Position Flexibility in Premier League

Anweshan Adhikari, Bruck Setu

2024-12-16

Table of contents

Introduction

Research Context

In modern football, positional versatility has become crucial as teams adapt to different tactical systems and manage packed schedules. While managers traditionally rely on intuition to identify versatile players, a data-driven approach could improve this process. This study uses supervised and unsupervised learning techniques to analyze player data and identify patterns that indicate positional versatility, potentially offering teams a systematic method for evaluating players' ability to adapt to different roles.

0.1 Research Question

To what extent can supervised and unsupervised learning algorithms predict alternative playing positions for Premier League players based on their performance metrics?

0.2 Data Overview

The analysis combines Premier League player statistics (2023/2024 season) and FIFA 24 player data, resulting in a dataset of 445 players with performance metrics. Key variables include:

Table 1: Overview of Variable Groups in Analysis

Variable Group	Description	Key Metrics
Player Information	Basic player attributes and characteristics	Player name, Nationality, Positions, Main Position, Age, Height, Weight, Preferred foot
Playing Time	Metrics tracking player participation	Matches played (MP), matches started, Minutes Played, 90s Played (X90s)
Goal Contributions	Direct offensive and disciplinary statistics	Goals (Gls), assists (Ast), penalties (PK), Yellow/Red cards (CrdY/CrdR)
Expected Performance	Advanced metrics predicting player output	Expected goals (xG), Expected Assists (xAG), non-penalty expected goals (npxG)
Progressive Actions	Actions advancing ball towards goal	Progressive carries (PrgC), Rasses (PrgP), Receives (PrgR)
Per 90 Metrics	Performance metrics normalized to full game	Goals per 90 (Gls_90), Assists per 90 (Ast_90), Expected goals per 90 (xG_90)
*		

1 Unsupervised Learning

1.1 Hierarchial Clustering

Hierarchical clustering helps us identify natural groupings within positions based on statistical similarities. The process works by id identifies the two positions with the most similar performance metrics, groups them together, and repeats this process until all positions are connected in a tree-like structure. The height of connections in our dendrogram shows the degree of difference between positions - higher connections indicate bigger statistical differences in performance metrics.

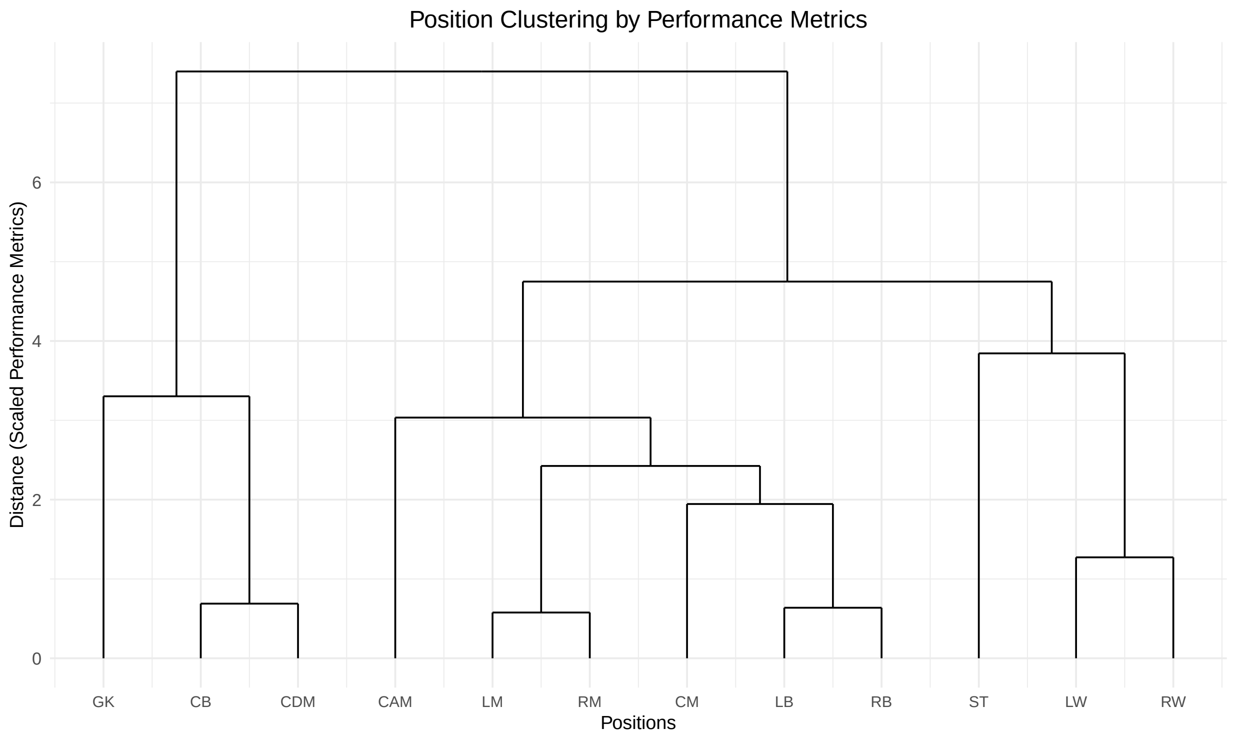
```r
pl_data<- read_csv("processed_data.csv")

position_summary <- pl_data %>%
  group_by(main_position) %>%
  summarise(
    Gls_90 = mean(Gls_90, na.rm = TRUE),
    Ast_90 = mean(Ast_90, na.rm = TRUE),
    xG_90 = mean(xG_90, na.rm = TRUE),
    xAG_90 = mean(xAG_90, na.rm = TRUE),
    PrgC = mean(PrgC, na.rm = TRUE),
    PrgP = mean(PrgP, na.rm = TRUE),
    PrgR = mean(PrgR, na.rm = TRUE)
  ) %>%
  # Remove overlapping positions (following class notes approach)
  filter(!main_position %in% c("CF", "LWB", "RWB"))

position_stats_scaled <- scale(position_summary[,-1])
rownames(position_stats_scaled) <- position_summary$main_position

position_distances <- dist(position_stats_scaled)
hc_positions <- hclust(position_distances)

ggdendrogram(hc_positions, theme_dendro = TRUE) +
  labs(title = "Position Clustering by Performance Metrics",
       y = "Distance (Scaled Performance Metrics)",
       x = "Positions") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.text.y = element_text(size = 10)
  )
```



This dendrogram shows the hierarchical clustering of English Premier League positions based on their similarity. Goalkeeper (GK) merging with the rest at a relatively high dissimilarity height indicating that it is different from the rest of the positions. Defensive positions cluster together, with center back (CB) and central defensive midfielder (CDM) showing strong similarity although the cluster with those 2 positions and GK join the cluster of all other positions very high indicating that these positions have very distinct roles from the rest. Left back (LB) and right back (RB) group with central midfielder (CM), as they have the similar defensive and transitional offense responsibilities. Left and right midfielders (LM & RM) join at the lowest height indicating how their responsibilities are identical but just on mirrored sides of the field. As for the attacking positions, left wingers (LW) and right wingers (RW) cluster relatively closely before merging with central attacking midfielder (CAM) and at the highest joining distance was strikers (ST).

The clustering shows certain positions naturally pair together based on their performance metrics: defensive positions (CB-CDM), wide midfielders (RM-LM), and fullbacks (LB-RB). These pairings suggest players in these roles produce statistically similar numbers in key metrics, indicating where our classification model might have difficulty drawing clear boundaries between positions.

2 Supervised Learning Method

2.1 Support Vector Machines

Support Vector Machines (SVM) are supervised machine learning models used for classification tasks. SVM works by finding a hyperplane that best separates data points into different classes while maximizing the distance between the closest use of each class, called support vectors. In this analysis, we employ a multi-variable SVM to classify player positions, where each player is represented as a point in a multi-dimensional space, each corresponding to a performance metric, such as goals, assists, or progressive actions.

The distance of a player from the hyperplane is calculated as:

$$d = \frac{|w_1x_1 + w_2x_2 + ... + w_nx_n + b|}{||w||}$$

where:

- $x_1, x_2, ..., x_n$ are the player's metrics
- $w_1, w_2, ..., w_n$ are weights showing how important each metric is
- b is the bias term that shifts the boundary to its optimal position
- $||w|| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$ normalizes the distance

To handle multiple classes (e.g., ST, LW, CB etc.) the e1071 package in R, which we will be using, applies a One-vs-One (OvO) strategy. For N classes(12 in our case), OvO creates N×(N−1)/2 binary SVM models, where each model distinguishes between a pair of classes. During prediction, each binary model "votes" for one of its two classes, and the class receiving the most votes is assigned as the final prediction.

2.1.1 Model Implementation

For SVM, Feature selection in our analysis was guided by the need to capture diverse aspects of player performance while avoiding redundancy. We focused on performance metrics normalized per 90 minutes (goals, assists, expected goals, and expected assists) to ensure fair comparison across players with different amounts of playing time. Progressive actions (carries, passes, and receives) were included as they capture how players move the ball forward, which is crucial for distinguishing between similar positions like central midfield roles. Physical attributes (height and weight) were added to help differentiate between positions that might require different physical characteristics.

```r
position_features <- c(
  # Base metrics per 90 (normalized for playing time)
  "Gls_90", "Ast_90", "xG_90", "xAG_90",
```

Table 2: Distribution of Positions in Training Data

Position	Count
CAM	17
CB	59
CDM	37
CM	40
GK	24
LB	26
LM	14
LW	15
RB	25
RM	14
RW	17
ST	42

```
  # Progressive actions (different aspect of play)
  "PrgC", "PrgP", "PrgR",

  # Physical attributes
  "height_cm", "weight_kg",

  # Technical attributes
  "weak_foot",

  # Disciplinary (different dimension)
  "CrdY", "CrdR",

  # Penalty responsibility
  "PK", "PKatt"
)
```

```
# 2. Split into training and test sets
set.seed(12142024)

train_index <- createDataPartition(svm_data$main_position, p = 0.7, list = FALSE)
train_data <- svm_data[train_index, ]
test_data <- svm_data[-train_index, ]
```

We split our data using a 70-30 ratio with createDataPartition(). With only 445 players divided across 12 different positions, we opted for a larger training set (70%) to ensure sufficient examples of each position for our model to learn from, particularly for less common positions.

The distribution of positions in training data shows significant imbalance: Center Backs (59) and Strikers (42) are heavily represented, while Left Midfield (14) and Right Midfield (14) have much smaller sample sizes. This imbalance likely impacts our SVM model's learning - particularly evident when the model later achieves 0% accuracy for RM only having 14 training examples. This suggests not just a sample size issue, but potentially that RM's statistical profile overlaps significantly with other positions, making it difficult for SVM to establish clear decision boundaries for these underrepresented classes.

2.1.2 SVM Models

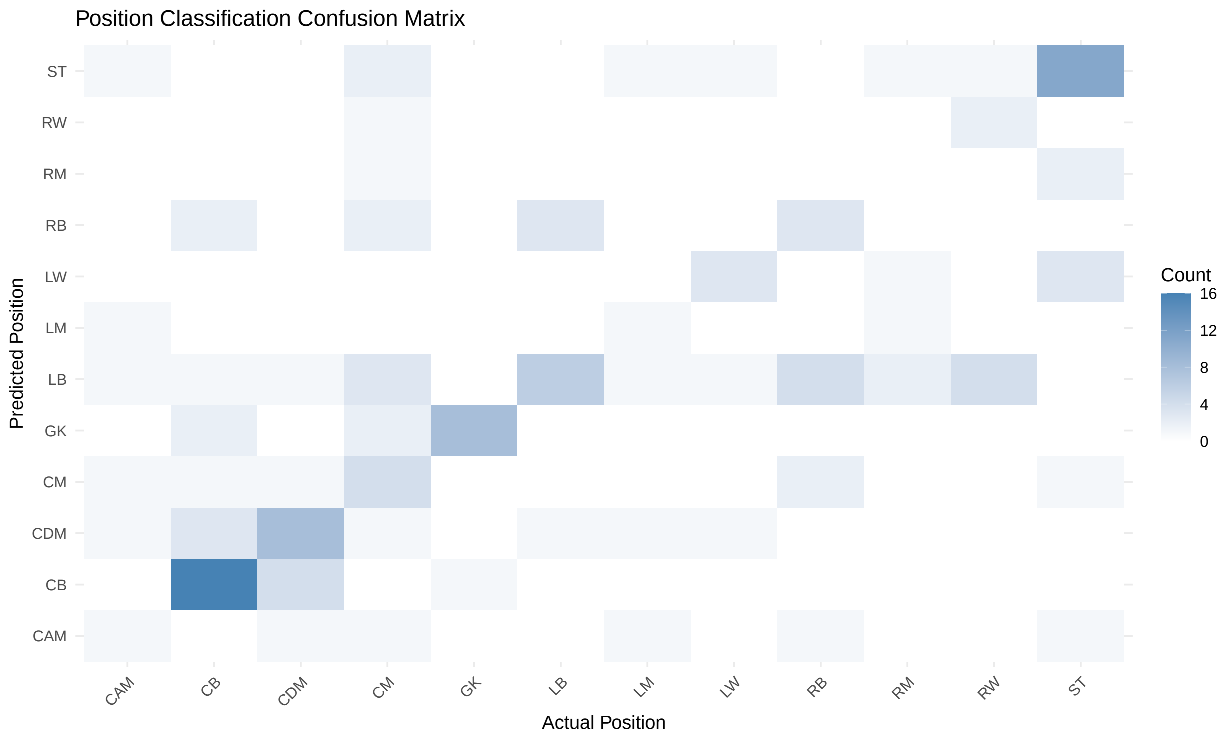
```r
svm_model <- svm(main_position ~ .,
                 data = train_data,
                 kernel = "linear",
                 type = "C-classification")


# Train SVM with radial kernel
svm_radial <- svm(main_position ~ .,
                  data = train_data,
                  kernel = "radial",
                  type = "C-classification")

# Train SVM with polynomial kernel
svm_poly <- svm(main_position ~ .,
                data = train_data,
                kernel = "polynomial",
                type = "C-classification")
```

We experimented with three different SVM kernel functions - linear, radial, and polynomial - to find the best approach for classifying player positions. We used C-classification, which is appropriate for our multi-class problem with 12 distinct positions. While we could have tuned the cost parameter to balance misclassification penalties, we kept the default cost=1 to avoid overfitting given our small dataset.

The linear kernel performed best with 47.4% accuracy. The simpler linear kernel, while not having the best accuracy, outperformed more complex kernels, suggesting that adding model complexity doesn't help capture the true relationships in our data.



The confusion matrix reveals patterns in misclassification that mirror our clustering analysis. For example, the model often confuses CB with CDM and vice versa (3 CBs predicted as CDM and 4 CDMs predicted as CB), and LM with RM, which matches how these positions clustered together in our dendrogram. This suggests that some position pairs are genuinely difficult to distinguish based on performance metrics alone. The model's Kappa value of 0.413 indicates moderate agreement beyond chance, suggesting that while the model performs better than random guessing, there's still substantial room for improvement in distinguishing between similar positions.

The matrix notably shows that fullbacks (LB/RB) are frequently misclassified across multiple positions - LBs were predicted as RB (4 times), RW (4 times), and CM, while RBs were often classified as LB (3 times) and CM (2 times). This aligns with the modern football trend where fullbacks take on hybrid roles, sometimes acting as wide midfielders or inverted fullbacks in possession.

We initially attempted binary classification for each position (e.g., "Striker" vs "Not Striker"), but encountered significant class imbalance issues. For instance, in the striker classification, we had only 55 strikers versus 408 non-strikers in our dataset. This imbalance led to high accuracy scores that were misleading - the models simply predicted "not striker" for almost every case, achieving high accuracy but failing to identify actual strikers (high specificity but very low sensitivity). Similar issues occurred across all positions, with the minority class (the position being classified) being overwhelmed by the majority class (all other positions). While adding weights on the positions could potentially address this imbalance, we opted to omit these binary classifications from our analysis as they didn't provide meaningful insights into positional flexibility. The main advantage of this binary approach would have been the ability to classify players into multiple positions simultaneously, better reflecting the modern game's tactical fluidity where players often excel in several roles. However, the class imbalance issues prevented us from effectively implementing this multi-position classification approach. This experience highlighted us the challenges of working with inherently imbalanced football position data.

Table 3: SVM Classification Accuracy by Position

Position	**Accuracy**
CAM	16.7%
CB	64.0%
CDM	53.3%
CM	23.5%
GK	88.9%
LB	60.0%
LM	20.0%
LW	50.0%
RB	30.0%
RM	0.0%
RW	28.6%
ST	61.1%

Breaking down accuracy by position reveals that the model excels at identifying specialized roles: Goalkeepers (88.9%), Center Backs (64%), and Strikers (61.1%). However, it struggles significantly with midfield positions - particularly Right Midfield (0%) and Central Attacking Midfield (16.7%). This aligns with football intuition and dendogram, as midfield roles often share overlapping responsibilities while goalkeepers, defenders, and strikers have more distinct performance metrics.

2.2 Random Forest

```r
data<- read_csv("processed_data.csv") |>
  mutate(main_position = case_when(main_position == "CF" ~ "ST",
                                   main_position == "RWB" ~ "RB",
                                   main_position == "LWB" ~ "LB",
                                   TRUE ~ main_position))
data_clean <- data[,1:31]
data_clean <- data_clean[!duplicated(data_clean$Player), ]
```

```r
set.seed(12142024)
training_rows <- sample(1:nrow(data_clean), size = 0.7 * nrow(data_clean))
soccer_train_data <- data_clean[training_rows, ]
soccer_test_data <- data_clean[-training_rows, ]
```

```r
set.seed(12142024)
modelrf <- train(factor(main_position) ~ .,
                data = soccer_train_data |>
                  dplyr::select(-Player),
                method = "rf")

soccer_data_pred1 <- soccer_test_data |>
  mutate(predicted_position = predict(modelrf, newdata = soccer_test_data))

c<-confusionMatrix(data = as.factor(soccer_data_pred1$predicted_position),
                   reference = as.factor(soccer_data_pred1$main_position))
```

2.3 Ranger

```r
set.seed(12142024)
modelrang <- train(factor(main_position) ~ .,
                data = soccer_train_data |>
                  dplyr::select(-Player),
                method = "ranger",
                trControl = trainControl(method = "oob"),
                importance = "impurity")
soccer_data_pred2 <- soccer_test_data |>
  mutate(predicted_position = predict(modelrang, newdata = soccer_test_data))

c<-confusionMatrix(data = as.factor(soccer_data_pred2$predicted_position),
                   reference = as.factor(soccer_data_pred2$main_position))
```

2.4 K-nearest neighbor

```r
set.seed(12142024)
soccer_train_data_scaled <- soccer_train_data |>
  mutate(across(where(is.numeric), scale))

soccer_test_data_scaled <- soccer_test_data |>
  mutate(across(where(is.numeric), scale))

modelknn <- train(factor(main_position) ~ .,
                data = soccer_train_data_scaled |>
                  dplyr::select(-Player),
                method = "knn")
```

```r
soccer_data_pred3 <- soccer_test_data_scaled |>
  mutate(predicted_position = predict(modelknn, newdata = soccer_test_data_scaled))

c<-confusionMatrix(data = as.factor(soccer_data_pred3$predicted_position),
                   reference = as.factor(soccer_data_pred3$main_position))
```

2.5 Treebag

```r
set.seed(12142024)
modeltree <- train(factor(main_position) ~ .,
                data = soccer_train_data |>
                  dplyr::select(-Player),
                method = "treebag")
soccer_data_pred4 <- soccer_test_data |>
  mutate(predicted_position = predict(modeltree, newdata = soccer_test_data))

c<-confusionMatrix(data = as.factor(soccer_data_pred4$predicted_position),
                   reference = as.factor(soccer_data_pred4$main_position))
```

2.6 LDA

```r
set.seed(12142024)
modellda <- train(factor(main_position) ~ .,
                data = soccer_train_data |>
                  dplyr::select(-Player),
                method = "lda")

soccer_data_pred5 <- soccer_test_data |>
  mutate(predicted_position = predict(modellda, newdata = soccer_test_data))

c<-confusionMatrix(data = as.factor(soccer_data_pred5$predicted_position),
                   reference = as.factor(soccer_data_pred5$main_position))
```

3 Supervised Learning Comparison

```r
model_results <- data.frame(
  Model = c( "Ranger", "RandomForest", "Treebag", "LDA", "Knn"),
  Accuracy = c(0.5074, 0.4559, 0.4265, 0.4265, 0.3676),
  Kappa = c(0.4449, 0.3892, 0.3565, 0.3506, 0.2847),
  'Total Accuracy' = c(0.5956, 0.5368, 0.5074, 0.5441, 0.4706)
)

model_results %>%
  kbl(caption = "Model Metrics Comparison Table") %>%
  kable_classic() %>% kable_styling(latex_options = c("hold_position"))
```

Table 4: Model Metrics Comparison Table

Model	Accuracy	Kappa	Total.Accuracy
Ranger	0.5074	0.4449	0.5956
RandomForest	0.4559	0.3892	0.5368
Treebag	0.4265	0.3565	0.5074
LDA	0.4265	0.3506	0.5441
Knn	0.3676	0.2847	0.4706

4 Model Results

After implementing five different models of classification from our Stats Learning course it is seen that Ranger's random forest had the highest accuracy when identifying English Premier League players primary position and not only does it have the highest accuracy but a good measure of if that accuracy is meaningful is the kappa value. The kappa of 0.4449 indicates that there is moderate agreement and that the model is doing much better than random guessing. The Ranger model also had the highest "total accuracy" this variable was calculated by taking the model prediction of a players position and for that given player seeing if their primary, secondary, or tertiary position was correctly predicted by the model. This allows us to further assess how well the model performed at understanding positionality. As seen the Linear Discriminant Analysis model was not great at predicting a players primary position but was second best when taking non-primary positions into account.

5 Inference

The Ranger Model was the most accurate model in terms of predicting a player's position given their statistics, height, and weight. Our process of using Ranger's random forest was that a model with high accuracy has learned how to categorize a player's position. Such that if a player is incorrectly classified by our model with high accuracy, it can be inferred that the incorrect prediction is actually the position they have an affinity for. While the Ranger's total accuracy of 59.56% suggests moderate success, both Ranger and SVM showed similar patterns in their

Table 5: Players Who's Main and Alternate Positions Were Not Caught by Ranger

Player	predicted_position	Main_Position	Position_2	Position_3
Scott McTominay	ST	CDM	CM	-
Jack Hinshelwood	ST	CDM	-	-
Aaron Cresswell	CDM	LB	CB	-
Jimi Tauriainen	RW	CM	CDM	-
Thomas Partey	CB	CDM	CM	-
Matty Cash	CDM	RB	RWB	-
Omari Kellyman	ST	CAM	RW	LW
Boubacar Traoré	LB	CDM	CM	-
Ben Godfrey	CM	CB	LB	RB
Matt Ritchie	ST	LM	LB	-
Stefan Bajcetic	RW	CDM	CM	-
Facundo Buonanotte	CM	RM	CAM	-
Daniel Jebbison	GK	ST	-	-
Jack Grealish	ST	LW	LM	-
Carlos Baleba	LB	CM	-	-
Marc Cucurella	CDM	LB	CB	LWB
Andrew Robertson	RB	LB	-	-
Angelo Ogbonna	GK	CB	-	-
Paul Dummett	CDM	CB	LB	-
James Shea	CDM	GK	-	-

predictions - excelling at identifying specialized positions like goalkeepers (88.9% in SVM) but struggling with fluid midfield roles. This consistent pattern across different algorithms reinforces that certain position transitions are more natural in football, aligning with what our hierarchical clustering revealed about position similarities.

The table output shows players that were not correctly identified by any of their positions, which could indicate either model limitations or, more interestingly, players whose statistical profiles suggest untapped positional versatility. Our analysis suggests that while machine learning can help identify positional patterns, the fluidity of modern football makes strict positional classification challenging - a finding supported by both our supervised and unsupervised learning approaches.

6 Conclusion

Conclusion Our study of positional flexibility in Premier League players uncovered significant insights while highlighting important limitations in our approach. The primary constraint was our dataset size of 445 players spread across 12 positions, leading to notable class imbalance issues. This was particularly evident in positions like Left Midfield and Right Midfield, where only 14 samples were available for training. The use of single-season data (2023/2024) also limited our ability to capture player development and position transitions over time, while the aggregated nature of our statistics failed to account for in-game positional fluidity and tactical variations.

The SVM model's performance (47.4% accuracy) was constrained by several factors. The linear kernel struggled to capture complex non-linear relationships between performance metrics, while the small dataset made hyperparameter optimization challenging. The multi-class classification task was difficult given the overlapping characteristics of certain positions. Similarly, the Ranger model, despite achieving a higher accuracy, showed limitations in handling class imbalance and potentially overfitting to dominant position classes.

To enhance future research in this area, several improvements could be implemented. The dataset could be expanded to include multiple seasons of data or leagues, incorporating additional metrics such as heat maps, passing networks, and defensive coverage. Contextual variables about team tactics and opposition quality would provide valuable insights, while minute-by-minute positional data could offer more granular analysis than aggregated statistics. Methodologically, implementing hierarchical classification based on the position groups identified in our clustering analysis could improve model performance. Techniques like associating weights could address class imbalance issues, while ensemble methods might better combine the strengths of multiple modeling approaches.

The validation approach could also be strengthened by cross-validating findings with actual position changes in subsequent seasons, and utilizing transfer market data to verify positional flexibility predictions. While our models showed promise in identifying positional patterns, particularly for specialized roles like strikers, the results underscore that the complexity of modern football positions requires a more sophisticated approach combining both data supported insights and domain expertise.

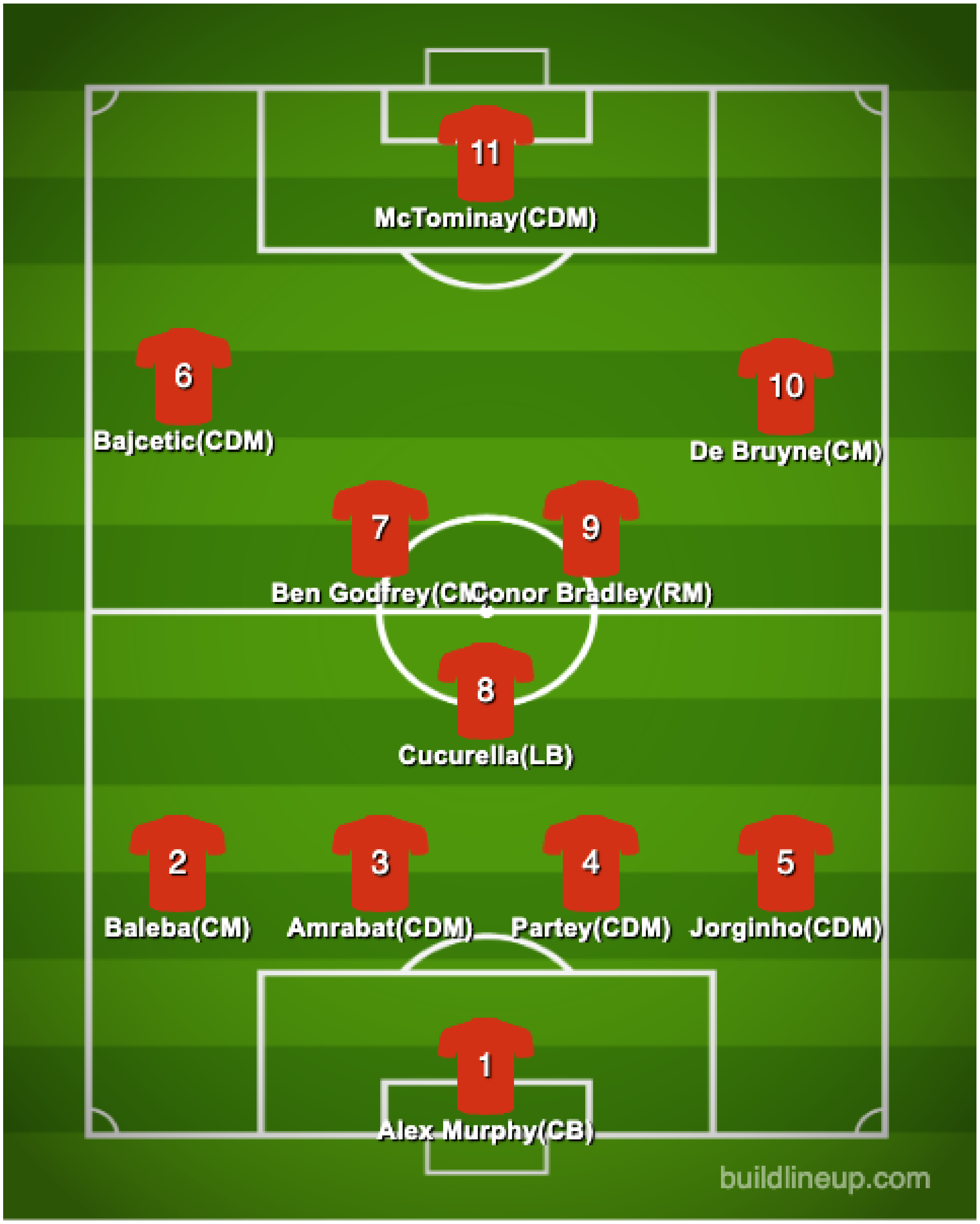


Figure 1: Ranger-Driven Alternative XI: Players in Their Model-Predicted Positions