using Support Vector Machine Classifier for prediction. The evaluation of the model for both batter and bowler is shown in the following

section.

5.1.2.1 Model Evaluation of Support Vector Machine Classifier

We have selected 10 random players, and averaged the accuracy values to achieve the final average value. Here, in the following table the training and testing accuracies for predicting the class of runs for top-order batters is shown for all the variations:

Table 5.5: Support Vector Machine Classifier Accuracy for Top Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	92.559	Training Accuracy	93.762	Training Accuracy	93.950	Training Accuracy	95.321
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	91.955	Accuracy	91.949	Accuracy	91.693	Accuracy	93.670

In the above-mentioned table, we can see that for top-order batters, the testing accuracy is 93.670% when the training: testing samples is in 90:10 ratio.

Here, in the following table the training and testing accuracies for predicting the class of runs for middle-order batters is shown for all the variations:

Table 5.6: Support Vector Machine Classifier Accuracy for Middle Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	97.223	Training Accuracy	96.249	Training Accuracy	97.150	Training Accuracy	95.991
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	95.410	Accuracy	95.090	Accuracy	95.833	Accuracy	93.610

The model gives 95.833% accuracy for the prediction of class of runs for middle-order batters.

Here, in the following table the training and testing accuracies for predicting the class of runs for lower-order batters is shown for all the variations:

Table 5.7: Support Vector Machine Classifier Accuracy for Lower Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	92.873	Training Accuracy	92.209	Training Accuracy	93.050	Training Accuracy	92.891
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	91.122	Accuracy	90.311	Accuracy	89.393	Accuracy	90.476

For lower-order batter, SVM classifier provides 91.122% accuracy at the time of predicting the class of runs.

Here, in the following table the training and testing accuracies for predicting the class of wickets for bowlers is shown for all the variations:

Table 5.8: Support Vector Machine Classifier Accuracy for Bowlers

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	91.143	Training Accuracy	93.110	Training Accuracy	94.904	Training Accuracy	93.201
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	89.583	Accuracy	89.317	Accuracy	90.521	Accuracy	91.233

SVM classifier shows 91.233% accuracy for bowlers in predicting the class of wickets.

5.1.3 Prediction of Performance Using K-Nearest Neighbors Classifier Algorithm

Using K-Nearest Neighbors Classifier, the result we have got is explained in the following part of the section. The evaluation of the model for both batter and bowler is shown in the

following section.

5.1.3.1 Model Evaluation of Support Vector Machine Classifier

We have selected 10 random players, and averaged the accuracy values to achieve the final average value. Here, in the following table the training and testing accuracies for predicting the class of runs for top-order batters is shown for all the variations:

Table 5.9: K-Nearest Neighbors Classifier Accuracy for Top Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	92.273	Training Accuracy	93.110	Training Accuracy	94.256	Training Accuracy	96.028
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	90.536	Accuracy	90.756	Accuracy	90.476	Accuracy	93.630

Here, we have obtained 93.630% of accuracy for the prediction of class of runs of top-order batters.

Here, in the following table the training and testing accuracies for predicting the class of runs for middle-order batters is shown for all the variations:

Table 5.10: K-Nearest Neighbors Classifier Accuracy for Middle Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	95.343	Training Accuracy	95.278	Training Accuracy	97.100	Training Accuracy	95.348
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	94.838	Accuracy	94.893	Accuracy	96.202	Accuracy	94.392

K-NN classifier gives 96.202% accuracy for middle-order batters for the prediction of class of runs.

Here, in the following table the training and testing accuracies for predicting the class of runs for lower-order batters is shown for all the variations:

Table 5.11: K-Nearest Neighbors Classifier Accuracy for Lower Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	90.371	Training Accuracy	91.965	Training Accuracy	88.201	Training Accuracy	89.367
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	88.976	Accuracy	88.889	Accuracy	86.528	Accuracy	88.235

For lower-order batters, 88.976% accuracy is shown for the setup of 60: 40 ratio of training: testing samples.

Here, in the following table the training and testing accuracies for predicting the class of wickets for bowlers is shown for all the variations:

Table 5.12: K-Nearest Neighbors Classifier Accuracy for Bowlers

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	93.991	Training Accuracy	93.253	Training Accuracy	93.990	Training Accuracy	90.279
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	92.093	Accuracy	91.761	Accuracy	91.965	Accuracy	89.430

For bowler, the model gives moderate accuracy of 92.093% for 60:40 ratio of training: testing samples. But, for the ideal 90:10 ratio, the model gives accuracy of only 89.430

5.1.4 Prediction of Performance Using Decision Tree Classifier Algorithm

In this section, we used the Support Vector Machine Classifier for the prediction of runs for batters and wickets for bowlers. The evaluation of the model for both batter and bowler is shown in the following section.

5.1.4.1 Model Evaluation of Decision Tree Classifier

We have selected 10 random players, and averaged the accuracy values to achieve the final average value. Here, in the following table the training and testing accuracies for predicting the class of runs for top-order batters is shown for all the variations:

Table 5.13: Decision Tree Classifier Accuracy for Top Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	91.101	Training Accuracy	90.209	Training Accuracy	92.428	Training Accuracy	93.001
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	90.230	Accuracy	88.912	Accuracy	89.102	Accuracy	91.776

For an ideal ratio of splitting, the decision tree classifier gives accuracy of 91.776% for top-order batters.

Here, in the following table the training and testing accuracies for predicting the class of runs for middle-order batters is shown for all the variations:

Table 5.14: Decision Tree Classifier Accuracy for Middle Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	95.456	Training Accuracy	94.273	Training Accuracy	94.665	Training Accuracy	94.527
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	93.964	Accuracy	93.709	Accuracy	92.868	Accuracy	92.186

For middle-order batters, the accuracy is high for less training examples but with the increasing training examples the accuracy scores lowers which gives a bad indication to use the model for prediction.

Here, in the following table the training and testing accuracies for predicting the class of runs for lower-order batters is shown for all the variations:

Table 5.15: Decision Tree Classifier Accuracy for Lower Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	92.378	Training Accuracy	90.230	Training Accuracy	92.559	Training Accuracy	96.396
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	89.920	Accuracy	88.692	Accuracy	91.666	Accuracy	95.614

For lower-order batters, the algorithm acts well and give 95.614% accuracy when training data is 90%.

Here, in the following table the training and testing accuracies for predicting the class of wickets for bowlers is shown for all the variations:

Table 5.16: Decision Tree Classifier Accuracy for Bowlers

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	87.458	Training Accuracy	90.130	Training Accuracy	89.331	Training Accuracy	90.550
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	86.447	Accuracy	87.832	Accuracy	88.888	Accuracy	87.879

When it is the time for predict the class of wickets for bowler, 80% training examples give better accuracy when the training examples are increased to 90%.

5.1.5 Prediction of Performance Using Random Forest Classifier Algorithm

Random Forest Classifier is the last classifier which we have used in our thesis work and been amazed by the result of the accuracy score. The evaluation of the model for both batter and bowler is shown in the following section.

5.1.5.1 Model Evaluation of Random Forest Classifier Algorithm

We have selected 10 random players, and averaged the accuracy values to achieve the final average value. Here, in the following table the training and testing accuracies for predicting the class of runs for top-order batters is shown for all the variations:

Table 5.17: Random Forest Classifier Accuracy for Top Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	90.097	Training Accuracy	91.138	Training Accuracy	92.009	Training Accuracy	90.112
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	86.447	Accuracy	87.832	Accuracy	88.888	Accuracy	87.879

Here, we can see a smooth transition in the accuracy when the training examples are going bigger, the accuracy score goes bigger as well. For top-order batters, we have achieved 94.620% of accuracy.

Here, in the following table the training and testing accuracies for predicting the class of runs for middle-order batters is shown for all the variations:

Table 5.18: Random Forest Classifier Accuracy for Middle Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	93.237	Training Accuracy	93.156	Training Accuracy	95.366	Training Accuracy	96.115
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	91.728	Accuracy	92.753	Accuracy	94.753	Accuracy	95.833

For middle-order batters, the highest accuracy is 95.833% when there are 90% training examples.

Here, in the following table the training and testing accuracies for predicting the class of runs for lower-order batters is shown for all the variations:

Table 5.19: Random Forest Classifier Accuracy for Lower Order Batters

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	91.166	Training Accuracy	89.995	Training Accuracy	94.025	Training Accuracy	94.002
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	90.526	Accuracy	88.692	Accuracy	92.592	Accuracy	93.650

For lower-order batters, the final accuracy is 92.592% when there are 80% training samples. If we increase the training examples, we can see a higher rate of accuracy which may have an over-fitting issue.

Here, in the following table the training and testing accuracies for predicting the class of wickets for bowlers is shown for all the variations:

Table 5.20: Random Forest Classifier Accuracy for Lower Order Bowlers

60% Training Samples- 40% Testing Samples		70% Training Samples- 30% Testing Samples		80% Training Samples- 20% Testing Samples		90% Training Samples- 10% Testing Samples	
Training Accuracy	93.025	Training Accuracy	95.658	Training Accuracy	95.002	Training Accuracy	94.958
Testing Accuracy		Testing Accuracy		Testing Accuracy		Testing Accuracy	
Accuracy	92.594	Accuracy	93.350	Accuracy	93.727	Accuracy	94.726

Lastly, for bowlers, the random forest classifier gives a decent amount of accuracy for the prediction of class of wickets and this is 93.727% when the training samples are made of 80

5.1.6 Comparison of results among all the Classifiers

We find that the random forest classifier is the most accurate predictor of runs for top-order, middle-order, and lower-order batters when predicting runs for both top-order and middle-order batters. Additionally, the random forest classifier model has the best accuracy when

predicting the number of wickets that would be taken by the bowlers. For predicting the class of runs for top-order, middle-order and lower-order batters and the class of the wickets of bowlers we have used Naive Bayes Classifier, Support Vector Machine Classifier, K Nearest Neighbor Classifier, Decision Tree Classifier and Random Forest Classifier. The accuracies of respective classifiers have been shown in the previous sections. In this section, we will show the table containing all classifiers where the accuracy values are given to visualize the comparison between these classifiers. When we consider 80% training samples and 20% testing samples, we can address the set-up as an ideal set-up for the good approximation of accuracy as an evaluation metric. Here, the table shows the accuracy of the prediction of top-order batters, middle-order batter, lower-order batter and bowler for all the classification model.

Table 5.21: Comparison of All Classifiers for Prediction Class of Runs and Wickets

Classifier	Top-order Batter	Middle-order Batter	Lower-order Batter	Bowler
	Accuracy			
Naive Bayes	91.039	94.169	86.393	92.145
SVM	91.693	93.833	89.393	91.233
K-Nearest Neighbor	90.476	93.202	86.528	89.430
Decision Tree	89.102	90.868	85.666	87.879
Random Forest	92.332	95.965	90.979	94.726

Here in the table, it is clear that, the random forest classifier gives the highest accuracy for top-order batter which is 92.332%, For middle-order batter the same model gives the accuracy of 95.965% Which is highest among all other models. For lower-order batter and bowlers, the model gives 90.979% and 94.726% accuracy respectively which is larger than the other accuracy values of other models.



Figure 5.1: Accuracy Comparison for 80:20 Split

From the graph we can see that the spike of random-forest model is higher for all types of players' performance prediction. By taking the average of the accuracy of four types of players, we can say that, the model we develop gives an accuracy of 93.501%

5.2 Result Analysis of Multiobjective Optimization Phase

This is the second part of our research work. In this part, we pass three objectives and one constraint to NSGA-II and SPEA2 algorithm to make feasible squads of 23 players for the tournament. The setting of parameters are described in the following section.

5.2.1 General Experimental Settings

In this part, we will go through all of the experimental conditions that were utilized during the study of implementing two multi-objective optimization algorithms. Certain experimental conditions associated with a specific phase will be discussed in more detail in the next

section. For both methods, we also use Integer Single Point Crossover [46] and Modified Simple Random Mutation [47], which is based on simple random mutation. The algorithms are performed separately 50 times for each part of the trials in order to enable statistical analysis of the findings. The general parameter settings for the NSGA-II and SPEA2 algorithms are:

Table 5.22: General parameter settings for NSGA-II and SPEA2

	NSGA-II	SPEA2
Population size	300	300
Offspring Population size	300	300
Crossover probability	0.9	0.9
Mutation probability	1	1
Maximum evaluations	6,00,000	6,00,000

The result analysis of the two algorithms distinctively are described in the following sections and the comparison between the results of the two algorithms is also shown in one section.

5.2.1.1 Squad Selection using NSGA-II

First and foremost, we employ the NSGA-II method, which is a well-known multi-objective optimization technique, in the formulation of optimum squads. We have obtained a total of 7296 solutions after running the NSGA-II algorithm 50 times with the previously specified parameter settings. We must concentrate our efforts on the Pareto optimum solutions among all of the alternatives available for selecting the optimal answers. A Pareto optimum solution is a type of solution in which we are unable to achieve a higher outcome from one aim without decreasing the importance of the other objective (s). Pareto optimum solutions provide us with the greatest possible perspective of the system's best meet of all of its objectives. We have 381 Pareto optimum options out of a total of 7296 possible solutions. The answer entails putting together the most optimum squads of 23 players, with three objectives and one constraint being implemented. Due to the fact that we employed three objectives for our study, the solution graph is plotted in three dimensions. The graph of the pareto-optimal solution will be examined further.

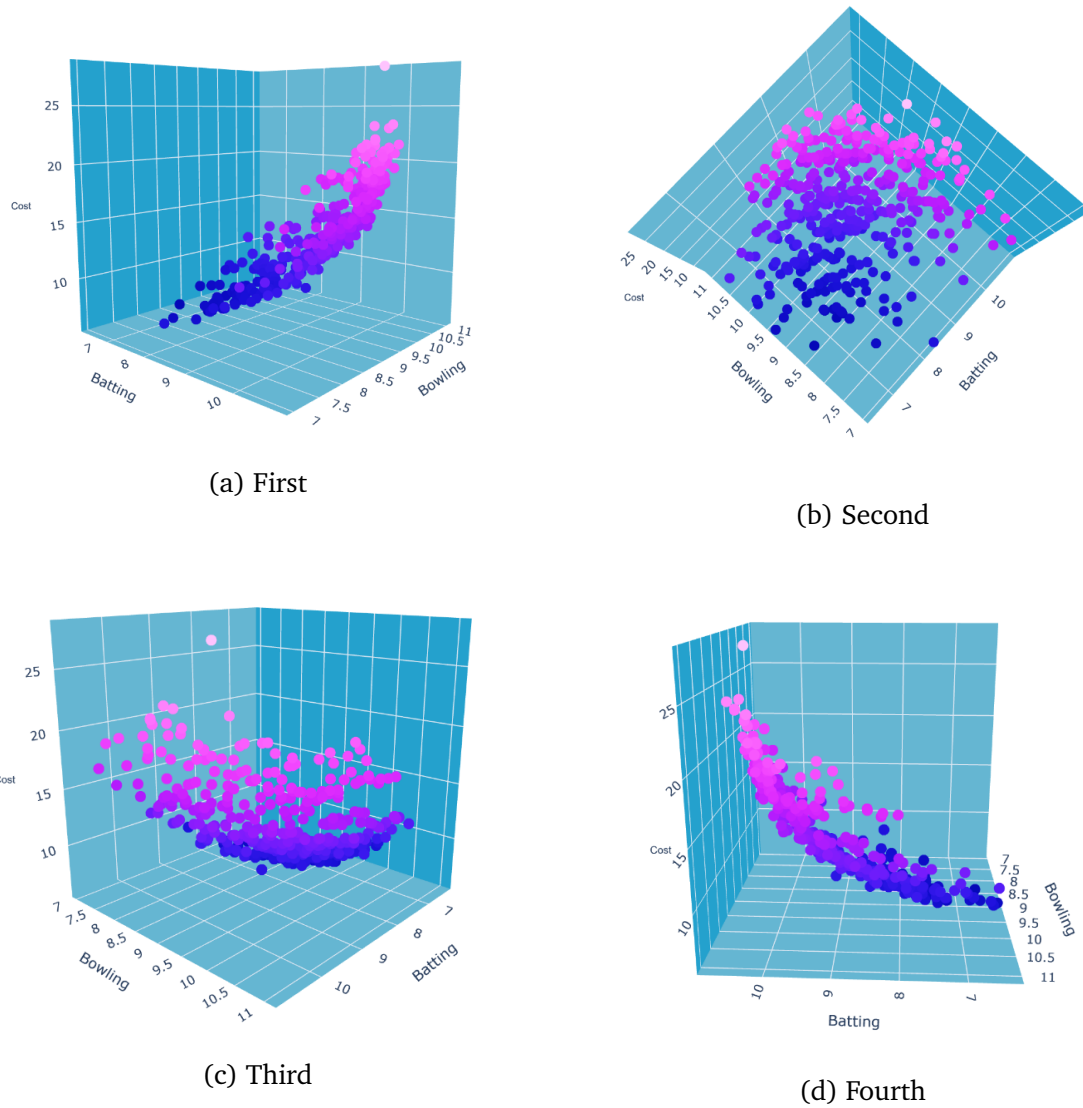


Figure 5.2: Different Views of 3D True Pareto Front for NSGA II.

There are four photos of a same graph taken from four different perspectives to make it easier to grasp. On three-dimensional space, each dot represents a squad of 23 players, and there are a total of 381 dots, also known as squads, in the graph, which represents the total number of squads in the league. The top most dot in picture Figure 5.2a defines the team with greatest budget. In picture Figure 5.2b the rightmost dot indicates a squad whose batting capability is more than 10 and the left-most squad represents bowling performance of greater than 10.5 while having moderate cost and batting performance. Due to the fact that there are two forms of performance, namely batting performance and bowling performance, it is reasonable to wonder why the performance of a team of 23 players falls behind 15 in both batting and bowling performance. This is due to the fact that we have assumed that batsmen who do not bowl and wicket-keepers have bowling performances of zero in our calculations. Because there are eight specialized bowlers in the roster, everyone else (the remaining 15 players) has some batting performance worth aside from them. As

a result, there is no possibility that the batting performance value will be more than 15. Same for the bowling performance, only the bowlers and the all-rounders have some bowling performance values. The squad consists of 8 bowlers and 5 all-rounders, which means that no more than 13 bowling performances may be achieved by the team as a whole. Here are several squads with better batting and bowling performance, as well as lower prices, to consider:

Table 5.23: Batting Oriented Team Formation

Team 1	Team 2	
Suryakumar Yadav	Suresh Raina	Player's Name
Karun Nair	David Warner	
David Warner	Shubman Gill	
Suresh Raina	Karun Nair	
Wriddhiman Saha	Wriddhiman Saha	
Lokesh Rahul	Quinton de Kock	
Jos Buttler	Jos Buttler	
Liam Livingstone	Riyan Parag	
Gurkeerat Singh Mann	Gurkeerat Singh Mann	
Riyan Parag	Manish Pandey	
Moeen Ali	Kieron Pollard	
Hardik Pandya	Shane Watson	
Shane Watson	Krishnappa Gowtham	
Kieron Pollard	Moises Henriques	
Kedar Jadhav	Hardik Pandya	
Rahul Chahar	Mohammed Shami	
Oshane Thomas	KM Asif	
Shardul Thakur	Billy Stanlake	
Sandeep Warrier	K Khaleel Ahmed	
KM Asif	Kulwant Khejroliya	
Jasprit Bumrah	Sandeep Warrier	
Billy Stanlake	Hardus Viljoen	
K Khaleel Ahmed	Yarra Prithviraj	
Batting Performance: 10.8296 Bowling Performance: 9.3461 Cost: 23.55 cr	Batting Performance: 10.7119 Bowling Performance: 8.5790 Cost: 20.85 cr	

These two teams had batting performances of 10.830 and 10.712 respectively, which is the greatest batting performance we have obtained using the NSGA-II method at the time of executing the algorithm.

Table 5.24: Bowling Oriented Team Formation

Team 1	Team 2	
Nitish Rana	Nitish Rana	Player's Name
Suryakumar Yadav	Suryakumar Yadav	
Shreyas Iyer	Suresh Raina	
Suresh Raina	Karun Nair	
Lokesh Rahul	Lokesh Rahul	
Quinton de Kock	Jos Buttler	
Jos Buttler	Jonny Bairstow	
Liam Livingstone	Riyan Parag	
Manish Pandey	Manish Pandey	
Riyan Parag	Gurkeerat Singh Mann	
Kedar Jadhav	Hardik Pandya	
Keemo Paul	Kedar Jadhav	
Dwayne Bravo	Sam Curran	
Sam Curran	Dwayne Bravo	
Shreyas Gopal	Shreyas Gopal	
Sandeep Warrier	K Khaleel Ahmed	
K Khaleel Ahmed	Sandeep Warrier	
Shardul Thakur	Billy Stanlake	
Billy Stanlake	Dhawal Kulkarni	
KM Asif	Shardul Thakur	
Umesh Yadav	KM Asif	
Kagiso Rabada	Mohammed Shami	
Ankit Rajpoot	Kagiso Rabada	
Batting Performance: 9.1612 Bowling Performance: 11.0580 Cost: 21.25 cr	Batting Performance: 9.6661 Bowling Performance: 11.0058 Cost: 20.95 cr	

Both of these squads have the best bowling performances, with 11.058 and 11.005 respectively, but their batting performance statistics are not as impressive.

Table 5.25: Cost-Effective Team Formation

Team 1	Team 2	
Mayank Agarwal	Nitish Rana	Player's Name
Suryakumar Yadav	Rahul Tripathi	
Shubman Gill	Karun Nair	
Rahul Tripathi	Suryakumar Yadav	
Shreevats Goswami	Heinrich Klaasen	
Heinrich Klaasen	Shreevats Goswami	
Nicholas Pooran	Nicholas Pooran	
Riyan Parag	Riyan Parag	
Nikhil Naik	Abhishek Sharma	
Akshdeep Nath	Gurkeerat Singh Mann	
Sherfane Rutherford	Deepak Hooda	
Stuart Binny	Sam Curran	
Mahipal Lomror	Shreyas Gopal	
Shreyas Gopal	Shivam Dube	
Sam Curran	Mahipal Lomror	
Sandeep Lamichhane	Nathu Singh	
Ankit Rajpoot	Navdeep Saini	
Prasidh Krishna	Deepak Chahar	
Sandeep Warrier	Jagadeesha Suchith	
Navdeep Saini	Mitchell_McClenaghan	
Avesh Khan	KM Asif	
KM Asif	Sandeep Warrier	
Mayank Markande	K Khaleel Ahmed	
Batting Performance: 6.6451 Bowling Performance:8.6178 Cost:5.9 cr	Batting Performance: 7.3982 Bowling Performance :8.6605 Cost :6 cr	

According to the base prices of the players in these two squads, they are the least expensive

squads in the league. These two squads are estimated to have cost around 5.9 crores Indian Rupee and 6 crores Indian Rupee, respectively, to purchase.

5.2.2 Squad Selection using SPEA2

Following the formation of squads with the help of the NSGA-II algorithm, we utilize the SPEA2 method to construct the most optimum squads. We have obtained 5158 solutions using the SPEA2 method, and we have obtained a total of 344 genuine pareto optimum solutions from the total number of solutions obtained using the SPEA2 technique. The depiction of the pareto graph is presented in the following part of this section, along with a more detailed description.

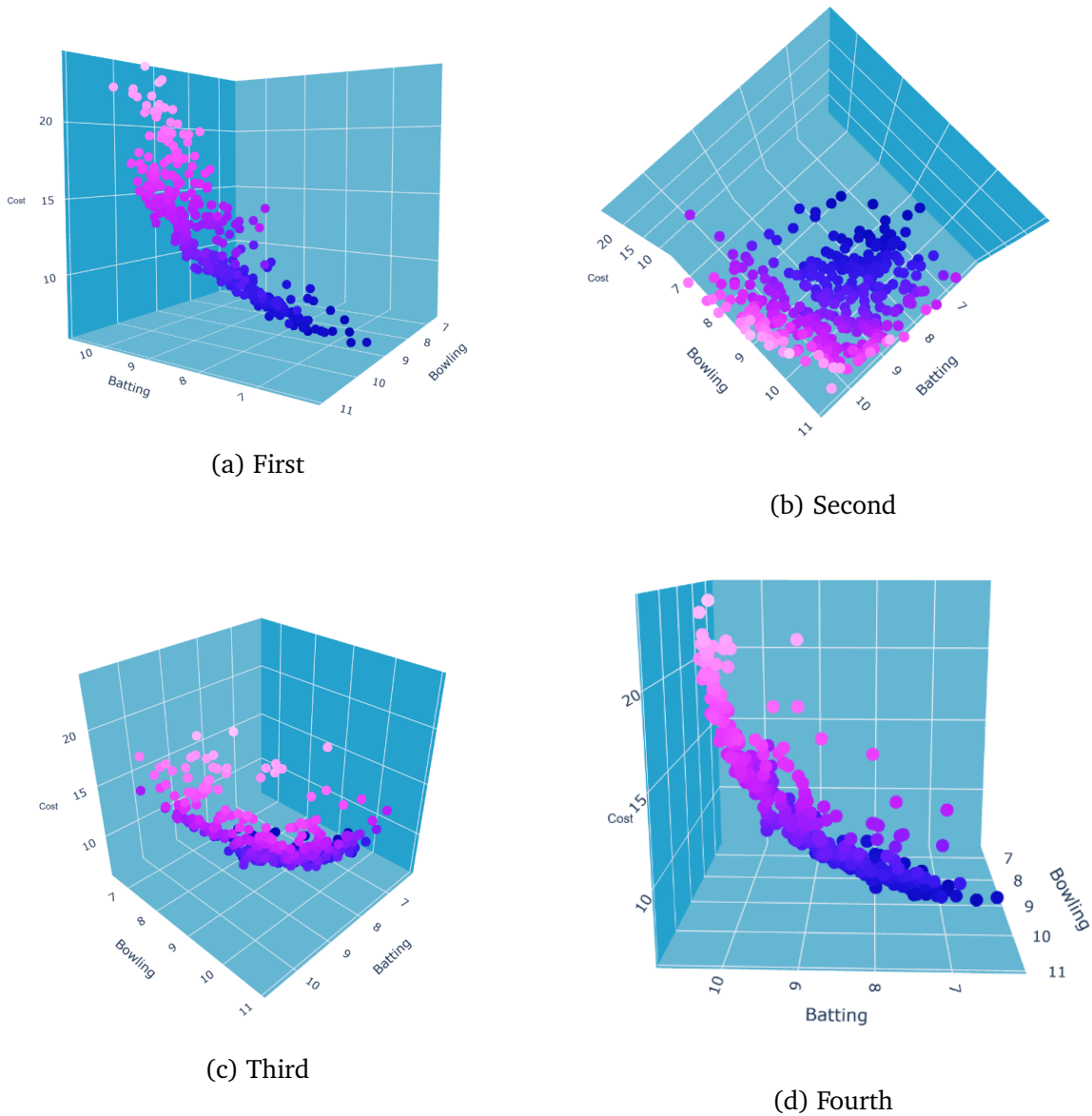


Figure 5.3: Different Views of 3D True Pareto Front for SPEA2.

The four pictures of the pareto optimal front are of a single front. The perspective varies as the visualization needs to be clearer. The shape here we have got is kind of a partial umbrella. From the front viewpoint, the front looks like a curve. The result we have got with NSGA-II method differs from the result of the SPEA2 method. In this method, we have got some uneven squads which breaks the symmetry of the figure. In image Figure 5.3a the top-most dot represents a squad which has cost of greater than 20 crores and the right-most bottom point represents a squad have cost of less than 10 crores. The curve looks hollow from the above. In image Figure 5.3d we can see some points that are not in the alignment of the curve which makes the algorithm not suited for our particular problem. We will discuss the comparison of the two methods in great detail in the upcoming section. Here are some squads which have the best batting performance, bowling performance and least cost.

Table 5.26: Batting Oriented Team Formation

Team 1	Team 2	
Suresh Raina	David Warner	Player's Name
Suryakumar Yadav	Suresh Raina	
Shubman Gill	Shubman Gill	
Karun Nair	Karun Nair	
Quinton de Kock	Jos Buttler	
Jos Buttler	Jonny Bairstow	
Lokesh Rahul	Wriddhiman Saha	
Gurkeerat Singh Mann	Riyan Parag	
Liam Livingstone	Gurkeerat Singh Mann	
Riyan Parag	Liam Livingstone	
Shane Watson	Shane Watson	
Kieron Pollard	Hardik Pandya	
Moises Henriques	Moeen Ali	
Moeen Ali	Andre Russell	
Krishnappa Gowtham	Kedar Jadhav	
Siddarth Kaul	Yarra Prithviraj	
K Khaleel Ahmed	Sandeep Lamichhane	
Sandeep Warriar	K Khaleel Ahmed	
Bhuvneshwar Kumar	Mayank Markande	
Yarra Prithviraj	Barinder Sran	
Dhawal Kulkarni	Navdeep Saini	
KC Cariappa	Sandeep Warriar	
Ankit Rajpoot	Avesh Khan	
Batting Performance: 10.7171 Bowling Performance:7.8246 Cost:19.8 cr	Batting Performance: 10.6778 Bowling Performance :8.5428 Cost :20.3 cr	

Working with the SPEA2 algorithms, the table depicts two squads that have the top two batting performance values in the sport of cricket. The maximum batting performance value for the above-mentioned squad was 10.717. Unlike the team in the first column, the squad in the second column has a less good batting performance value of 10.677, and this squad has a better bowling performance value than the squad in the first column. Pareto optimum

solutions are designed in such a manner that if we want to improve the batting performance, we must make a trade-off with the cost of or reduce the value of the bowling performance in order to do it.

Table 5.27: Bowling Oriented Team Formation

Team 1	Team 2	
Shubman Gill	Nitish Rana	Player's Name
Nitish Rana	Karun Nair	
Suryakumar Yadav	Suryakumar Yadav	
Rahul Tripathi	Rahul Tripathi	
Wriddhiman Saha	Sanju Samson	
Sanju Samson	Wriddhiman Saha	
Jonny Bairstow	Jos Buttler	
Liam Livingstone	David Miller	
Abhishek Sharma	Riyan Parag	
Rinku Singh	Rinku Singh	
Sam Curran	Hardik Pandya	
Dwayne Bravo	Andre Russell	
Keemo Paul	Sam Curran	
Shreyas Gopal	Kedar Jadhav	
Kedar Jadhav	Shreyas Gopal	
K Khaleel Ahmed	Billy Stanlake	
Billy Stanlake	Mohammed Shami	
Imran Tahir	KM Asif	
Sandeep Warriar	Ishant Sharma	
Shardul Thakur	K Khaleel Ahmed	
Dhawal Kulkarni	Ankit Rajpoot	
Mohammed Shami	Sandeep Warriar	
KM Asif	Kagiso Rabada	
Batting Performance: 7.7561 Bowling Performance:10.9355 Cost:14.55 cr	Batting Performance: 8.6894 Bowling Performance :10.8929 Cost :17.65 cr	

Two squads with higher bowling performance values are shown in the following table. With SPEA2, we may achieve the greatest possible bowling performance value of 10.936. In both

the batting and bowling performance situations, we obtained the maximum performance value using the NSGA-II algorithm when compared to the SPEA2 method in the other example.

Table 5.28: Cost-Effective Team Formation

Team 1	Team 2	
Nitish Rana	Shubman Gill	Player's Name
Suryakumar Yadav	Karun Nair	
Shubman Gill	Nitish Rana	
Mayank Agarwal	Rahul Tripathi	
Shreevats Goswami	Nicholas Pooran	
Nicholas Pooran	Shreevats Goswami	
Heinrich Klaasen	Heinrich Klaasen	
Sarfaraz Khan	Riyan Parag	
Ambati Rayudu	Rinku Singh	
Mandeep Singh	Abhishek Sharma	
Jofra Archer	Jofra Archer	
Krishnappa Gowtham	Mahipal Lomror	
Sam Curran	Mitchell Santner	
Shreyas Gopal	Shreyas Gopal	
Stuart Binny	Krishnappa Gowtham	
Mitchell_McClenaghan	Sandeep Warrier	
Sandeep Warrier	Sandeep Lamichhane	
Arshdeep Singh	Varun Aaron	
Deepak Chahar	Harshal Patel	
Prasidh Krishna	Rahul Chahar	
K Khaleel Ahmed	Navdeep Saini	
KM Asif	K Khaleel Ahmed	
Mayank Markande	Deepak Chahar	
Batting Performance: 6.8420 Bowling Performance: 8.5880 Cost: 6.3 cr	Batting Performance: 7.2292 Bowling Performance : 7.9879 Cost : 6.4 cr	

The lower-cost squads in the competition, as seen in the preceding table, do not have very strong batting and bowling performance values at all.

5.2.3 Performance Evaluation between NSGA-II and SPEA2

Because we utilized two techniques in our research, there is an apparent requirement for a performance assessment of the two algorithms that we used. The term "performance evaluation" refers to the process of determining the effectiveness or suitability of a technique. There are many different assessment metrics that may be used to evaluate the performance of different methods. We already know that the hypervolume [48] indicator, when utilized in evolutionary multi-objective optimization, is a set measure that is used to evaluate the performance of search algorithms as well as to guide the search. For the assessment of our techniques, we utilized the hypervolume values to compare the results of the two different approaches to the problem. Hypervolume values for both methods are shown in the following table, along with their respective means and standard deviations.

Table 5.29: Mean and Standard Deviation for Hypervolume Values of NSGA-II and SPEA2

Algorithm	Evaluation Metric	Mean	Standard Deviation
NSGA-II	Hypervolume	0.56150080398	0.02792188350421498
SPEA2		0.55061390096	0.022246406807299612

Because the mean and standard deviation of hypervolume values for the NSGA-II technique are greater than those for the SPEA2 method, the NSGA-II method is found to be more effective than the SPEA2 algorithm for our issue, as shown in the following table. The visualization is shown with the help of box-plots below.

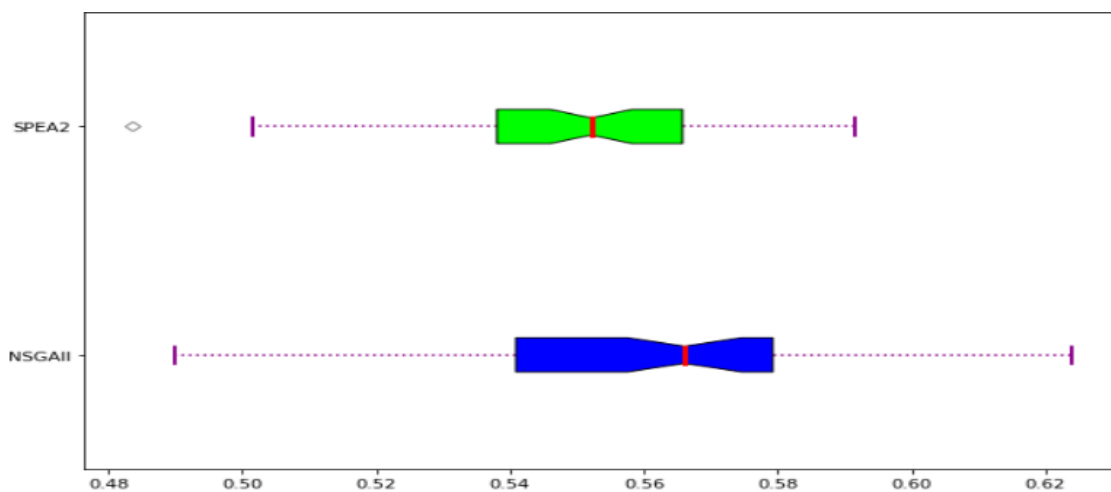


Figure 5.4: Multi-objective Optimization Dataset

As seen in the picture, the hypervolume values in NSGA-II are sparser than those in SPEA2. Another assessment measure that we use is the Mann-Whitney U test [49], which we use to compare the performance of the two techniques against one another.

Table 5.30: Mann-Whitney U-test

Evaluation Metric	Compare NSGA-II	Compare SPEA2
Hypervolume	0.948	0.000018832277663534073

The Mann-Whitney U test result for the NSGA-II technique is higher, as seen in the table. We may infer, then, from our investigation into the relative importance of different evaluative measures that the NSGA-II algorithms perform much better than the SPEA2 algorithm in our suggested model of obtaining optimum squads for a tournament.

Chapter 6

Future Work

As part of our study, we attempted to put together the best possible squad of players based on their predicted performance in the tournament as predicted by our model. Our model is separated into two stages. The first phase is concerned with forecasting the performance of the player in the forthcoming tournament via the use of machine learning techniques. It is the second phase that takes as its input the output of phase one, which is the performance of the players, and with some additional features creates a pool of players from which to create optimal squads by keeping the batting and bowling performance of the squads high while keeping the cost of the squads as low as possible. We have successfully completed both steps, and in the first phase, the Random Forest Classifier provides us with the highest accuracy for performance prediction, while in the second phase, the NSGA-II method provides us with a better result for squad formation. Our future plan is to reduce the drawbacks of our proposed system and develop the model more accurately. The limits, as well as our proposed solutions, will be discussed in more detail in the following sections.

6.1 Limitations

While attempting to forecast the performance of the players, we have experimented with several classification algorithms. As runs and wickets are discrete forms of values, the use of class imply a range of runs and wickets. So, the application of the machine learning classifier algorithms to forecast the class of runs and wickets do not offer us the actual worth of runs and wickets rather we have received an insight into the runs and wickets that players may score or take correspondingly. Another limitation is that the data is not enough for all the players as some of the players have only played a few matches in Indian Premier League, so the machine learning classifiers do not work well on those players' performance prediction. Players go to a particular tournament once or twice a year to compete, but they

also play for other teams throughout the year. We have just taken into account IPL playing time and statistics for our study, which may not be sufficient to forecast performance since we may not be aware of how the player performs from one Indian Premier League season to another in the midst of another. For the second phase, where we work on the multi-objective optimization techniques we have assumed the runs and wickets from first phase by taking the upper bound of each class of runs and wickets. The rationale of doing so is that the players may have the capacity of scoring that many runs or getting the greatest number of wickets are these. If we pick the lower limit of the ranges, it is possible that the value will be high for certain matches and that the value will not be traceable by the management when they are putting up a squad. There is a constraint in that the precise value of runs and wickets cannot be taken into consideration for the second stage of the research.

6.2 Possible Future Approach

To develop a model, one must consider the model from a variety of perspectives and rethink the model in order to make it better. There are numerous working scopes for our proposed model that must be considered in order to make the model robust and operational. Due to a data shortage for players who have recently enrolled in the IPL, we may collect data on their domestic cricketing performance in order to make an accurate estimation of the caliber of the batter or bowler. Even if we limit ourselves to the data from the Indian Premier League, there are many leagues around the world that may need this kind of recommendation model to propose what type of squad to build based on the performance of the players throughout the competition. It is possible to gather player data for the whole year in order to create a solid dataset, with the aid of which the machine learning algorithms may perform very well and accurately forecast the outcome of the game, thus improving the efficiency of the model. We currently have 152 players with their predicted performance for the upcoming IPL; however, in the real-life auction, there will be more than 500 players, both local and international, who will compete for the top prize. We are unable to examine all of the players in order to forecast their performances due to a lack of data. Our goal is to make the model as complete as possible, so that for local players, we can take into account their local playing data, and for foreign players, we can take into account the data from the leagues in which they compete. If we are successful in our endeavor, the creation of squads using multi-objective optimization may prove to be a successful strategy for a diverse group of individuals. It has been our experience that in auction-based tournaments, the prices of the players may sometimes rise to levels that we are unable to anticipate; for this reason, we operate with the players' base prices. In the future, it may be possible to investigate price deflections and include pricing into our model in order to make the model more practical for management to use. These methods may aid in improving the accuracy and robustness

of our model when applied to a real situation.

Chapter 7

Conclusion

Putting together the best possible squad for a tournament is the most essential step in participating in a game. Because of the enthusiasm and acceptance that franchise-based events have gained across the globe, virtually every cricket-playing country now organizes such tournaments. Participating in these types of tournaments appeals to investors who wish to earn money in a short period of time. Aside from the prize money, the revenue from sponsorships and other sources is very profitable for the organizers of these competitions. When participating in a competition, it is not uncommon to see teams who are just somewhat proficient at hitting or bowling. We encounter certain teams with a lesser budget on a regular basis. At the auction, we may be so focused on getting a high-priced player that we overlook a player who has the potential to do well in the next event. When the human brain is put to work on this, it is difficult to consider all of the players' advantages and disadvantages and come up with the best possible team for the competition. Even a tournament lasts a certain amount of time, and some managers forget that if their star player is injured during the tournament, they may need to find a suitable replacement for that player as soon as possible. In the era of artificial intelligence, we should not only rely on historical data to make decisions. We should use historical data to predict the performance of the players, and the team should be assembled in the most efficient manner possible based on the projection. This is the primary focus of our investigation. However, although there are certain shortcomings, we have achieved 92.332 percent accuracy for top-order hitters, while achieving 95.965 percent accuracy for middle-order batters, 90.979 percent accuracy for lower-order batters, and 94.726 percent accuracy for bowlers, respectively. We also obtained a total of 381 optimum squads using the NSGA-II technique, which provides a more sparse picture of the hypervolume metric than the previous method. If the proposed squad cannot be put together precisely, the management may search for players of comparable quality who were not included in the auction and can be added later. Due to certain constraints, we may not be able to cover all aspects of the research; nevertheless, our model provides sufficient in-

sight into how to form a squad and which players to choose in order to increase the winning probability for the tournament.

References

- [1] D. Thenmozhi, P. Mirunalini, S. Jaisakthi, S. Vasudevan, V. V. Kannan, and S. S. Sadiq, "Moneyball-data mining on cricket dataset," in *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pp. 1–5, IEEE, 2019.
- [2] S. Singh and P. Kaur, "Ipl visualization and prediction using hbase," *Procedia computer science*, vol. 122, pp. 910–915, 2017.
- [3] A. Mitra, S. Banerjee, D. Ganguly, R. Majumdar, and K. Chatterjee, "Innovative ranking strategy for ipl team formation,"
- [4] P. Shah and M. Shah, "Form-a new cricket statistics," *American Journal of Sports Science*, vol. 2, no. 3, pp. 53–55, 2014.
- [5] F. Ahmed, A. Jindal, and K. Deb, "Cricket team selection using evolutionary multi-objective optimization," in *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 71–78, Springer, 2011.
- [6] G. Karale, "Cricket player performance prediction."
- [7] K. Passi and N. Pandey, "Predicting players' performance in one day international cricket matches using machine learning," *Computer Science & Information Technology (CS & IT)*, 2017.
- [8] C. D. Prakash, C. Patvardhan, and C. V. Lakshmi, "Data analytics based deep mayo predictor for ipl-9," *International Journal of Computer Applications*, vol. 152, no. 6, pp. 6–10, 2016.
- [9] V. Kanungo *et al.*, "Data visualization and toss related analysis of ipl teams and batsmen performances.," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 9, no. 5, 2019.
- [10] M. J. Hossain, M. A. Kashem, M. S. Islam, E. Marium, *et al.*, "Bangladesh cricket squad prediction using statistical data and genetic algorithm," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEE-iCT)*, pp. 178–181, IEEE, 2018.

- [11] S. B. Jayanth, A. Anthony, G. Abhilasha, N. Shaik, and G. Srinivasa, "A team recommendation system and outcome prediction for the game of cricket," *Journal of Sports Analytics*, vol. 4, no. 4, pp. 263–273, 2018.
- [12] F. Ahmed, K. Deb, and A. Jindal, "Multi-objective optimization and decision making approaches to cricket team selection," *Applied Soft Computing*, vol. 13, no. 1, pp. 402–414, 2013.
- [13] M. G. Jhanwar and V. Pudi, "Predicting the outcome of odi cricket matches: A team composition based approach," in *MLSA@ PKDD/ECML*, 2016.
- [14] S. Anjali, V. Aswini, and M. Abirami, "Predictive analysis with cricket tweets using big data," *International Journal of Scientific & Engineering Research*, vol. 6, no. 10, pp. 63–70, 2015.
- [15] M. Haghighat, H. Rastegari, N. Nourafza, N. Branch, and I. Esfahan, "A review of data mining techniques for result prediction in sports," *Advances in Computer Science: an International Journal*, vol. 2, no. 5, pp. 7–12, 2013.
- [16] S. Muthuswamy and S. S. Lam, "Bowler performance prediction for one-day international cricket using neural networks," in *IIE Annual Conference. Proceedings*, p. 1391, Institute of Industrial and Systems Engineers (IISE), 2008.
- [17] S. Shah, P. J. Hazarika, and J. Hazarika, "A study on performance of cricket players using factor analysis approach," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 656–660, 2017.
- [18] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques third edition," *The Morgan Kaufmann Series in Data Management Systems*, vol. 5, no. 4, pp. 83–124, 2011.
- [19] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 3, pp. 1480–1483, IEEE, 1996.
- [20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [21] R. A. King, K. Deb, and H. Rughooputh, "Comparison of nsga-ii and spea2 on the multi-objective environmental/economic dispatch problem," *University of Mauritius Research Journal*, vol. 16, no. 1, pp. 485–511, 2010.
- [22] R. W. Saaty, "The analytic hierarchy process—what it is and how it is used," *Mathematical modelling*, vol. 9, no. 3-5, pp. 161–176, 1987.

- [23] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.
- [24] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of nsga-ii for optimizing machining process parameters," *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011.
- [25] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Engineering*, vol. 5, no. 1, p. 1502242, 2018.
- [26] H. H. Maheta and V. K. Dabhi, "An improved spea2 multi objective algorithm with non dominated elitism and generational crossover," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 75–82, IEEE, 2014.
- [27] "Indian premier league official website." <http://www.iplt20.com/>.
- [28] "Howstat! the cricket statisticians." <http://www.howstat.com/>.
- [29] "Cricmetric." <http://www.cricmetric.com/>.
- [30] "Espncricinfo." <http://www.espncricinfo.com/>.
- [31] "How flying seriously messes with your mind."
- [32] C. M. LLC, "Flight time and distance calculator."
- [33] "Ipl auction 2018 : Ipl 11 player auction - updated teams players - ndtv sports."
- [34] "Ipl auction 2018: Complete list of all players sold and unsold," Oct 2020.
- [35] Mykhelcom, "Ipl auction 2018: Ipl auction 2018 players list, updated teams players."
- [36] C. A. Depken and R. Rajasekhar, "Open market valuation of player performance in cricket: Evidence from the indian premier league," *Available at SSRN 1593196*, 2010.
- [37] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the first international conference on genetic algorithms and their applications, 1985*, Lawrence Erlbaum Associates. Inc., Publishers, 1985.
- [38] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural optimization*, vol. 4, no. 2, pp. 99–107, 1992.
- [39] C. Fonseca and P. Fleming, "Multiobjective genetic algorithms in iee colloquium on genetic algorithms for control systems engineering," 1993.

- [40] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [41] D. W. Corne, J. D. Knowles, and M. J. Oates, "The pareto envelope-based selection algorithm for multiobjective optimization," in *International conference on parallel problem solving from nature*, pp. 839–848, Springer, 2000.
- [42] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "Pesa-ii: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd annual conference on genetic and evolutionary computation*, pp. 283–290, 2001.
- [43] N. Srinivas and K. Deb, "Muultiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [44] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.
- [45] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International conference on parallel problem solving from nature*, pp. 832–842, Springer, 2004.
- [46] "crossover.py."
- [47] "mutation.py."
- [48] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [49] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

Appendix A

Resources and Data sets

- Code of Top-order Batter: [Top-order Batter](#)
- Code of Middle-order Batter: [Middle-order Batter](#)
- Code of Lower-order Batter: [Lower-order](#)
- Dataset of Players (Machine Learning Phase): [Dataset of Machine Learning Phase \(All Player's Individual Data](#)
- Dataset of Players (Multi-objective Optimization Phase): [Dataset of All Players \(Multiobjective Optimization\)](#)
- Squad Selection Code Using NSGAI and SPEA2 Multi-objective Optimization Methods: [Code of Multi-objective Optimization\)](#)

Generated using Undergraduate Thesis L^AT_EX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Friday 2nd July, 2021 at 4:21pm.