Sports Analytics

- What is it?

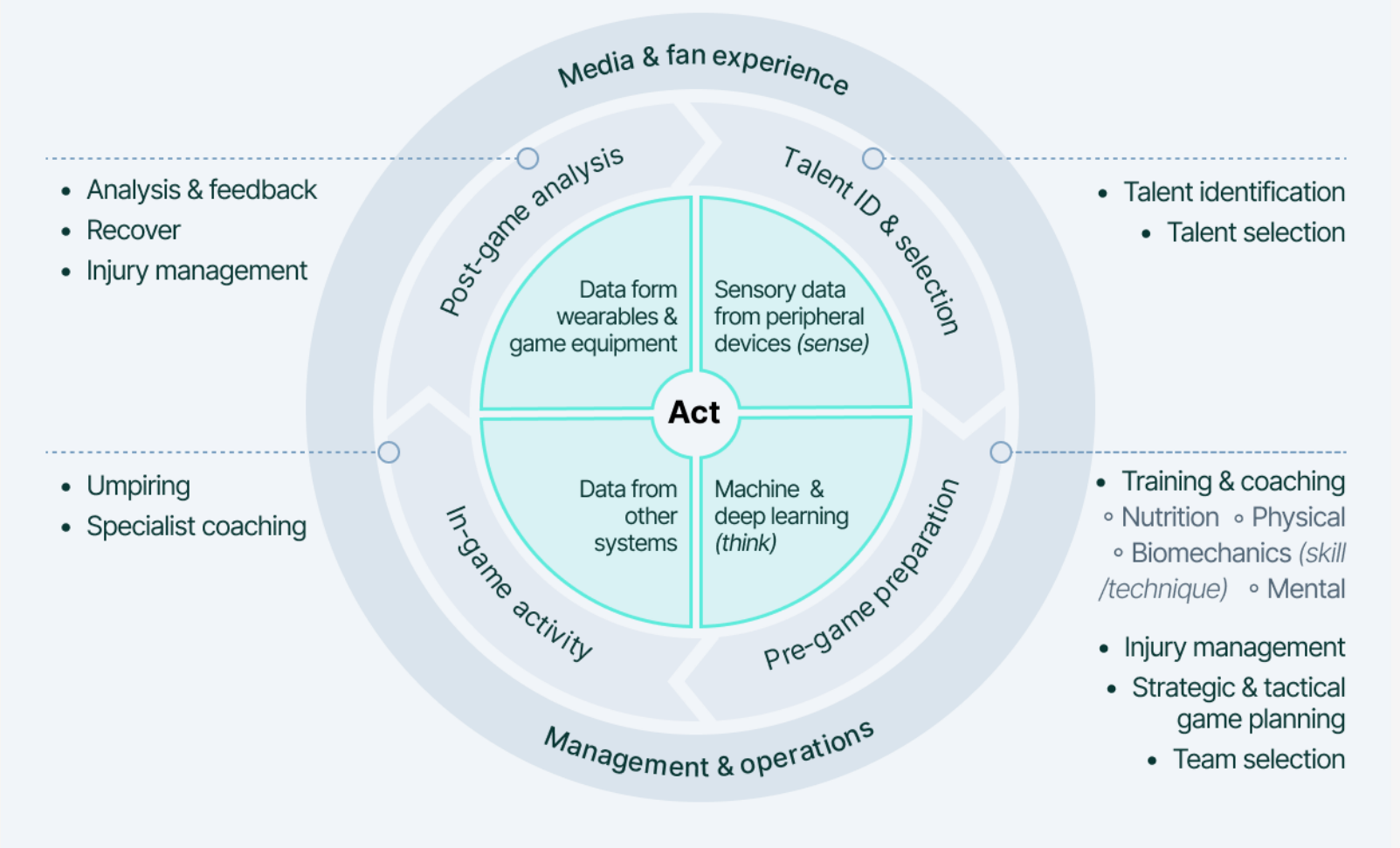
Types

- Field-level analysis
  - focused on the behavior of athletes, coaches, and referees
    - who is the best opener?
    - Player performance
    - Opponent performance
    - Game strategy
- Analysis of management and policymakers' decisions
  - surface patterns and insights through data that would help increase ticket and merchandise sales, improve fan engagement.
- Analysis of the literature that uses sports data to address various questions in the fields of economics and psychology.

Applications

- Injury Predictions
- Player Valuations
- Team Strategy
- Evaluating Ticket Churn
- Ticket Pricing
- Sports Betting

Applications



Applications

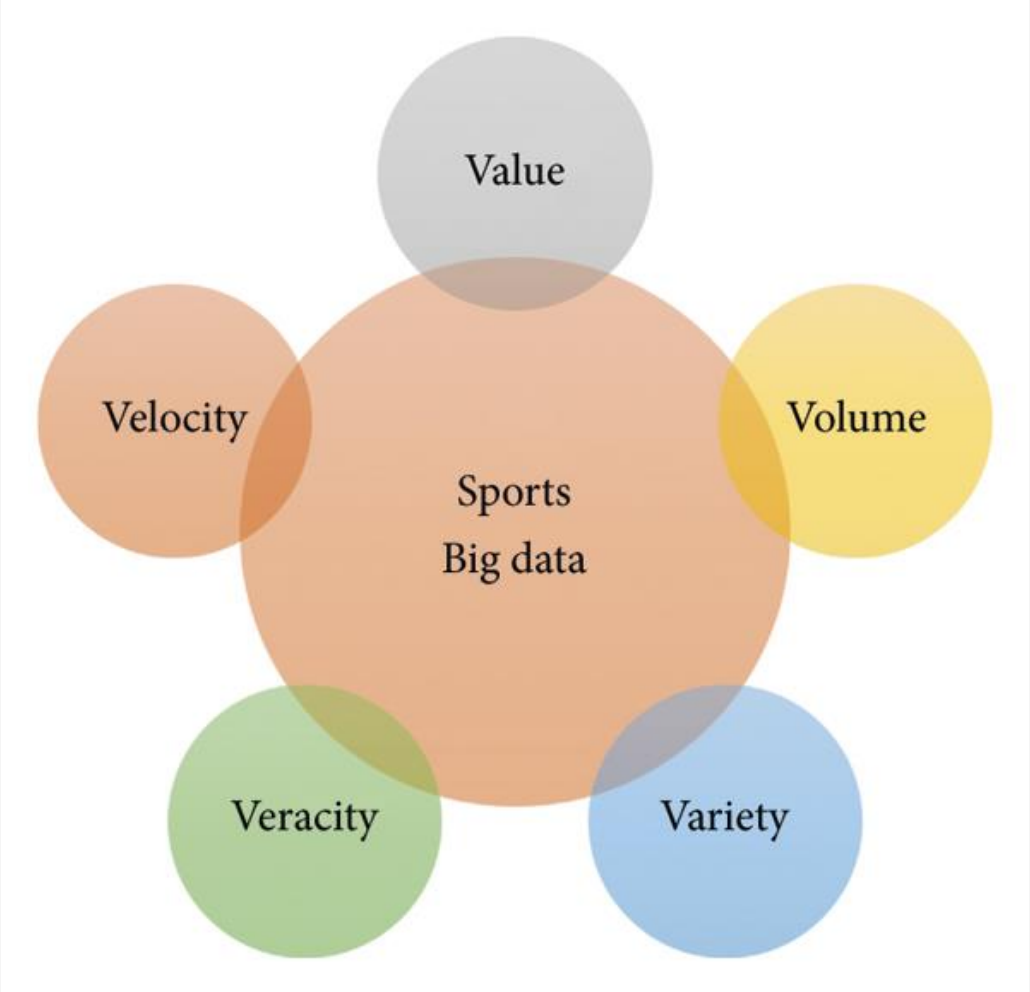
Media & fan experience

- Fan relationship management
- News & content
- Factory sports
- eSports
- Sports betting
- Media rights

Management & opeations

- Competition management
- Club & team management
- Venues, events & ticketing
- Sponsoring
- Merchandising
- Payments

How?



Sample Problem

- We want to create a cricket score predictor that takes the inputs:

- Country playing – C
- Runs scored so far in the match – Rs
- Wickets remaining in the match – Wr
- Overs remaining – Or
- Our predictor should tell us what could be the final score at end of 50 overs.

- For example, we could ask, “India scored 52 runs in 10 overs losing 1 wicket. What would be the final score?” and our predictor can provide a guess – say 305 runs.

What variables are relevant?

- The final score of a team in a cricket match depends on many things, including:

  - The playing team and their batsmen
  - The opposition team and their bowlers, fielders
  - The ground and pitch they play on
  - The weather and historical weather
  - Crowd attending the game and how much they are cheering
  - Mood of the umpire!!!

Factors

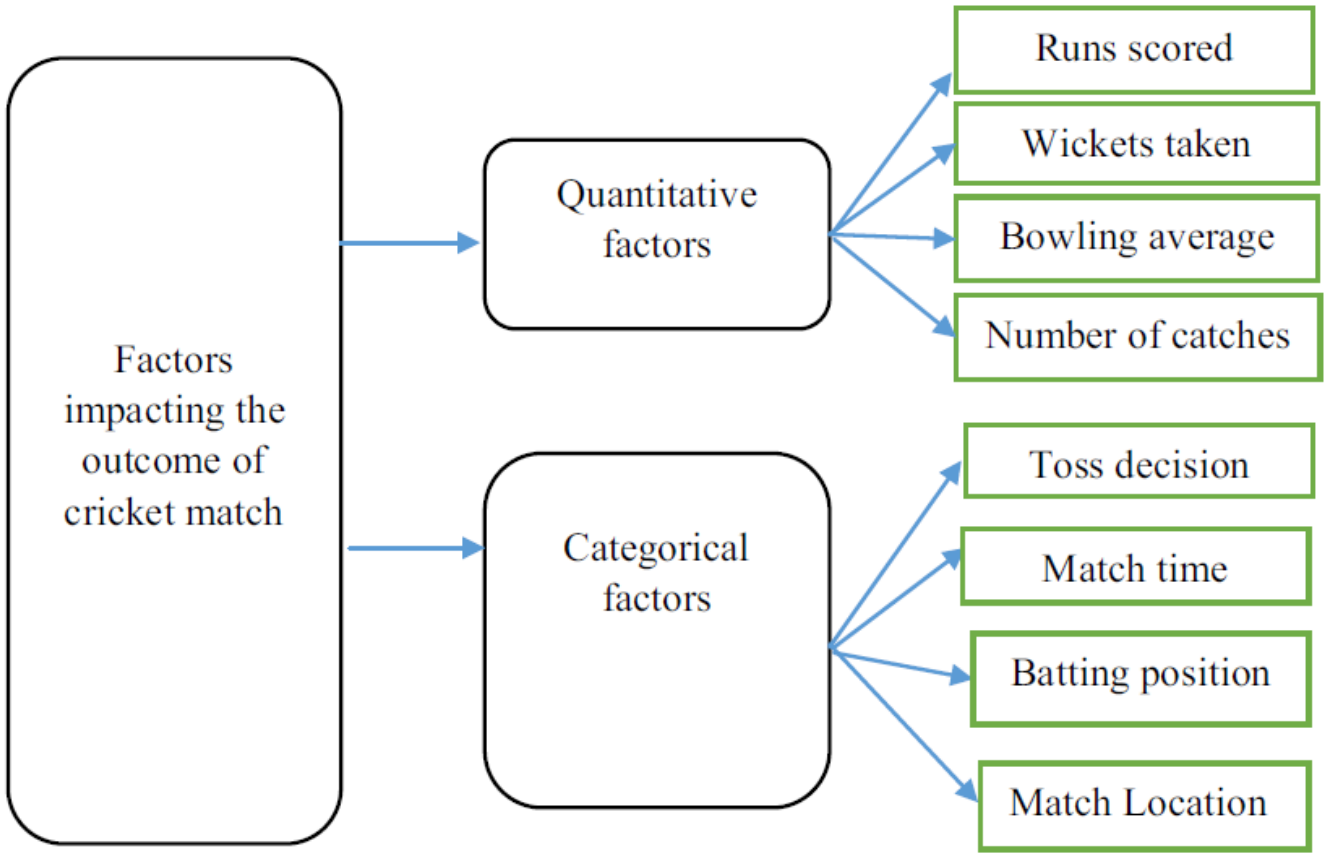


Fig. 1 The emergence of factors from prior studies

Our model as an equation

- Let’s say Rp denotes our predicted Runs. We can define Rp as

- Rp = f(C, Rs, Wr, Or)
  where
  - C = Country
  - Rs = Runs already scored
  - Wr = Wickets Remaining
  - Or = Overs Remaining

- f is a prediction function that does some magic to calculate Rp

More variables?

- Let's define few more variables.

- Os (Overs so far) = 50 – Or
- RRs (Run Rate so far) = Rs / (50-Or)
- RRr (Run Rate for remaining overs) – this should be predicted
- Given these variables, we can rewrite Rp (Runs predicted) as
  - Rp (Runs Predicted) = Rs + Or * RRr
- So if we can build a model to predict RRr, we can calculate Predicted score.

- We can argue that Run Rate in remaining overs will be a function of (country, run rate so far, overs remaining, wickets remaining)

- RRr = f(C, RRs, Wr, Or)

Country specific function

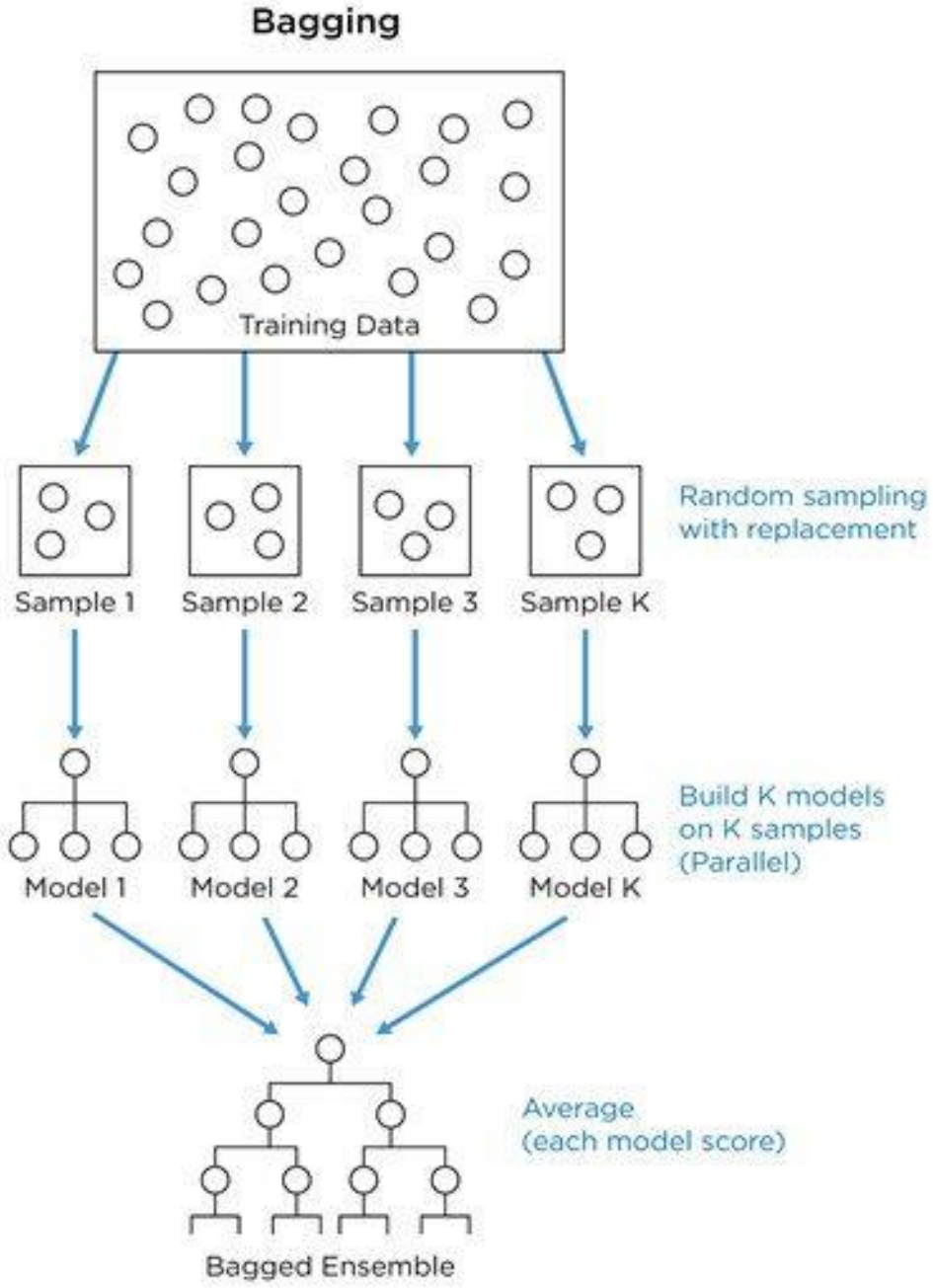
- RRr for country n = fn(RRs, Wr, Or)

Final function

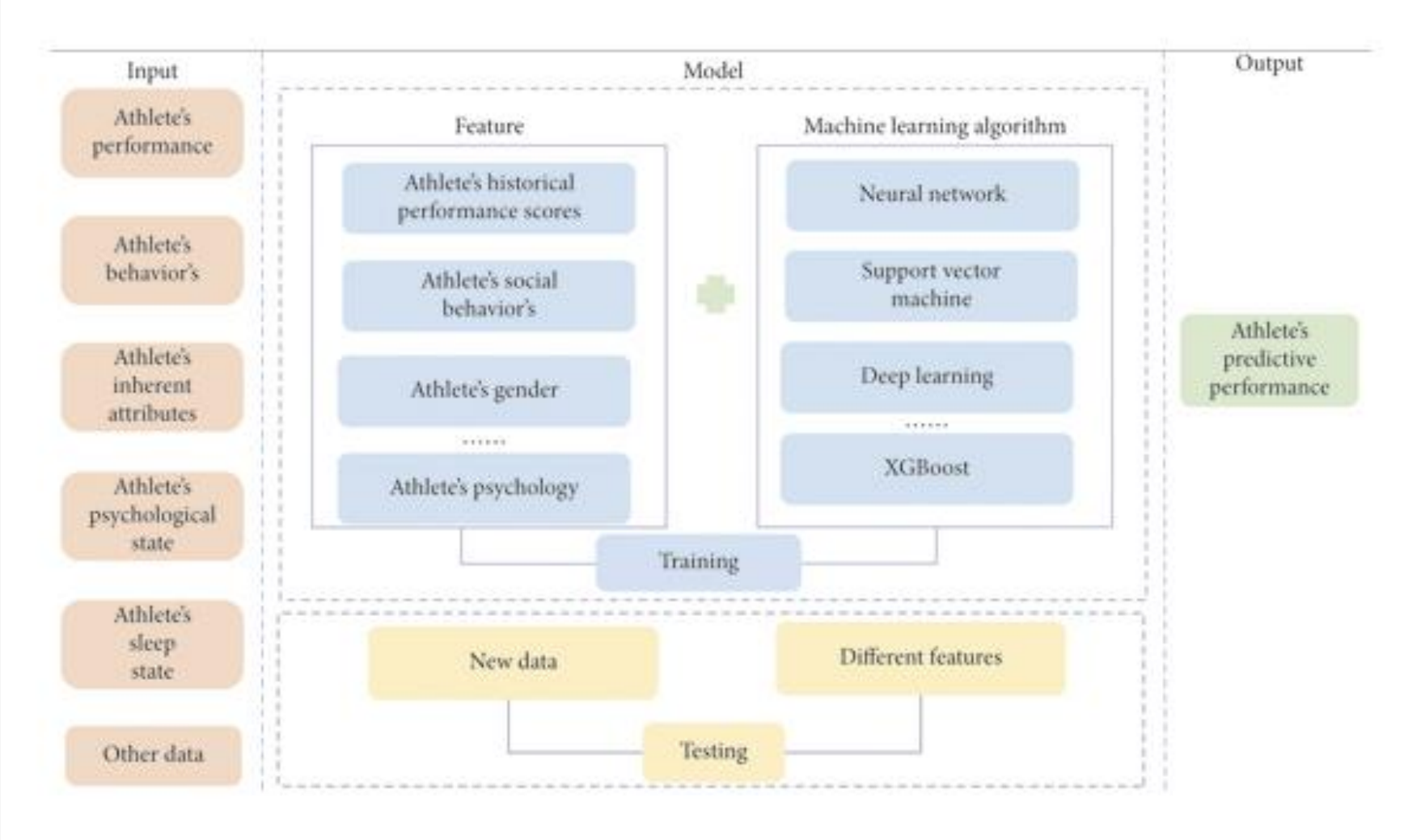
- RRr = m1*RRs + m2*Wr + m3*Or + const

But why only regression

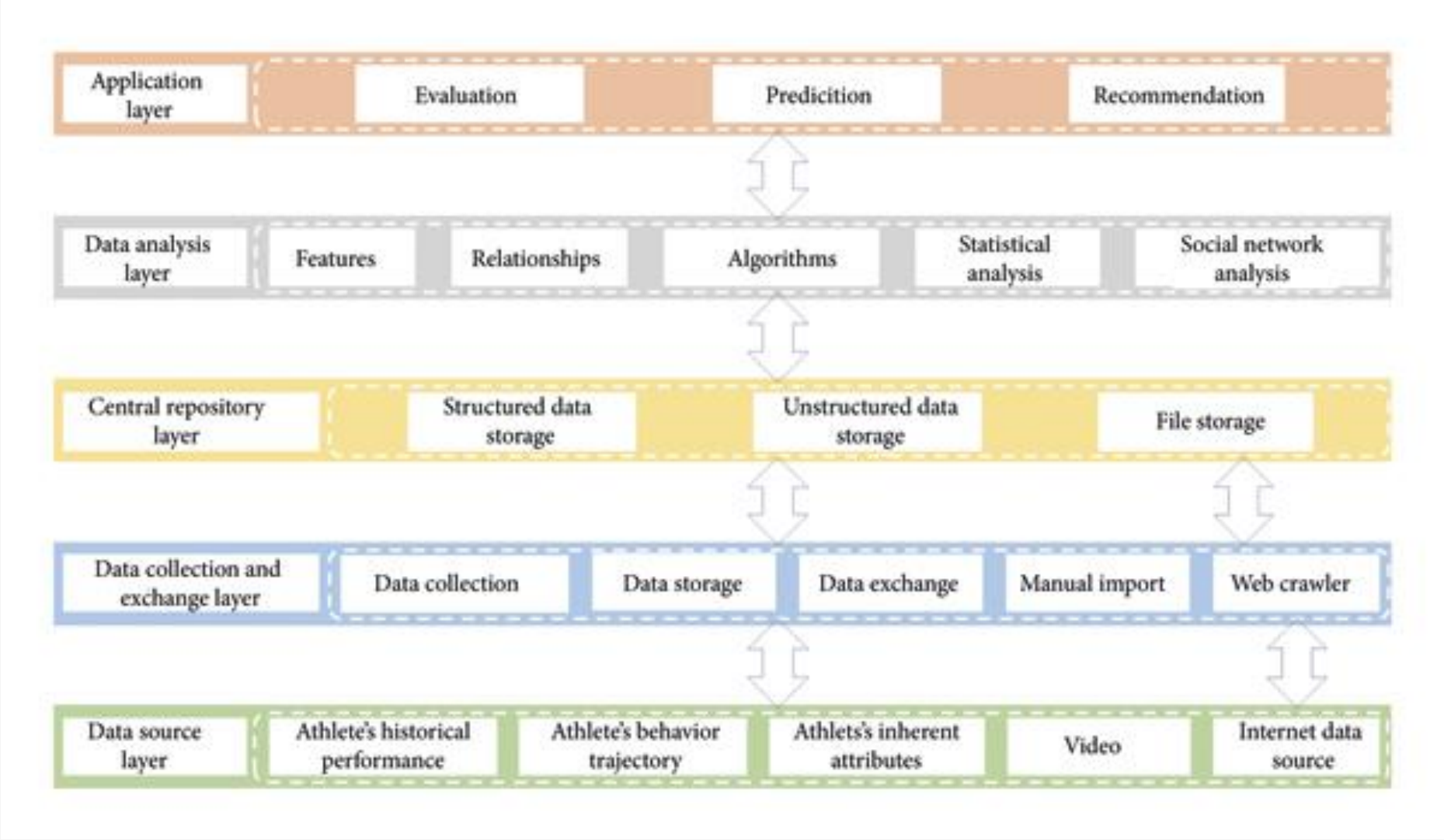
- Ensemble Modeling

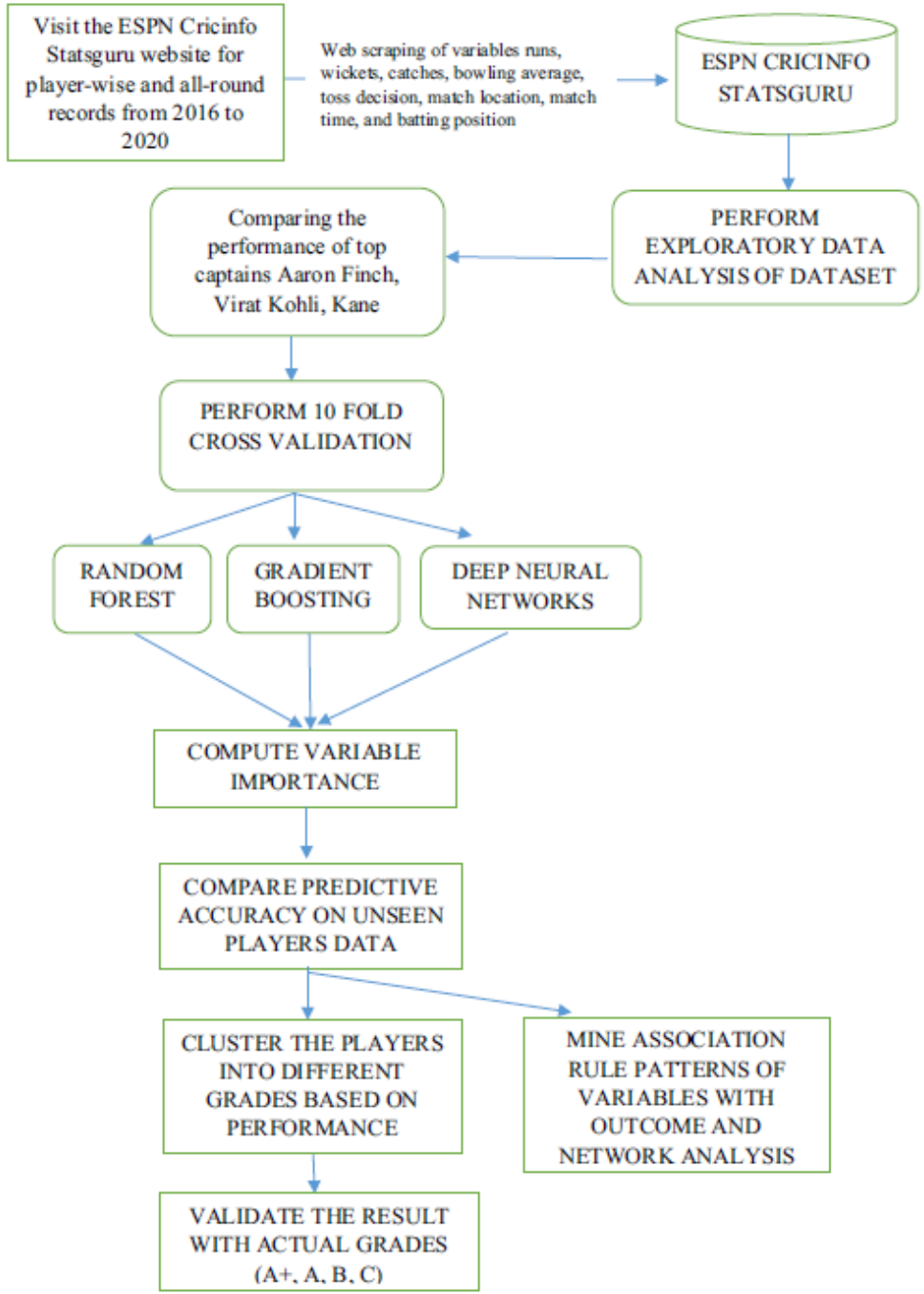


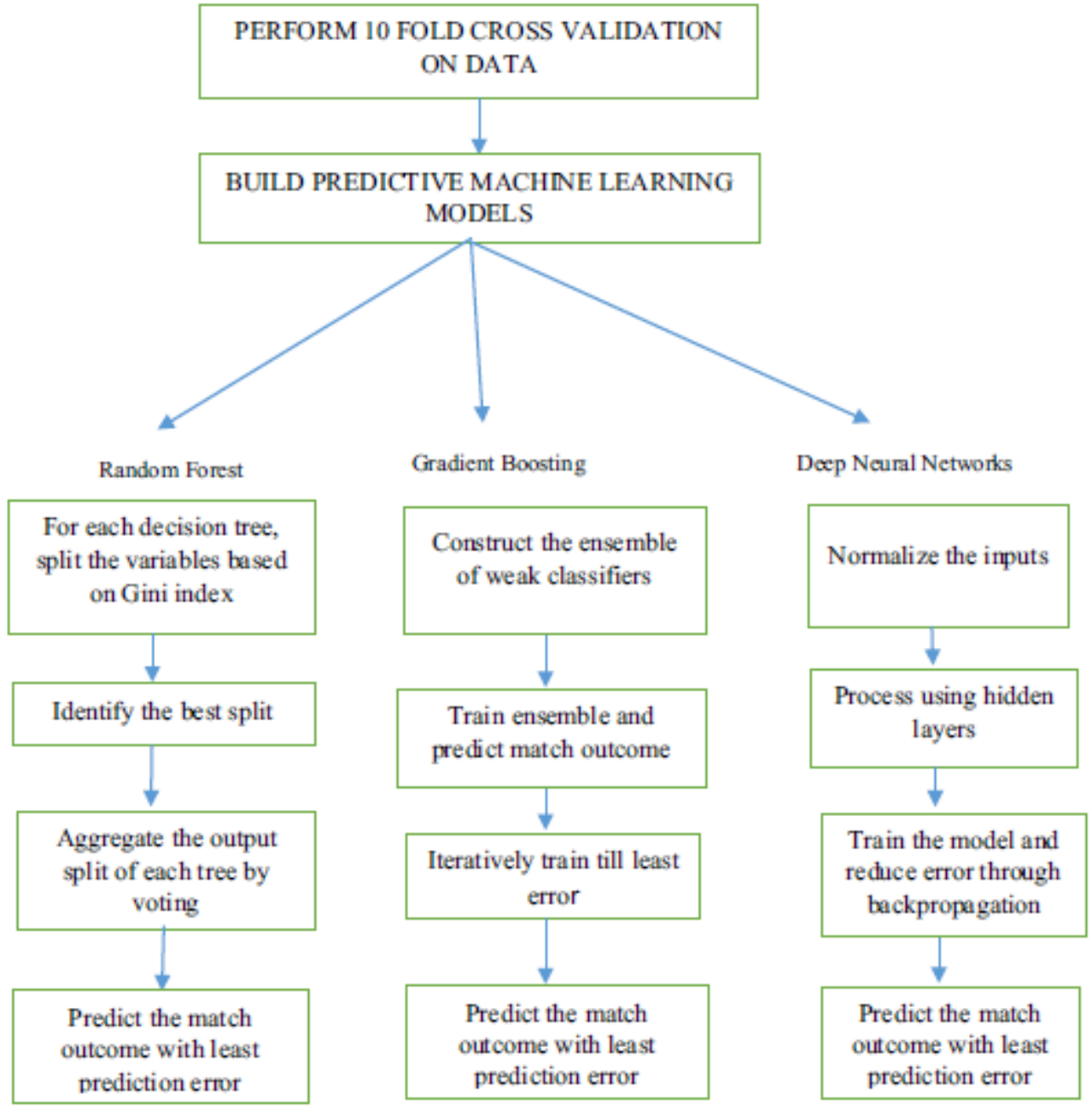
Requirements



Requirements




Visit the ESPN Cricinfo Statsguru website for player-wise and all-round records from 2016 to 2020
Web scraping of variables runs, wickets, catches, bowling average, toss decision, match location, match time, and batting position
ESPN CRICINFO STATSGURU
PERFORM EXPLORATORY DATA ANALYSIS OF DATASET
Comparing the performance of top captains Aaron Finch, Virat Kohli, Kane
PERFORM 10 FOLD CROSS VALIDATION
RANDOM FOREST
GRADIENT BOOSTING
DEEP NEURAL NETWORKS
COMPUTE VARIABLE IMPORTANCE
COMPARE PREDICTIVE ACCURACY ON UNSEEN PLAYERS DATA
CLUSTER THE PLAYERS INTO DIFFERENT GRADES BASED ON PERFORMANCE
MINE ASSOCIATION RULE PATTERNS OF VARIABLES WITH OUTCOME AND NETWORK ANALYSIS
VALIDATE THE RESULT WITH ACTUAL GRADES (A+, A, B, C)


PERFORM 10 FOLD CROSS VALIDATION ON DATA
BUILD PREDICTIVE MACHINE LEARNING MODELS
Random Forest
For each decision tree, split the variables based on Gini index
Identify the best split
Aggregate the output split of each tree by voting
Predict the match outcome with least prediction error
Gradient Boosting
Construct the ensemble of weak classifiers
Train ensemble and predict match outcome
Iteratively train till least error
Predict the match outcome with least prediction error
Deep Neural Networks
Normalize the inputs
Process using hidden layers
Train the model and reduce error through backpropagation
Predict the match outcome with least prediction error