Temporal Match Analysis and Recommending Substitutions in Live Soccer Games

Yuval Berman
Department of Mathematics and Statistics
University of Western Australia
yuval_berman@hotmail.com

Sajib Mistry, Joby Mathew, and Aneesh Krishna
Discipline of Computing, School of EECMS
Curtin University
{sajib.mistry, a.krishna}@curtin.edu.au, joby.mathew@postgrad.curtin.edu.au

***Abstract*—Soccer is one of the most complex and dynamic games. It is challenging to figure out the game's pattern in real-time. We propose a novel network metric and entropy-based live soccer analytic framework (NMELSA) that identifies the opponent team's tactics in a live soccer match by observing all the events until the specified minute of the game. We design a live game replacement model which recommends substitute players based on the on-field players' live game ratings. Experimental results on a real-world dataset demonstrate the efficiency of our proposed approach.**

***Index Terms*—Feature detection, Live soccer analysis, Event stream data, Forest Deep Neural Network, Temporal Clustering**

I. Introduction

A soccer match is a tumultuous and indeterminate dynamic system subject to unpredictable forces and changes. The dynamicity complicates identifying and analysing changes in such a system. Understanding how a soccer match develops and how making a micro-scale alteration (e.g., substituting a specific player at a specific time) affects the macro-scale (game trajectory/outcome) is extremely beneficial to soccer clubs and stakeholders. Because of the game's dynamic nature, coaches are usually in charge of making decisions. We offer two unique automatic replacement models based on machine learning techniques in this work, which can help a team's chances of winning.

Automatic service classification and clustering are critical in the discovery, selection, and composition of services [1]. Machine learning has recently become popular in the classification of services. Although promising findings have been produced, previous methods have only been evaluated on datasets with small-scale and balanced data, limiting their real-world applications. We solve this classification and clustering problem in this study with observable categories and unbalanced data.

In the last decade, the usage of analytics services in sports science has increased dramatically [2]. Soccer teams operate in high complexity, i.e., they are interdependent groups working towards a common objective, with the nature and efficacy of their interactions frequently deciding their success or failure. It's also because additional, more sensitive data is now available. This data records events such as passes, dribbles, and tackles that occur throughout a game and tags them with the location and time of the incident [3]. Such a dataset is known as *event stream* data. It can be used to perform a wide variety of machine learning powered service discovery, from individual players' performance, to creation and analysis of passing networks and services [2], [4], [5] and even detecting team tactics [6]. From this data, it is possible to construct *passing networks*, composed of nodes (players) and links (passes between players), whose organisation is in no way random and often carefully planned by a team's master tactician: the coach. The qualities of soccer passing networks, as well as the services they can provide, are constantly evolving during a match [2], according to research. Clustering, betweenness, degree centrality, and entropy are among the network metrics that have been analysed. In this research, we adopt a network approach to create more player-specific information, which we then use as a guide for our substitution models.

The use of machine learning for service discovery in soccer analytics has also seen a surge in the last decade. The majority of this work is based on historical data and either revolves around developing player ratings to help give teams an advantage when constructing their squads [7]–[9], or for accurate match outcome predictions [10]–[12]. Decroos et al. [6] used hierarchical agglomerative clustering methods to cluster similar plays together and were able to infer some team tactics relating to goal kicks, corners and set pieces. Pappalardo et al. [7] used a machine learning approach to rank players, with their approach being based on evaluating statistical features from event stream data for each player, which are then used to learn feature weights in a supervised learning framework (relative to match outcome). The authors then used the learned weights to compute a player's rating.

Garcia et al. [13] determined the technical-tactical behaviours of players according to position and checked the capacity of machine learning services to discover the most influential variables in each of the positions, and Gyarmati et al. [14] defined service to find the similarity of players by extracting the movement characteristics of the players and identify potential candidates to replace the given player from other teams. Both these could have been extended to create substitution models; however, to the best of our knowledge, no automated substitution models have been deployed in soccer

Corresponding Author: Sajib Mistry, School of EECMS, Curtin University, sajib.mistry@curtin.edu.au

978-1-6654-8143-4/22/$31.00 ©2022 IEEE
DOI 10.1109/ICWS55610.2022.00066

matches prior to this paper.

This paper proposes the usage of live event stream data to create the service-oriented machine learning model to create passing networks and evaluate players' performance according to their performance in six key network metrics and the features transformed from the event stream data. As the game progresses, these aspects can be updated, resulting in aggregate player and team measurements, informing the next services.

We use these player features and network metrics to predict match outcomes using state-of-the-art machine learning service-oriented algorithms. We then develop two novel substitution models which suggest a substitution to the coach and show that these methods both correlate significantly to actual substitutions made by the coach in the game.

The key contributions of the paper are summarised below:

- Clustering a similar pattern of gameplay to identify possible tactics implemented by the opponent through the addition of network-based metrics.
- Novel substitution model that offers substitutions in a live game using event stream data.

The source codes are available online[1] to evaluate the correctness of the approach and for easy reproducibility.

II. Related Work

While there has been a lot of research into soccer analytics since the millennium, valuing players with a complicated feature space of actions and locations is still relatively new. Memmert et al. [15] tried to analyse the tactics involved in the game by measuring inter-player coordination, inter-team coordination and team-team interaction. The authors used compactness as a primary measuring point and service in determining the tactics. Bialkowski et al. [16] made use of a service to analyse team styles by identifying the formation of the team. The authors used the tracking data and added a label for the player's position in that frame.

Grunz et al. [17] made use of an extension service of self-organising maps known as Dynamically Controlled Network to recognise tactical patterns in soccer games by analysing the tracking data. Kempe et al. [18] improved further on this study and applied the merge self-organising map to the tracking data. In comparison with Dynamically Controlled Network, the authors obtained better results. However, all these methods were used on tracking data, which involved a frame-by-frame dataset of the game that depicts the players' clear position and the ball at a given instance. Research by Gyarmati et al. [19] presented a service methodology used to compute the similarity of the players by extracting the movement characteristics of the players and identifying potential candidates to replace the given player. A major drawback of this technique is that the similarity-based substitution service model may not align with the dynamics of tactical formations. For example, if the team changes its tactics from attacking to defensive, we may have to replace a striker with a defender.

[1]https://github.com/yuval-b/soccer

Decroos et al. [9] used the event data to study the tactics involved by clustering all the data based on a specific pattern phase that occurred during the game. A phase is an action sequence that starts from a player and ends eventually with a goal or a shot attempt when the ball goes out of play like a corner or throw-in and when the referee stops the game. These sequential phases were divided into 10 clusters that depict the team's type of ball movement and attack. Thus, the service model discovered interesting tactics from the event data. Haaren et al. [20] also used a similar service for automatically detecting the strategy from the event data.

Kusmakar et al. [21] used a sequencing service like Decroos et al. [9] which was used to predict the outcome of a segment of the game. This was very useful in the dynamic analysis of the game as all the outcomes of these segments were further extended to predict the game's outcome. Guiliang et al. [22] made use of an adaptive service called the Deep Reinforcement Learning model to learn an action-value Q function from the event data of the match. This relied heavily on the past data of the game. Furthermore, there is no use case for them in a live game. Here, we aim to investigate player-to-player interactions in place to direct the creation of a dynamic planning service model capable of identifying and predicting the opponent team's tactics.

The existing approaches mainly focus on a sequence as a discrete collection of ten or fewer actions in the lead up to goals and assign values in a descending fashion from the moment of goal time reverse. To the best of our knowledge, the population of actions and locations as potentially influential on the match outcome are yet to be considered. Moreover, commonalities in player actions and locations and their relation to success across a pool of a large real game dataset are yet to be analysed. To the best of our knowledge, this is the first approach in temporal match analysis and developing substitution recommender systems in live soccer games.

III. Network Metric and Entropy-based Live Soccer Analytic Model (NMELSA)

A. Passing Networks

We start by creating weighted, directed passing networks from this events dataset, mostly using the pass, shot and free-kick event types to record interactions between two players on the team. So if a pass was recorded as an event by player A, and the following event (after some data manipulation to ensure exceptions were handled appropriately) was committed by player B, then we can assume the pass was successful, and we draw a connection (edge) going from the node of player A to player B (hence the directionality). The weight of each edge is how many times there was a pass from player A to player B. We can hence create passing networks for all teams in all matches.

As an example, figure 1 shows the passing network for the first half of the game between Real Madrid and FC Barcelona on Dec 23, 2017. Throughout the paper, we will use this game as a case study to make examples and explain the methodology.

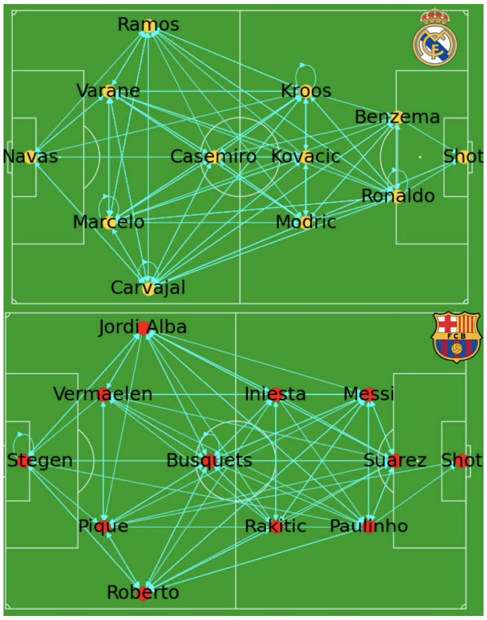

Fig. 1. Passing network for the first half of the El Classico between Real Madrid (top) and FC Barcelona (bottom).

Furthermore, we can break these down into temporal networks, which provide insight into how a team was playing within a small amount of time (e.g. 10 minutes) and infer insights about their playing styles.

We generate a passing network from the first 10-minute's worth of events data to analyse how the game evolves. We then iterate this window forward by 30 seconds and generate a passing network again. We repeat this process until we reach the 90-minute mark.

We calculate six key network metrics for each team's passing network created for each window. These reflect network metrics used in other network studies of soccer or similar games [3], [23]. These six metrics are explained below, and all of them except entropy were calculated using the library networkX 2.6.3, and its algorithms [24].

Mean out-degree centrality of a network is the mean number of players a player has passed to. It is independent of weight, but depends on the direction. Its calculation is shown in equation 1, where k_i^{out} is the out degree of a node i, in an N node network [25],

$$\langle k_{out} \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i^{out}. \tag{1}$$

Betweenness centrality is the extent to which the flow of passes through a team's network depends on that particular player [23]. The betweenness centrality of a node i is the sum of the fraction of all-pairs shortest paths that pass through i [26], as shown in equation 2,

$$c_B(i) = \sum_{s,t\epsilon V} \frac{\sigma(s,t|i)}{\sigma(s,t)}, \tag{2}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest paths between nodes s and t, and $\sigma(s,t|i)$ is the number of those paths passing through a node i which is not s or t. Here, we use the weighted edges, with more weight between two nodes corresponding to a shorter path. To get the mean betweenness centrality, we get the average of the 11 players.

Clustering Coefficient of a network is the extent to which the set of players a particular player passes with also pass with each other. For unweighted graphs, the clustering of a node i is the fraction of possible triangles through that existing node. In this case, we use the unweighted network to characterise connections that occur in the network. For a node i with degree k_i, the local clustering coefficient is defined as:

$$C_i = \frac{2L_i}{k_i(k_i-1)} \tag{3}$$

Where L_i is the number of edges between the neighbours of node i [25]. The global clustering coefficient is then the mean of all local clustering coefficients.

Closeness centrality measures how well-connected and central a player is within their team's passing structure. It does this by measuring the average shortest path length from that player to all other players. It is calculated as the average shortest path distance to i overall $n-1$ reachable nodes:

$$C(i) = \frac{n-1}{\sum_{i=1}^{n-1} d(j,i)}, \tag{4}$$

where $d(j,i)$ is the shortest path distance between j and i and n is the number of nodes that can reach i [23]. Since our network is directed, we calculate the outward and inward closeness for each node and average them out for the overall closeness centrality.

Eigenvector centrality standard deviation can be treated as an asymptotic probability a player would have the ball. Therefore by calculating the standard deviation of eigenvector centralities, we can infer how much a team shares the ball between its players [23].

The eigenvector centrality for node i is the i-th element of the vector defined by the following equation:

$$Ax = \lambda x \tag{5}$$

Where A is the adjacency matrix of the network with eigenvalue λ. By the Perron–Frobenius theorem, there is a unique solution x, all of whose entries are positive, if λ is the largest eigenvalue of the adjacency matrix [27].

entropy of a network is a measure of the unpredictability of a team in terms of passing between different players, defined similarly to conventional information theory entropy. This is defined in Small's 2013 paper [28], with the entropy of a single node being where $P(i,j)$ is the probability that a pass occurred between player i and j:

$$\mu(i) = \sum_j P(i,j) \log(P(i,j)), \tag{6}$$

We calculate the metrics on each network created for a ten-minute window. For each given game, we compare the results between the two teams playing, creating six time-series of how the metrics comparatively evolved between the two teams throughout the game.

Figure 2 shows the time series of three of the six network metrics for the Barcelona vs Real Madrid game. In the case of the entropy and closeness time series, Barcelona goals often correlate with a period of Barcelona dominance in those metrics.

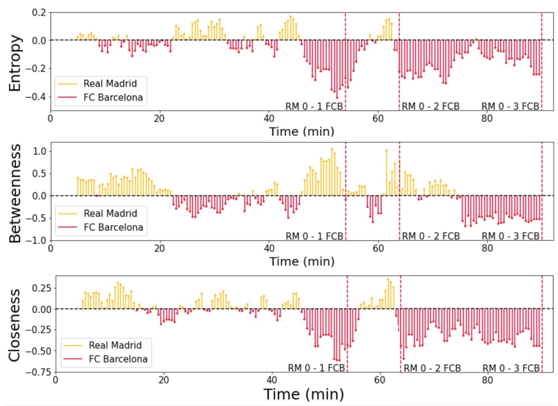

Fig. 2. Timeseries of three of the six network metrics for the Barcelona vs Madrid game. The dashed lines show Barcelona goals.

B. Match Prediction from Network Metric Timeseries

Based on the results above and in figure 2, we decided to investigate using these multivariate time series for each game to predict match outcomes. We use three well known Machine Learning algorithms for this.

Time Series Forest is a random forest classifier for time series. A random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. This transformer extracts three features from each window: the mean, the standard deviation and the slope. The total number of features is thus equal to $3*w$, where w is the number of windows. Then a random forest is built using these features as input data. This was implemented using the library `pyts` version 0.12.0 [29]. A MultivariateClassifier wrapper was added to it so that it was able to take in all six metrics time series as input.

Column Concatenator Classifier is a method developed by `sktime` (version 0.9.0) [30] to deal with multivariate time series- it is a transformer which concatenates multivariate time series/panel data into long univariate time series by simply concatenating time series in time. We then feed it into a time series forest again.

ROCKET (Random Convolutional KErnel Transform) is another method in the `sktime` library, first introduced by Dempster et al. [31], which transforms time series using random convolutional kernels, and uses the transformed features to create a linear classifier, thus being computationally cheaper than existing methods and still achieving the state of the art accuracy.

C. Feature Extraction and Prediction Modelling

A novel machine learning model which creates thousands of features based on pitch location and events that occurred at each location and the outcome of the event is used here. It then aggregates the feature measures for each player in each game and uses three machine learning algorithms to classify the aggregates for all features for each team towards a win, loss or draw. We consider the following feature-based learning approaches:

Random Forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting, implemented by `scikit-learn` version 1.0.2 [32].

Forest Deep Neural Network (fDNN) is a recently proposed algorithm equipped with a supervised forest feature detector that integrates a deep neural network. In the forest stage, supervised observations are used to fit the model. Then, predictions from each of the trees are fed into a fully-connected Deep Neural Network (DNN) [33].

xGBoost stands for Extreme Gradient Boosting and is an algorithm that combines decision trees with a boosting mechanism that effectively learns complex non-linear boundaries and performs feature selection when training, implemented using the `xgboost` library version 1.5.2 [34].

We extend the exiting classifiers with network metrics as new features. Rather than creating a time series of the metrics for the whole team across the whole game, we created networks of everything that happened before the 10-minute mark, 20-minute mark, and so on. We then calculate the described six metrics for each player for each game and append them to a database. We add the six network metrics (by merging data frames) to the existing feature aggregate data frames and then run the three state of the art classifier algorithms again.

D. Developing Player Ratings

It will be vital in our next step to create player ratings as a basis for substitutions. We first get a vector for the feature importance as determined by the classification algorithm to develop player ratings. First, the highest classification model is identified. Let us assume the model is X, where X can be any Random forest, FDNN or XGBoost. For example, X will be xGBoost's feature importance if xGBoost has higher predictive accuracy. Note that the prediction accuracy depends on the dataset and parameter tuning. We mainly focus on generalising analytical framework development; finding the best classifier is out of the research scope.

We develop player ratings based on multiplying the aggregate features (i.e. how often the players committed each feature) by the feature importance and summing that all together. We normalise it by making the best possible player rating for a given game as 94 and having 0 as having no aggregates.

IV. Team Tactics Clustering based Substitution Model

A. Maximising win predictions based on NMELSA

This first service model relies on the developed player ratings. We evaluate the player ratings of all players at the 60-minute mark and take the lowest performing player who is not a goalkeeper as our test case. Based on all player aggregates at that point, we also evaluate the team's chances of winning, losing, or drawing the match according to proposed NMELSA predictions.

Following that, we iterate over the players on the bench who aren't goalkeepers, developing what their aggregates would have looked like at the T mark based on their previous player aggregates. The prediction intervals are then run again to examine how they affect the team's odds of winning, losing, or drawing the game. The player with the greatest positive influence on outcome prediction is then recommended as a replacement for the player with the lowest rating.

This model is perhaps best illustrated by an example. Consider the Real Madrid vs Barcelona game again. At the 60 minute mark, Thomas Vermaelen, a Barcelona defender, had a player rating of 35, the lowest on his team. Barcelona's current prediction intervals were a 7% loss, 24% draw, and 69% win, according to NMELSA prediction intervals. Iterating through the six players on the bench and what their aggregate features would have been like instead of Vermaelen's (by taking their historical average), we then run the prediction intervals of xGBoost on the Barcelona team again. We see that Aleix Vidal would have increased Barcelona's chances of winning whilst simultaneously decreasing their chances of losing by the largest amount had he been playing. With Vidal playing the last sixty minutes instead of Vermaelen, Barcelona's outcomes were 3% loss, 15% draw, and 82% win. As a result, the model suggests substituting Vermaelen and putting Vidal on for the best chances of winning.

B. Team Tactics Clustering

The second model adopts a different service-oriented approach. Firstly, we normalise the results of all features—meaning each player at each game can have a feature score between 0 and 1 for each feature. We then multiply the normalised features by the xGBoost feature importance derived earlier. This is done to value each feature (dimension) appropriately in the next steps. We now have fingerprint representations of each game, which give us information about how a team likes to play and their tactics.

Hence, we use kMeans [32] as the automatic service classification to classify each game into one of seven clusters based on the result in all the features. These data points and clusters exist in N-dimensional space, with N being the number of features used.

Now that every team in every game is allocated to one of seven clusters, we take every match up of clusters that occurred. For every match up, we award the winning cluster 3 points, the loser 0 points, and if it's a draw, both clusters get 1 point. To deal with differences in the number of matchups between clusters, we take the point ratio, which is the points won divided by the total possible points (i.e. number of matchups times 3). The cluster that scores the highest against a particular cluster is then the *desired* ideal cluster.

V. Experiments

A. The Data

The dataset used is the largest open collection of soccer logs ever released [35], containing all the spatio-temporal events (passes, shots, fouls, etc.) that occurred during each match for an entire season of seven prominent soccer competitions (Spanish first division, Italian first division, English first division, German first division, French first division, European Cup 2016, World Cup 2018). The data was compiled by the proprietary firm Wyscout and has a total of 1,941 matches, 3,251,294 events and 4,299 players in multiple relational tables. The events table includes seven types of events, as is presented in Table I.

TABLE I
EVENT TYPES, SUBTYPES AND THEIR POSSIBLE TAGS RECORDED IN THE EVENTS DATASET.

Type	Subtype	Tags
pass	cross, simple pass	accurate, not accurate, key pass, opportunity, assist, goal
foul		no card, yellow, red, 2nd yellow
shot		accurate, not accurate, block, opportunity, assist, goal
duel	air duel, dribbles, tackles, ground loose ball	accurate, not accurate
free kick	corner, shot, goal kick, throw in, penalty, simple kick	accurate, not accurate, key pass, opportunity, assist, goal
offside touch	acceleration, clearance, simple touch	counter attack, dangerous ball lost, missed ball, interception, opportunity, assist, goal

B. Match outcome prediction accuracy

First, we tested the ability of those three algorithms, i.e., TSF, CCC, and ROCKET, to predict match outcome (win, draw, loss) from the six-time series input at different times of the match. Drawing on the five home and away leagues in the dataset (Spanish La Liga, Italian Serie A, German Bundesliga, English Premier League and French Ligue 1), we had a training dataset of 1551 matches and a test dataset of 387 matches.

We tested predictions at the 10, 20, 30, ..., and 90-minute marks of games- meaning we truncated the time series to only include the information from that amount of time. The results of the three algorithms can be seen in figure 3. We can see that, at best, the Time Series Forest and Column Concatenator Classifier methods achieve an accuracy of 54%, however, this is only at the 90-minute mark, i.e. after the entire game has been played.

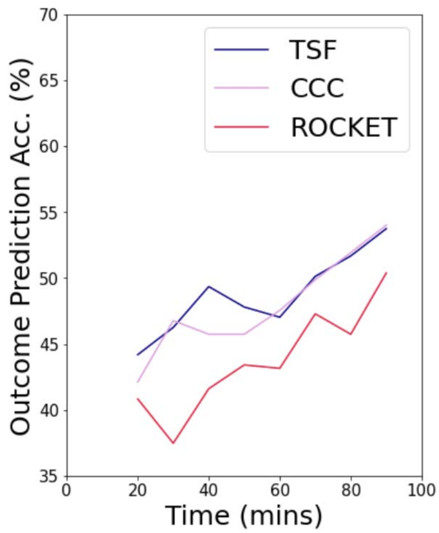

Fig. 3. Match outcome prediction accuracy of the three multivariate timeseries classifiers at certain times of the match.

We thus conclude that we can't rely solely on network metrics and their evolution throughout a game to predict match outcomes. To develop a robust and accurate model, we need some more ingredients.

Next, we compared the prediction accuracy of three feature-based algorithms, i.e., Random Forest, FDNN, XGBoost **extended with network matrices**. Taking the features developed and feeding them into the three algorithms they proposed, they achieved better match predictions, as shown in figure 4. The xGBoost and fDNN are the standouts, achieving around 65% accuracy at the end of the 90 minutes.

Finally, we compare the efficiency of the proposed network feature extension. The result is an improvement in two of the three algorithms. The random forest algorithm improved by 0.5% on average, and xGBoost improved by 0.8% on average (this is shown in figure 5). This is a significant improvement considering the uncertain nature of the soccer game and that now the xGBoost predictions peak at 67%. We can conclude that passing networks and network metrics have an important role in uncovering more and learning about the structure and game of soccer.

C. Efficiency of the proposed substitution model

In all matches studied, the average time for the first substitution was at the 58th minute. As a result, the service models

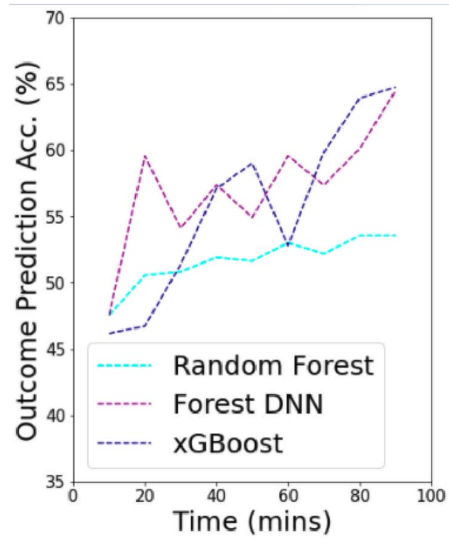

Fig. 4. Match outcome prediction accuracy using feature based algorithms

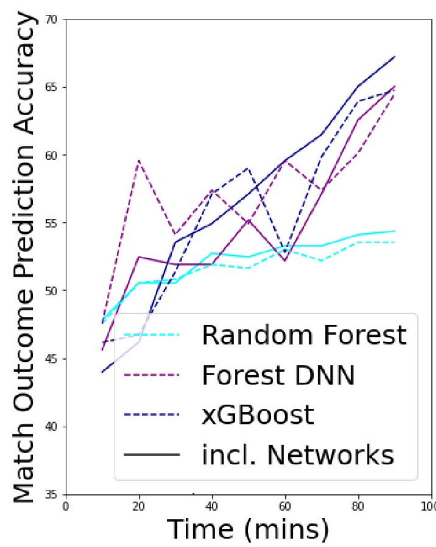

Fig. 5. Match outcome prediction accuracy using feature based algorithms (dashed lines), and the same algorithms but with network metrics included (solid lines).

implemented here are programmed to suggest the substitution based on the first 60 minutes' worth of data for a game and suggest substitution at the 60th-minute mark.

We have developed a service model that suggests substitutions that increase the team's chances of winning. On average, the service model improved the team's chances of winning according to xGBoost by 9% where a substitution was made (at times, no player on the bench improved the chances of winning). Interestingly, 14% of substitutions recommended by the model matched the naive model of simply substituting the lowest-ranked on-field player in terms of player ratings with the highest-ranked bench player. However, the model did display a high correlation to the coach's own opinions;

of the players recommended to be substituted off, 71% of them were substituted off at some point in that game by the coach, and similarly, 78% of the players recommended to be substituted on by the algorithm were substituted on at some point. These are vitalising results depicting that the service models can detect something about the game and the players' underlying performance, which the coaches also detect. It is also encouraging that there are still some disagreements between the service model and the coach, meaning that perhaps the service model can uncover some behaviour of that the coach is not yet aware of.

Table II shows how clusters matched up against one another. Note that the cluster labels are arbitrary and change between runs due to randomness. We choose seven clusters as it gives us a good amount of data for each cluster and its possible matchups, whilst still being a large enough value that we can characterise a team's tactics in several different ways. However, this cluster number is not set in stone and can be varied to examine the relationship between cluster numbers and the model's success.

TABLE II
THE DESIRED CLUSTERS TO BE IN BASED ON OPPONENT CLUSTERED GAME STYLE.

Opponent Cluster	Desired Cluster	Match-ups	Point Ratio
0	3	86	0.80
1	5	89	0.82
2	3	94	0.43
3	5	49	0.84
4	5	18	0.96
5	4	78	0.50
6	3	15	0.47

The service model follows that we now want to minimise our euclidean distance to the desired cluster's centroid. We iterate through every possible combination of substitutions: we check how close we would be to the desired cluster centroid if a given bench player had been playing instead of a given on-field player for the last 60 minutes. The algorithm then suggests the substitution which minimises the distance to the desired cluster centroid.

As an example, consider the Real Madrid vs Barcelona game again. Based on their current performances, both clubs are located in cluster 5. Looking at Table II, we see that Barcelona should aim to be in cluster 4 to maximise its chances of winning. Currently, the Euclidean distance to the desired cluster centroid is 9.50. By checking all bench players and all on-filed players, we see that the minimal Euclidean distance to the cluster 4 centroid is 8.78. This would happen if Denis Suárez were playing instead of Sergi Roberto. As a result, we suggest making this substitution.

Again, we see a high correlation in this model to the coach's own opinions; 53% of the players recommended to be substituted off were taken off by the coach, and conversely, 71% of players recommended to be substituted on actually were by the coach. It is likely that the percentage of players to be substituted off that matched the coach's opinion isn't as high as the model as this time, we checked all on-field players rather than just the lowest rating player, meaning some of the substitutions made may have been less obvious. The distance to the desired cluster centroid is reduced by 10% when making the recommended substitution.

While the proposed substitution model matches up well with the coach in its suggestions of who to substitute off and who to substitute on, the exact substitution percentage is considerably lower. Only 6.1% and 2.2% of the recommended substitutions occurred *exactly* as suggested for models 1 and 2, respectively (meaning the player out was the same as the player suggested *and* the player in was the same as suggested). This is likely because we have chosen not to include positional filters in our models, meaning the model could suggest substituting a forward for a defender. We wanted to start without positional filters to see how the model matched up and what was predicted. In the future positional filters could be added depending on the desired result. So in our Barcelona example, the Vidal for Vermaelen substitution suggested by the model didn't happen, and the D. Suarez for Roberto substitution didn't happen. However, there was a substitution of Vidal for Roberto, with Vidal going on to score Barcelona's third goal. This shows that the models and the coach were still in some agreement, even if it was not an exact one. In the future, services like this could be used in a live soccer match to act as a virtual assistant for the coach, supporting them in making crucial decisions and gaining an advantage over their competition.

A limitation of the proposed approach is that they are effectively 'substituting in the past'- meaning that they are checking what the team's chances of winning / cluster would be like right now if the suggested player had been playing the last 60 minutes, rather than how the game trajectory might change in the future. In the future, it would be ideal to develop models that have more forward vision and can predict how the game trajectory might change based on the substitution. Another limitation is that the model could be more 'context aware'; for example, depending on the score, a coach may substitute a defender for a forward to defend a lead rather than extend it. A model which is aware of contexts such as the score, red cards, and other important factors of the game would further extend the models developed here. Currently, the model is also only set up at the 60-minute mark, and while this can be changed easily to the game time desired, it means that it currently still considers players that had been substituted off before the 60-minute mark or have received red cards (meaning it may suggest to substitute them). These models were only deployed at the 60-minute mark as proof of concept. However, the service methodology is robust and can be used at any game point.

VI. CONCLUSION

We have demonstrated how a network metrics based analytic service may be used to augment an existing soccer analytics model and result in more accurate match predictions. We present an adaptive substitution model based on individual player characteristics and network measures that can be used in a live soccer match. The proposed NMELSA are designed using state-of-the-art machine learning methods like xGBoost, temporal clustering and data-driven network entropy. The proposed approach resembles genuine replacements performed by on-field coaches. Still, they also give coaches the ability to make data-driven decisions by utilising machine learning methods. This research can be further extended by making the NMELSA highly context-aware.

ACKNOWLEDGMENT

This initiative was funded by the Department of Defence and the Office of National Intelligence, Australia under the AI for Decision Making Program, delivered in partnership with the Curtin University and Pawsey Supercomputing Centre in Western Australia. The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] J. Zhang, Y. Chen, Y. Yang, C. Lei, and D. Wang, "Servenet-lt: A normalized multi-head deep neural network for long-tailed web services classification," in *ICWS*, pp. 97–106, 2021.

[2] J. H. Martínez, D. Garrido, J. L. Herrera-Diestra, J. Busquets, R. Sevilla-Escoboza, and J. M. Buldú, "Spatial and temporal entropies in the spanish football league: A network science perspective," *Entropy*, vol. 22, no. 2, 2020.

[3] J. Gudmundsson and M. Horton, "Spatio-temporal analysis of team sports," *ACM Computing Surveys*, vol. 50, p. 1–34, Jun 2017.

[4] F. Clemente, M. Couceiro, F. Martins, and R. Mendes, "Using network metrics in soccer: A macro-analysis," *Journal of Human Kinetics*, vol. 45, pp. 123–134, 03 2015.

[5] J. M. Buldú, J. Busquets, J. H. Martínez, J. L. Herrera-Diestra, I. Echegoyen, J. Galeano, and J. Luque, "Using network science to analyse football passing networks: Dynamics, space, time, and the multilayer nature of the game," *Frontiers in Psychology*, vol. 9, p. 1900, 2018.

[6] T. Decroos, J. Van Haaren, and J. Davis, "Automatic discovery of tactics in spatio-temporal soccer match data," in *ACM SIGKDD*, KDD '18, (New York, NY, USA), p. 223–232, Association for Computing Machinery, 2018.

[7] L. Pappalardo, P. Cintia, P. Ferragina, E. Massucco, D. Pedreschi, and F. Giannotti, "Playerank: Data-driven performance evaluation and player ranking in soccer via a machine learning approach," *ACM Trans. Intell. Syst. Technol.*, vol. 10, sep 2019.

[8] L. Bransen, J. Van Haaren, and M. van de Velden, "Measuring soccer players' contributions to chance creation by valuing their passes.," *Journal of Quantitative Analysis in Sports*, vol. 15, no. 2, pp. 97 – 116, 2019.

[9] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, "Actions speak louder than goals: Valuing player actions in soccer," in *ACM SIGKDD*, KDD '19, pp. 1851–1861, ACM, 2019.

[10] D. Berrar, P. Lopes, and W. Dubitzky, "Incorporating domain knowledge in machine learning for soccer outcome prediction," *Machine learning*, vol. 108, no. 1, pp. 97–126, 2018.

[11] E. Tiwari, P. Sardar, and S. Jain, "Football match result prediction using neural networks and deep learning," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 229–231, IEEE, 2020.

[12] A. C. Constantinou, "Dolores: a model that predicts football match outcomes from all over the world," *Machine learning*, vol. 108, no. 1, pp. 49–75, 2018.

[13] A. García-Aliaga, M. Marquina, J. Coterón, A. Rodríguez-González, and S. Luengo-Sánchez, "In-game behaviour analysis of football players using machine learning techniques based on player statistics," *International journal of sports science and coaching*, vol. 16, no. 1, pp. 148–157, 2021.

[14] L. Gyarmati and M. Hefeeda, "Competition-wide evaluation of individual and team movements in soccer," *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 144–151, 2016.

[15] M. D., Lemmink, K. A. P. M., and S. J, "Current approaches to tactical performance analyses in soccer using position data," *Sports Med*, vol. 47(1), p. 1–10, 2017.

[16] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Identifying team style in soccer using formations learned from spatiotemporal tracking data," *IEEE International Conference on Data Mining Workshop*, 2014.

[17] A. Grunz, D. Memmert, and J. Perl, "Tactical pattern recognition in soccer games by means of special self-organizing maps," *Hum Mov Sci*, vol. 31(2), pp. 334–343, 2012.

[18] M. Kempe, A. Grunz, and D. Memmert, "Detecting tactical patterns in basketball: Comparison of merge self-organising maps and dynamic controlled neural networks," *Eur J Sport Sci*, vol. 15(4), pp. 249–255, 2015.

[19] G. Liu, Y. Luo, O. Schulte, and T. Kharrat, "Deep soccer analytics: learning an action-value function for evaluating soccer players," *Data Mining and Knowledge Discovery*, vol. 34(5), pp. 1531–1559, 2020.

[20] J. Van Haaren, S. Hannosset, and J. Davis, "Strategy discovery in professional soccer match data," in *Proceedings of the KDD-16 Workshop on Large-Scale Sports Analytics*, pp. 1–4, 2016.

[21] S. Kusmakar, S. Shelyag, Y. Zhu, D. Dwyer, P. Gastin, and M. Angelova, "Machine learning enabled team performance analysis in the dynamical environment of soccer," *IEEE access*, vol. 8, pp. 90266–90279, 2020.

[22] G. Liu and O. Schulte, "Deep reinforcement learning in ice hockey for context-aware player evaluation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3442–3448, International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[23] C. Braham and M. Small, "Complex networks untangle competitive advantage in australian football," *Chaos (Woodbury, N.Y.)*, vol. 28, no. 5, pp. 053105–053105, 2018.

[24] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *The 7th Python in Science Conference* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11 – 15, 2008.

[25] A.-L. Barabási and M. Pósfai, *Network science*. Cambridge: Cambridge University Press, 2016.

[26] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[27] M. E. J. Newman, *Networks: an introduction*. Oxford; New York: Oxford University Press, 2010.

[28] M. Small, "Complex networks from time series: Capturing dynamics," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2509–2512, 2013.

[29] J. Faouzi and H. Janati, "pyts: A python package for time series classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020.

[30] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, "sktime: A Unified Interface for Machine Learning with Time Series," in *Workshop on Systems for ML at NeurIPS 2019*, 2019.

[31] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, p. 1454–1495, Jul 2020.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] Y. Kong and T. Yu, "A deep neural network model using random forest to extract feature representation for gene expression data classification," *Scientific Reports*, vol. 8, 11 2018.

[34] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[35] L. Pappalardo, P. Cintia, A. Rossi, E. Massucco, P. Ferragina, D. Pedreschi, and F. Giannotti, "A public data set of spatio-temporal match events in soccer competitions," *Scientific Data*, vol. 6, 10 2019.