Enhancing Live Commentary Generation in Soccer Video Games through Event Prediction with Machine Learning Methods

JAKUB KOŚCIOŁEK, University of Twente, The Netherlands

This paper addresses the challenge of enhancing live commentary generation in soccer video games through the prediction of in-game events using machine learning methods. Traditional prerecorded commentary systems fail to adapt dynamically to the evolving narrative of the game, often resulting in repetitive commentary. Attempts to solve this problem by integration of Large Language Models and Text-to-Speech technology generate additional challenges connected with overhead that is needed for those technologies resulting in noticeable delay. This research focuses on mitigating these issues by leveraging Support Vector Machine (SVM) and Artificial Neural Network (ANN) models to predict events such as Goal Kicks, Free Kicks, Corners, and Throw Ins, seconds before they occur. Utilizing data from the Google Football Environment, we trained and tested these models, examining their performance. Our findings indicate that while there is potential in such a methodology, further improvements need to be made to ensure that the model is working well in real-world scenarios. This study provides a foundation for future improvements in real-time commentary generation, emphasizing the potential of machine learning to minimize delays in generated commentary, thereby enhancing the immersive experience of soccer video games.

Additional Key Words and Phrases: Live commentary generation, soccer video games, event prediction, real-time data analysis, Google Football Environment.

1 INTRODUCTION

In contemporary video gaming, the integration of commentary has become a defining feature, enriching the immersive experience across various genres such as soccer, basketball, racing, and combat sports. However, while prerecorded commentary has been a staple in enhancing gameplay, it comes with inherent limitations, particularly in dynamically adapting to the evolving narrative of the game.

Among sports video games, soccer simulations stand out for their intricate gameplay dynamics and the potential for immersive commentary experiences. This has been noted in the works of [13], [14], [6] and [1], which explore various approaches to optimize and enhance commentary generation in soccer game settings.

The work of [1], which utilizes Large Language Models (LLMs), has particular shortcomings due to the overhead created by LLMs and Text-to-Speech algorithms. This substantially delays the commentary generation to on average 6 seconds after the event has already happened, which negatively impacts the immersiveness and enjoyment of the video game experience.

Meanwhile, research has been done in the direction of event prediction and classification in soccer game settings. Papers such as [2], [9] or [10] present different approaches, solving problems from soccer highlights predictions to labeling soccer events.

TScIT 41, July 5,2024, Enschede, The Netherlands

These approaches can be leveraged as solutions to the delay problem from [1]. By integrating event prediction in soccer video games which provides extremely precise and deterministic data, we can anticipate the event seconds before it happens, removing or significantly shortening the delay in the commentary.

The aim of this research can be divided into two questions:

(1) What is the efficacy of different machine learning methods in accurately predicting soccer events using data extracted from soccer video games?
(2) Can machine learning models mitigate the delay in live soccer commentary generation?

To achieve this, we will focus on the Support Vector Machine model [15] and the Artificial Neural Network model presented in [10]. To train and test these models, we will use data from the Google Football Environment [5], an open-source environment to run soccer simulations, which provides easy ways to extract the data.

This paper is organized as follows. In Section 2, we examine related work. Section 3 explains the data foundation and its properties. In Section 4, we present the methodology, including an explanation of model choices, their parameters, and data preparation. Section 5 offers a comprehensive evaluation using predefined metrics. Finally, Section 6 addresses the questions posed above, the limitations of the system, and the possibilities for future work.

2 RELATED WORK

The foundation of this research lies in the field of live commentary generation. Significant contributions in this field include the works of [13], [6] and [14]. All of those papers focus on automating the commentary generation in the context of soccer video games by utilizing various machine learning methods. Despite the great results, they all pose the problem of repetitive commentary or a restricted scope of possible comments.

These limitations have been addressed in the work of [1], who intended to solve those issues by utilizing advancements in Large Language Models. Based on the Google Football Environment [5], he extracted game data and pared it into the predefined prompts that he later provided to the GPT-3.5 model [7]. Despite great results in terms of the accuracy of the generated commentary, it has posed a new challenge. Namely, events such as Goal Kicks or Free Kick, are noticed by the game only after they happen, which in combination with the overhead from LLM and Text-to-Speech processing time produce substantial delays.

This problem has led to an exploration of another branch of research related to predicting events and game outcomes in soccer. Notable studies include works by [9], [2] or [4] present interesting approaches to utilize different techniques combined with soccer game data to predict scores, winners, or game events. [9], developed

a computer vision-based Soccer Goal Predictor, which uses fine-tuned 3D convolutions to learn spatiotemporal information directly from videos and predict future goals 2 seconds in advance. It resulted in a precision of 76% while keeping the recall at 90%.

[2] created an algorithm called POGBA, which given an event stream containing 40 different types of events such as receiving the ball, shooting toward a goal, or fouling other players, can give the probability that a critical event occurs in the next predefined number of seconds. [4] used an L2-regularized logistic regression model with three feature sets to predict the winner of the soccer match, resulting in up to 84% accuracy.

Ultimately, the work of Richly et al. [10] presents an Artificial Neural Network model, which based on spatio-temporal data from real-live soccer matches is able to detect and classify events such as Kick and Reception up to 89, 90, and 89 percent for Precision, Recall, and F1-Score respectively.

In this work, we aim to address the problem of delay highlighted in [1] by implementing a predictive model based on the framework established in [10] and incorporating additional advanced machine learning techniques. Our model is designed to predict specific soccer events seconds before they occur. We operate under the assumption that, given the strong performance of existing models trained on imprecise data from real soccer matches, our model should achieve even greater results and reliability when applied to the highly precise and deterministic data provided by Google Football Environment [5].

3 DATA FOUNDATION

To train the models, extensive, high-quality data needs to be collected. For this purpose, we used a football simulation environment developed by the Google AI Team [5]. This is an open-source platform that provides us with tools to run soccer games and easily extract the current status of the game every 1 second (which we define as a 'data collection point'). An overview of the information provided by the simulation can be found in Table 1.

We gathered data by running 1247 games in the bot vs bot setting. Despite its limitations connected with under-representing certain events, bot vs bot games have the advantage of not requiring constant human interaction, giving the opportunity to collect more data in a shorter time.

To achieve our goal, we carefully selected features based on their relevance and statistical properties. We excluded Team Roles due to its low variance and Team Tired Factor because of the high correlation between players' fatigue levels. Additionally, we removed the Team Active, Active, and Designed properties as they showed low relevance to our desired outcome. Ultimately, we discarded Sticky Actions to reduce the size of the input layer, thereby decreasing the computational complexity required to train the models.

As the result, we have decided to use the following game information: (x,y,z) coordinates for ball position, ball movement vector, and ball rotation angle in radians, and (x,y) coordinates of each team's player position and each team's player movement vector. This information accounted for 97 distinct features per one data collection point.

To identify when the event happened, we used the *Game Mode* property consisting of the following modes: No Event (none of the events happened), Kick-off (the game has started), Goal Kick (the defending team is restarting play from their goal area after the ball has gone out of bound past the goal line), Free Kick, Corner, Throw In, and Penalty.

Additionally, to the *Game Mode* property, we can also identify when the goal is scored and a yellow card is given. This is possible by noticing changes in *Score* and *Team Yellow Card* properties.

Due to the limitations connected with the games that have been played only by bots, certain events like Goals, Yellow cards, Kick-offs, or Penalty have been under-represented, namely for every 400 games, we have experienced 5 Goals, 0 Yellow cards, 0 Kick-offs, and 1 Penalty. This restricted us to predicting only the following set of events: Goal Kick, Free kick, Corner, Throw In, and No Event.

Name	Description
Ball	[x, y, z] position of the ball.
Ball Direction	[x, y, z] ball movement vector.
Ball Rotation	[x, y, z] rotation angles in radians.
Ball Owned Team	Indicates which team owns the ball
Ball Owned Player	The index of the player who owns the ball
Team	[x, y] positions of players.
Team Direction	[x, y] movement vectors of players.
Team Tired Factor	The fatigue levels of players on the team, with 0 being fully energetic and 1 being completely tired.
Team Yellow Card	The number of yellow cards each player on the team has.
Team Active	Indicates whether each player on the team is active in the game
Team Roles	The roles of each player on the left team
Active	The index of the currently controlled player.
Designated	The index of the designated player, usually the one leading the game, like the ball owner.
Sticky Actions	A 10-element vector indicating which actions are currently active. Possible actions: left, top left, top, top right, right, bottom right, bottom, bottom left, sprint and dribble
Score	The number of goals scored
Steps Left	How many steps are left till the end of the match.
Game Mode	Current game mode, indicating the event that has been done. Possible actions: No action (none of the events happened), Kick-off (the game has started or restarted after a goal), Goal Kick (the defending team is restarting play from their goal area after the ball has gone out of bounds past the goal line), Free Kick, Corner, Throw In, Penalty.

Table 1. Description of the game state information

4 METHODOLOGY

4.1 Data preparation

To effectively use the models, comprehensive data preparation was essential. Initially, data collection points were flattened and grouped by classes (excluding the No Event class).

Since each of the *Game mode* values was registered for a certain duration, consecutive data collection points often recorded the same values. After the event happens, for example, a player takes a shot and misses, the game mode changes to the Goal Kick for the duration when the ball is out of play. To ensure that we register only the moment when the event happened (regarding labels that are not No Event class), we grouped consecutive occurrences and excluded ones that were not the first in the group.

As the model should predict the action happening in advance, we needed to map the event with the game information that happened a predefined time before. For that purpose we mapped each event to an interval of 3 data collection points, ending 3 data collection points before the event occurred. Given that each data collection point has 97 features, this process resulted in an array of 291 instances representing the merged features and one instance representing the label of the upcoming event.

Feature standardization was performed by removing the mean and scaling to unit variance [12]. This step ensured that all features contributed equally to the model training process.

Additionally, for the Richly et al. model, labels were one-hot encoded to transform the categorical target variables into a suitable format for training. This technique ensures that each class is represented as a unique binary vector in a multidimensional output space, facilitating the use of categorical cross-entropy as the loss function. This method enhances the model's ability to differentiate between distinct classes, thereby improving overall performance.

4.2 Model selection

4.2.1 Support Vector Machine (SVM). Support Vector Machine (SVM) is a robust supervised machine learning algorithm primarily used for classification and regression tasks. Originally introduced by Vapnik [15], SVM operates on the principle of finding the optimal hyperplane that best separates the classes in a high-dimensional space. This separation is achieved by maximizing the margin between the nearest data points of any class (support vectors) and the hyperplane, thus providing a clear distinction between classifications.

The effectiveness of SVM in this context stems from its ability to efficiently manage high-dimensional data. By utilizing kernel functions, the model can operate in a transformed feature space without explicitly computing the coordinates in that space, thereby avoiding computationally expensive processes. This capability makes SVM well-suited for our dataset, which comprises 291 features requiring rapid predictions.

For optimal SVM performance, key parameters such as the kernel type, the regularization parameter C, and kernel parameters like *gamma* need adjustment. The parameter C controls the trade-off between achieving low training data error and minimizing model complexity for better generalization, while *gamma* affects the granularity of the decision surface by defining the influence of individual training samples on the decision boundary. To identify the best parameter combination, a grid search has been implemented with the following parameters to be tested:

Evaluation of the grid search has been made with 5-fold cross-validation.

Parameter	Choices
C	0.1, 1, 10, 100, 1000
gamma	scale, auto, 1, 0.1, 0.01, 0.001
kernel	linear, poly, rbf, sigmoid

Table 2. Parameter Grid for SVM

4.2.2 Richly et al. model. The model proposed by Richly et al. [10] presents a significant advancement in event detection using artificial neural networks (ANNs) for analyzing soccer game data. This model is particularly adept at handling spatio-temporal data, identifying specific game events like passes or shots with high precision, recall, and F1-scores by using features derived from players' and ball's positional data [10].

Neural Network Architecture provided in the paper consists of three layers. For the input and hidden layer sigmoid activation function has been used and for the output layer, the softmax function. To optimize the model, a grid-search approach has been implemented to pick the best parameters for a number of hidden units, learning rate, and dropout value.

To imitate the above architecture, we have made some modifications. The size of the input layer has been increased to 291 to match our number of features, and the size of the output layer has been expanded to 5 to correspond to the number of proposed labels. Additionally, for the grid search, the number of hidden units to be tested has been adjusted to [64, 128, 256] to accommodate the larger input layer. The following presents all the parameters of the grid search:

Parameter	Choices
Hidden Units	64, 128, 256
Learning Rate	0.1, 0.05, 0.01, 0.005
Dropout Rate	0, 0.01, 0.05, 0.1, 0.2

Table 3. Parameter Grid for ANN

Evaluation of the grid search has been made with 5-fold cross-validation as well.

5 EVALUATION

5.1 Preliminaries

To evaluate the performance of our models, we will use the following metrics:

Metric 1: Recall - This metric indicates how well the model is at correctly identifying actual events for each label on average. It is the proportion of correctly predicted events out of all actual events, averaged for each label. A low recall score suggests that many actual events are being missed by the model.

Metric 2: Precision - This metric reflects how many of the events predicted by the model were correct. A low precision score indicates that many of the predicted events are incorrect.

Metric 3: F1-score - The F1-score, which is the harmonic mean of precision and recall, will help us compare the performance of the models based on a single value.

Due to the imbalance in our dataset, accuracy is not a reliable evaluation metric, as it is overly sensitive to such imbalances.

Given the nature of our task, we are less concerned with mistakenly labeling actual events (Goal Kick, Free Kick, Corner, and Throw In) as No Event than with the opposite error. If we incorrectly predict a No Event scenario when an actual event occurs, we can still apply the approaches from [1] to provide commentary with a delay. However, if we incorrectly predict an event when there is none, the model will generate commentary on a nonexistent event. Therefore, in addition to the above metrics, we will closely examine the confusion matrix to gain deeper insights into the data and associated issues.

By focusing on these specific metrics, we aim to gain a comprehensive understanding of our models' performance in identifying and classifying events, particularly in the context of the challenges posed by the imbalanced dataset.

5.2 Test dataset

We can distinguish 2 datasets prepared for the evaluation.

5.2.1 Balanced dataset. Firstly we have tested our models on the balanced dataset. We employed down-sampling to match the number of labels to the minority class, resulting in 572 samples per class.

Later, we used five-fold cross-validation, which splits the data into 5 different groups of randomly assigned training and testing sets. For each iteration metrics presented in the previous sub-section have been calculated and presented as an average. This provides us with more credible results, which are not influenced by the specification of the datasets that have been picked for the training and testing.

Given the imbalanced nature of our dataset, this may create results that are not the most representative if it comes to the proportion of occurrences of different events. Thus this dataset provides us a guideline to see if the given features are enough for the model to rightfully predict the events, disregarding big imbalances in occurrences. If at this point the model provides bad performance, we need to optimize the architecture or provide more high-quality data.

5.2.2 Imbalanced dataset. This dataset is based on 23 games, reflecting the true proportion between the labels (Table 4). This ensures that the performance metrics accurately represent the current scenario. Only by analyzing these results can we determine if the model is ready for real-world application.

Event	Count
No event	65059
Goal Kick	157
Free Kick	58
Corner	20
Throw In	9

Table 4. Proportion of the event count for imbalanced test dataset

Bad performance at this dataset, given good performance at balanced datasets indicates that the model is capable of good prediction, however, there needs to be more work done in accustoming it to the imbalances.

5.3 Results

5.3.1 Support Vector Machine (SVM). Initially, the grid search was performed with possible parameters that can be found in Table 2 The grid search determined that the best parameters for our SVM model were $C = 100$, $kernel = rbf$, and $gamma = scale$. This systematic exploration ensured the selection of the most suitable settings for the SVM model applied to our specific dataset.

After training the SVM model, we evaluated its performance using the metrics discussed in the previous sections. Table 5 provides a comparison of the recall, precision, and F1-score between balanced and imbalanced datasets.

Metric	Balanced	Imbalanced
Recall	0.79	0.53
Precision	0.798	0.9956
F1-Score	0.79	0.69

Table 5. SVM Performance Metrics

As demonstrated, the model trained on balanced data outperformed the one trained on imbalanced data in terms of F1-Score, with all metrics around 79%. The confusion matrix (Figure 1) indicates that most errors occurred in distinguishing between Free Kick and No Event, as well as Goal Kick and Corner.

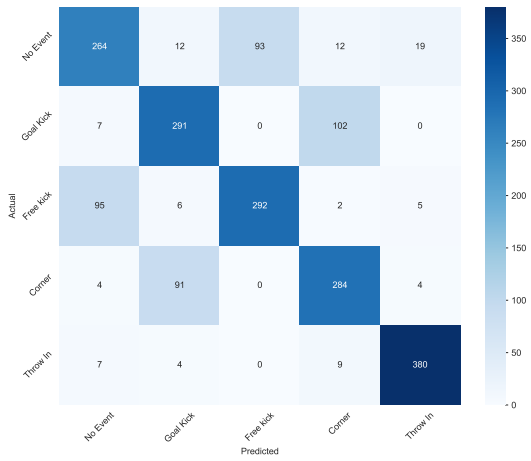

Fig. 1. SVM confusion matrix for Balanced Dataset

Free Kick events often arise from offside situations, which the model struggles to generalize due to the specific rule that the attacking player's x coordinate must be closer to the opponent's goal than any defender's x coordinates. Similarly, Goal Kick and Corner are frequently confused because their distinction depends solely on which team last touched the ball.

Given the systematic nature of these errors, targeted optimizations could potentially improve performance by better highlighting the differences between these classes.

Conversely, when evaluating the model on imbalanced data (Table 5), the results for Recall and F1-Score were slightly worse compared to SVM. However, the precision achieved an impressive score of 99%. A closer examination of the Confusion Matrix (Figure 2) reveals that the model actually performed poorly. Metrics such as Recall, Precision, and F1-Score are heavily influenced by the significant imbalance between the No Event label and other labels, with an average proportion of 2403 No Event labels to 1 of any other label.

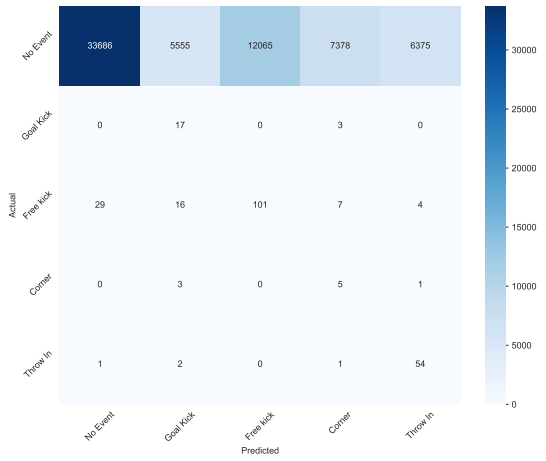

Fig. 2. SVM confusion matrix for Imbalanced Dataset

To address this issue, we explored different class weights for the SVM model, assigning weights based on the labels' proportions in the test dataset (Table 6). However, this approach did not yield significant improvements.

Label	Weight
No Event	7200
Goal Kick	2
Free Kick	15
Corner	1
Throw In	5

Table 6. Class weights for SVM model

5.3.2 Richly et al. model. Again we started training by performing the grid search to find the best parameters within Table 3. The grid search determined that the best parameters for the Richly et al. model were 256 hidden units, 0.005 learning rate, and 0.01 dropout rate. This ensured that the parameters were right for our dataset.

Regarding outcomes from Table 7, we can see that the Richly et al. model has performed slightly worse than SVM. For balanced data Recall and F1-Score are oscillating around 65% for balanced data, whereas precision went up to 75%. This may be caused by the simple architecture of the model, which is provided with much more complex data. SVM is a much simpler method concentrating on linearly separable data, whereas Neural Networks captures more complex, non-linear data.

Metric	Balanced	Imbalanced
Recall	0.627	0.408
Precision	0.756	0.995
F1-Score	0.68	0.576

Table 7. Richly et al. Model Performance Metrics

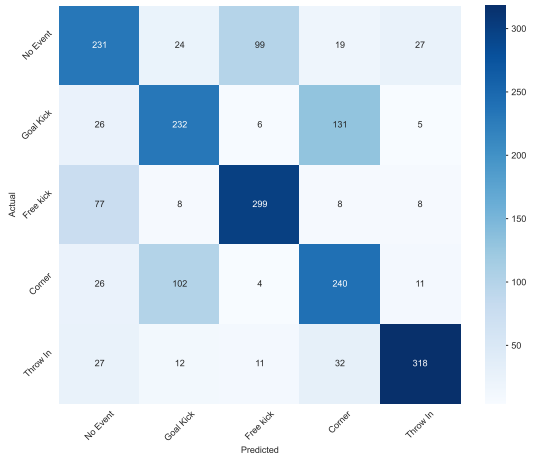

Fig. 3. Richly et al. model confusion matrix for Balanced Dataset

The problem with this model can be also the number of samples provided for each label. Neural Network models usually need big datasets to be able to generalize complex relations well. Interestingly, in their paper, the Richly et al. model performs well on only 68 events. This can be influenced by smaller label sets and less complex tasks (they do not predict events before they happen, they only classify them).

Seeing the Confusion Matrix for balanced data, we can see that this model struggles with distinguishing differences between Free kick and No event, and Goal Kick with Corner, as well. This is a strong indicator that differences between those labels are not only tight to SVM methodology, thus investigation into a way to distinguish them better can yield great results.

Further looking at the confusion matrix for imbalanced data (Figure 4) we can see similar problems as in the SVM model. The model performs badly, by predicting the events regarding regular frequency. That is why a lot of predictions have been wrong regarding the No Event label as other events.

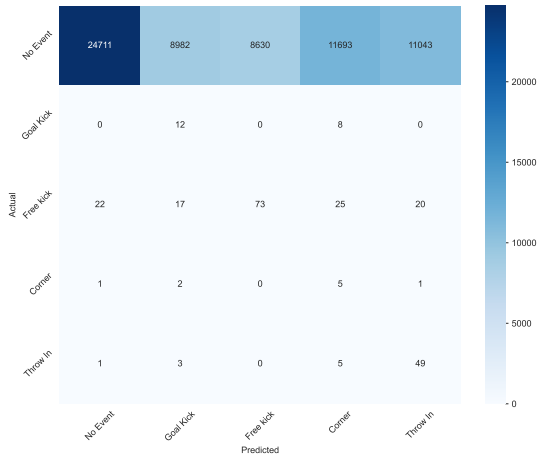

Fig. 4. Richly et al. model confusion matrix for Imbalanced Dataset

6 DISCUSSION

6.1 Limitations

Despite the promising results, several limitations must be acknowledged. First, the study relies solely on data from the Google Football Environment, which may generate the problem of low adaptability to other soccer video games given the different arrangements of the data. Secondly, the model has been trained only on bot vs bot games, which may not accurately reflect the complexities and unpredictable nature of human gameplay, potentially reducing the generalizability of our findings. Additionally, due to the under-representation of certain events such as goals, yellow cards, and penalties, we focused on predicting a limited set of events (goal kick, free kick, corner, and throw-in). This narrow scope may overlook the full spectrum of possible game events, limiting the applicability of our predictive models to more diverse and less frequent scenarios.

Moreover, each event in the game is detected for a certain duration, which adds complexity to our model's predictions. Our model operates under the assumption that the game has a mechanism to disable further predictions once an event has been detected for its entire duration. This assumption may not hold in actual game settings, potentially posing additional challenges. Integrating our model into the real-time dynamics of a live game could require significant adjustments to ensure that the model does not produce redundant or conflicting predictions during ongoing events.

6.2 Future work

To further develop this work, various approaches can be taken. These can be categorized into three main areas: data preparation, model enhancement, and the choice of a Large Language Model. The first two areas focus on refining the methodology presented in this paper, while the third explores potential future directions for reducing delays in the work of [1], thus extending beyond the current scope.

6.2.1 Data preparation. Changing the approach to data preparation is a critical area for improvement. One aspect to consider is experimenting with different detection timeframes. By merging a higher amount of data collection points, and adjusting the size of the window before an event happens, we can potentially capture more context and improve prediction performance.

Additionally, employing sampling techniques such as merging only every second or third data collection point could be explored. Consecutive points may exhibit too little variance to provide useful information, so this method can help reduce data redundancy and focus on more significant changes in the game state.

Different features and labels can also be selected to enhance model performance. Expanding the number of labels by incorporating underrepresented events such as yellow cards, goals, or penalties, despite their low occurrence, could provide a more comprehensive prediction model. Furthermore, features that would show better the distinction between No Event and Free Kick, as well as between Goal Kick and Corner could be added. Examples of such features include the information of who touched the ball last (*Designed* property from game state information Table 1) and the difference between the defenders' line x-coordinate and the attackers' x-coordinate.

Moreover, enhancing data collection by including player vs. bot games, rather than only bot vs. bot, can provide more realistic and varied data. This approach can help train models that better reflect real-world scenarios and player behaviors, thereby improving the robustness of the prediction system.

6.2.2 Model enhancement. Model selection and development are critical areas for future work. One significant challenge is addressing the poor performance of models on imbalanced data. Despite its strong performance on balanced data, the SVM model may lack adequate tools to address this issue. While further investigation into class weighting could be conducted, our initial efforts did not yield significant results.

Improving the neural network architecture for the Richly et al. model appears to offer substantial potential for enhancement. Given the model's excellent performance on simpler tasks and data, as described in [10], exploring architectural enhancements to handle more complex tasks and data could be highly beneficial. This might involve adding additional hidden layers and increasing the number of hidden units.

Finally, investigating models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) [11] is another promising direction. RNNs are a class of neural networks designed for sequential data, capturing temporal dependencies by maintaining a hidden state that evolves over time. LSTMs are a specialized type of RNN that addresses the vanishing gradient problem by incorporating memory cells and gating mechanisms, enabling them to learn long-term dependencies more effectively. These models are particularly useful for tasks involving time series, and any scenario where understanding the order and context of the data is crucial.

6.2.3 Large Language Model choice. Future research could explore the selection of more optimal Large Language Models (LLMs). While [1] utilized the GPT-3.5 model, advancements in LLM technology suggest that alternative models, such as GPT-3.5 Turbo or GPT-4

[8], could offer improved speed and reliability [3]. Additionally, developing smaller, specialized models trained specifically on soccer commentary data could further reduce processing times significantly. This tailored approach could enhance both the efficiency and accuracy of live commentary generation.

6.3 Conclusion

In this paper, we aimed to enhance soccer live commentary generation based on the work of [1] by decreasing the delay caused by the overhead of using Large Language Models (LLMs) and Text-To-Speech (TTS) algorithms. Given the average delay between an event and its commentary of 6 seconds, reducing this delay to even 3 seconds would yield significant improvements in gameplay enjoyment and immersiveness. To achieve this goal we utilized Support Vector Machine (SVM) and Neural Network (NN) models to predict the soccer events 3 seconds before they happen.

The results reveal the potential of our approach, with F1-Scores of 79% and 69% for SVM, and 68% and 57% for the Richly et al. model on balanced and imbalanced datasets, respectively. However, the performance on imbalanced data is greatly influenced by the disproportionate number of No Event labels compared to other events. This suggests a need for further investigation to make the models more suitable for real-world scenarios. Our approach to adjusting class weights for the SVM model did not provide any notable benefits.

The SVM model outperformed the neural network model, likely due to the relatively small sample sizes for each label (572) given the complexity of the task, and the simple architecture of the neural network, which might not be sufficient for learning complex patterns.

Interestingly, both models struggled with distinguishing between Free Kick and No Event, as well as between Goal Kick and Corner. This implies that there are high similarities between these labels.

Theoretically, this approach reduces the potential delay to just under 3 seconds, including a processing time of less than 1 second for SVM or NN predictions. However, inaccuracies in predictions introduce additional challenges by generating incorrect commentary. This indicates that while the delay in commentary generation can be mitigated, it comes with the trade-off of potentially increased errors.

If further advancements in the data preparation (6.2.1) and model enhancement (6.2.2) do not significantly improve model performance, particularly with imbalanced, real-world data, alternative approaches must be considered. One potential direction involves the selection of a more suitable Large Language Model (6.2.3).

7 APPENDIX: USE OF AI TOOLS

During the preparation of this work, the author used ChatGPT in order to get assistance during the process of programming the experiments, checking the grammatical structure of the sentences, structuring the LaTeX file, and Grammarly for spellcheck. After using these tools, the author reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

8 REFERENCES

[1] Michał Czaplicki. "Live commentary in a football video game generated by an AI". en. In: (2023).

[2] Tom Decroos et al. "Predicting Soccer Highlights from Spatio-Temporal Match Event Streams". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (Feb. 12, 2017). Number: 1. ISSN: 2374-3468. DOI: 10.1609/aaai.v31i1.10754. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10754 (visited on 05/21/2024).

[3] GPT for Work. *OpenAI API and Other LLM APIs Response Time Tracker*. 2024. URL: https://gptforwork.com/tools/openai-api-and-other-llm-apis-response-time-tracker.

[4] Matthew G. S. (Matthew George Soeryadjaya) Kerr. "Applying machine learning to event data in soccer". Accepted: 2016-01-04T19:58:11Z Journal Abbreviation: Application of machine learning technique to discover useful knowledge from event data in soccer. Thesis. Massachusetts Institute of Technology, 2015. URL: https://dspace.mit.edu/handle/1721.1/100607 (visited on 05/21/2024).

[5] Karol Kurach et al. "Google Research Football: A Novel Reinforcement Learning Environment". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04 (Apr. 2020), pp. 4501–4510. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v34i04.5878. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5878 (visited on 05/07/2024).

[6] Greg Lee, Vadim Bulitko, and Elliot Ludvig. "Automated Story Selection for Color Commentary in Sports". In: *Computational Intelligence and AI in Games, IEEE Transactions on* 6 (June 1, 2014), pp. 144–155. DOI: 10.1109/TCIAIG.2013.2275199.

[7] OpenAI. *OpenAI*. Accessed: 2023-06-17. 2015–2023. URL: https://openai.com.

[8] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[9] *Predicting soccer goals in near real time using computer vision | AWS Machine Learning Blog*. Section: Amazon SageMaker. Dec. 8, 2020. URL: https://aws.amazon.com/blogs/machine-learning/predicting-soccer-goals-in-near-real-time-using-computer-vision/ (visited on 05/21/2024).

[10] Keven Richly, Florian Moritz, and Christian Schwarz. "Utilizing Artificial Neural Networks to Detect Compound Events in Spatio-Temporal Soccer Data". In: Aug. 2017.

[11] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2019.132306. URL: https://www.sciencedirect.com/science/article/pii/S0167278919305974.

[12] *StandardScaler — scikit-learn 1.5.0 documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (visited on 06/30/2024).

[13] Kumiko Tanaka et al. "MIKE: 1998 International Conference on Multi Agent Systems, ICMAS 1998". In: *Proceedings - International Conference on Multi Agent Systems, ICMAS 1998*. Proceedings - International Conference on Multi Agent Systems,

ICMAS 1998 (1998). Publisher: Institute of Electrical and Electronics Engineers Inc., pp. 285–292. ISSN: 081868500X. DOI: 10.1109/ICMAS.1998.699067. URL: http://www.scopus.com/inward/record.url?scp=84867450692&partnerID=8YFLogxK (visited on 05/07/2024).

[14] Yasufumi Taniguchi et al. "Generating Live Soccer-Match Commentary from Play Data". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 7096–7103. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v33i01.33017096. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4691 (visited on 05/07/2024).

[15] Vladimir N. Vapnik. "The Support Vector method". In: *Artificial Neural Networks — ICANN'97*. Ed. by Wulfram Gerstner et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 261–271. ISBN: 978-3-540-69620-9.