Utilizing Artificial Neural Networks to Detect Compound Events in Spatio-Temporal Soccer Data

Keven Richly
Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3
Potsdam, Germany
keven.richly@hpi.uni-potsdam.de

Florian Moritz
Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3
Potsdam, Germany
florian.moritz@student.hpi.uni-potsdam.de

Christian Schwarz
Signavio GmbH
Kurfürstenstraße 111
Berlin, Germany
christian.schwarz@signavio.com

ABSTRACT

In the world of professional soccer, performance analytics about the skill level of a player and the overall tactics of a match are supportive for the success of a team. These analytics are based on positional data on the one hand and events about the game (e.g. pass, shot on target) on the other hand. The positional data of the ball and players is tracked automatically by cameras or via sensors. However, the events are still captured manually by human, which is time-consuming and error-prone. In this paper, we introduce a novel approach to detect events in soccer matches by utilizing artificial neuronal networks. As input for the neuronal network, we used several time-dependent features, which were calculated on basis of the positional data. The evaluation of the results showed that it is possible to recognize soccer events in spatio-temporal data with a high accuracy. Apart of that, we discovered that the size of the used model and the data granularity have a strong influence on the quality of the predicted results.

KEYWORDS

event detection, neural networks, soccer analytics, spatio-temporal data

ACM Reference format:
Keven Richly, Florian Moritz, and Christian Schwarz. 2017. Utilizing Artificial Neural Networks to Detect Compound Events in Spatio-Temporal Soccer Data. In *Proceedings of SIGKDD'17 Workshop on Mining and Learning from Time Series (MiLeTS), Halifax, Nova Scotia Canada, August 2017 (MiLeTS'17),* 7 pages.
https://doi.org/tbd

1 INTRODUCTION

In recent years the use of spatio-temporal data strongly increased in various areas. Especially in the highly competitive sport sector new insights gained by positional information of players – tracked by different systems and methods during a game – can have a major impact on the training and tactic of a team [6]. For professional soccer clubs performance analysis is an integral part of the coaching process [4]. In the context of performance analysis in soccer, many analyses are based on manually tracked and chronological ordered lists of game events on the one hand or the positional information of the players on the other hand [13]. For that reason, the significance and accuracy of analysis strongly correlates with the quality of the provided data. Detecting events manually is a time-intensive and error-prone task. Based on the data of matches of the German Bundesliga, we discovered that the events are not time-synchronized with the positional information and sometimes associated with the wrong player.

Therefore, in this paper we present the implementation and evaluation of a system that leverages a trained artificial neural network to automatically detect events in the positional data of soccer matches. Based on the data of the ball we computed a set of different significant features to characterize basic events in soccer (e.g. a pass). In order to train and test the neural network, we also created a gold standard on the basis of the positional data and video recordings of the matches. Additionally, we used a grid-search approach to optimize the configurations of the applied supervised learning algorithm. To evaluate the accuracy of our results we used the metrics precision, recall, and F_1-score.

The paper is organized in the following structure. In Section 2 we examine related work. Afterwards, we explain the properties of the provided data and introduce the created gold standard. In following section, we describe how the features are computed based on the positional data and Section 5 shows how we used these features to train an artificial neural network. We also provide an evaluation about the quality of our results (see Section 6). Before we conclude the paper in Section 8, we present an overview about future work.

2 RELATED WORK

Event detection from time-series data is an important task in many areas. There are various publications, which demonstrate that artificial neural networks are a promising approach to achieve this task. For example in the biomedical domain, neural networks are used to detect epileptic spikes from EEG signals [8]. In industrial security, data from wireless sensor networks is monitored to detect fires or other hazards [19]. Neural networks are a common machine learning model for

MiLeTS'17, August 2017, Halifax, Nova Scotia Canada

ACM ISBN tbd. . . $15.00
https://doi.org/tbd

this tasks [7, 8, 17]. In the world of sports, analytics and statistics are an important aspect in various decision processes of coaches, analysts, scouts, and managers. Wickramaratna et al. use neural networks to detect goal events from video data [20]. Lee et al. use neural networks to classify baseball hits based on video data [11]. The use of spatio-temporal data for sports and soccer analytics has received some attention from researchers. Yue et al. use statistical methods to evaluate player and team behavior of a soccer match based on two-dimensional data [21]. Kim et al. discuss several features that can be computed from two-dimensional tracking data [9]. Miller et al. use two-dimensional position data to detect shooting habits of basketball players [14]. Richly et al. compare different the three machine learning approaches k nearest neighbors, support vector machines, and random forests to recognize kick events in soccer data [16]. There is also research on how positional data can be used to gain strategic insights. Lucy et al. compare team strategies in home and away games using a k-nearest neighbor approach with ball possession data [12]. As tracking data can be large in volume, Bialkowski et al. research conducting player and team analysis on a large data set for one complete soccer season [2]. Kim et al. take soccer analytics one step further and present a system that predicts short-term future ball positions based on motion fields calculated from video [10].

3 DATA FOUNDATION

As mentioned before, there are various providers of spatio-temporal data for professional soccer games. The quality, granularity, and accuracy of the data vary between different competitors and also strongly depend on the used tracking technology. The provided data sets typically consist of the positional information of the players and the ball, the manually tracked list of game events as well as some meta data about the teams and players. In this paper, we focus on data of games of the German Bundesliga. Defined by the pitch size, the range of the two-dimensional coordinates goes from -52.5 to 52.5 for x and the data range of y goes from -34 to 34 (for pitches of the size of 105 m * 68 m). Since the pitch size is not exactly defined, these numbers can differ for other stadiums. The center of the pitch has the coordinates $(0, 0)$. The position values can exceed these limits. This indicates that the ball went out of bounds. Figure 1 shows the schematic layout of a soccer pitch and the coordinates of its bounds.

The list of game events includes the timestamp, event type and involved players. All events are classified in the categories pass, shot on target, neutral contact, clearance, duel, foul, offside, caution, and substitution. Several events, such as fouls, cautions or substitutions, cannot be detected just by the positional data of the ball and players. They also depend on other information, e.g. the signals of the referee. Additionally, the events are not synchronized with the positional information. The delay can be up to several seconds. To evaluate and train the supervised machine learning algorithms, we manually created a gold standard based on the video

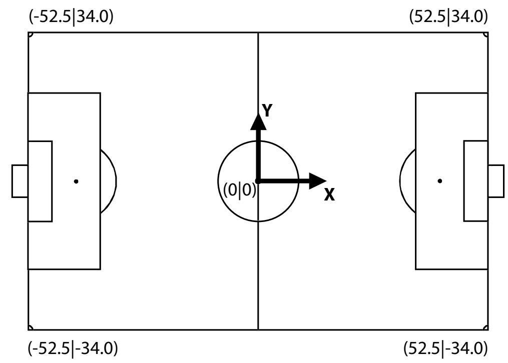

Figure 1: Soccer pitch with dimensions of bounds

recordings of the games and by taking into consideration the acceleration values of the ball. The gold standard includes the following match sections:

- **Set A_{25}**
Match: Berlin vs. Mainz
Season: 2014/15
Time: 00:00 - 03:08
Temporal resolution: 25 Hz
- **Set A_{10}**
Match: Berlin vs. Mainz
Season: 2014/15
Time: 00:00 - 03:08
Temporal resolution: 10 Hz
- **Set B_{25}**
Match: Berlin vs. Mainz
Season 2014/15
Time: 25:00 - 31:42
Temporal resolution: 25 Hz
- **Set C_{10}**
Match: Berlin vs. Braunschweig
Season: 2013/14
Time: 70:00 - 73:20
Temporal resolution: 10 Hz

The presented data sets have different temporal resolution. The original data have a resolution of 25 Hz. To filter noise and smooth the data, we applied a simple smoothing function on the provided data sets A and C. We suspect that this step could simplify the classification, especially for small data sets.

From the selected sections, we excluded the times, when the ball was out of bounds or the game was paused. Afterwards we compared the gold standard with the provided event list. We were able to find 121 out of 194 (62.4%) matching events, within a time period of two seconds and with the same event type as our event. These events had an average time delay of 0.77 seconds. As a next step we examined the assigned player for these events. For the matched events, 18 out of 121 (14.9%) players were not assigned correctly.

Table 1: Tagged events for gold standard

	Set $A_{25/10}$	Set B_{25}	Set C_{10}	Total
Pass	49	36	50	135
Reception	17	17	12	46
Clearance	0	5	1	6
Shot on Target	2	3	2	7
Total Events	68	61	65	194
Played Time	3:08 min	6:42 min	3:20 min	13:10 min
Excluded Time	0:58 min	1:49 min	1:36 min	4:23 min
Total Time	2:10 min	4:53 min	1:44 min	8:47 min

4 FEATURE COMPUTATION

Multiple features of the tracked objects characterize specific events in soccer matches. These objects move on the soccer pitch and influence each other mutually. Events occur when one or multiple features show a specific characteristic. In this section, we present the definition of the implemented features. All features are computed based on the positional data described in the previous section. The positional data is received per tracked object in a 2-by-n matrix where n is the number of collected data points in a specific time period. Each column vector represents the position of the object o at time t.

$$Pos_{o,n} = \begin{pmatrix} x_{o,t_1} & x_{o,t_2} & \cdots & x_{o,t_n} \\ y_{o,t_1} & y_{o,t_2} & \cdots & y_{o,t_n} \end{pmatrix} \tag{1}$$

We can derive the following definitions from the received positional data. The position of object o at time t is defined as $p(o,t)$. Whereas the horizontal position of object o at time t is $p_x(o,t)$ and the vertical position of object o at time t is $p_y(o,t)$. Based on the spatio-temporal data, we calculated the time-dependent movement features velocity, acceleration, and change of direction. In this context, we concentrated primarily on features of the ball, because it represents the main interaction point in the game.

To determine the velocity of two consecutive positions $p(o,t_1)$ and $p(o,t_2)$ with $t_2 = t_1 + 1$, we initially compute the Euclidian distance of these points. Based on the distance, we can compute the average velocity or rate of change of position over time as defined in Equation 3.

$$\begin{aligned} &dist(o,t_1) = \\ &\sqrt{\left(p_x\left(o,t_2\right) - p_x\left(o,t_1\right)\right)^2 + \left(p_y\left(o,t_2\right) - p_y\left(o,t_1\right)\right)^2} \\ &\qquad\qquad\qquad\qquad\qquad\qquad with\ t_2 = t_1 + 1 \end{aligned} \tag{2}$$

$$v(o,t) = \frac{\Delta dist(o,t)}{\Delta t} \tag{3}$$

Accordingly, Equation 4 determines the acceleration as the rate of change of the velocity over time.

$$a(o,t) = \frac{\Delta v(o,t)}{\Delta t} \tag{4}$$

While objects move on the soccer pitch they will eventually change their direction d.

$$d(o,t_1) = p(o,t_2) - p(o,t_1)\ with\ t_2 = t_1 + 1 \tag{5}$$

A linear movement results in no significant change of the direction feature, whereas rapid movement tends to have a notable change of direction. We computed the change of direction as visualized in Figure 2.

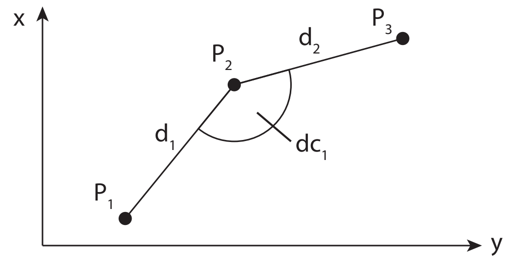

Figure 2: Direction change of object

Given the three position data points $P_0 = p(o,t_0)$, $P_1 = p(o,t_1)$ and $P_2 = p(o,t_2)$, the first direction vectors are defined as $d_0 = d(o,t_0)$ and $d_1 = d(o,t_1)$. The angle created by d_0 and d_1 is the change of direction dc_1. Possible values for direction changes are in the range from 0 to 180. To determine the direction change value, the *arccos* function is applied to the quotient of the scalar product of d_0 and d_1 and the product of length of d_0 and d_1. The direction change dc of object o at time t_{n+1} is defined in the following way:

$$dc(o,t_{n+1}) = \arccos\left(\frac{d(o,t_n) \cdot d(o,t_{n+1})}{|d(o,t_n)| \cdot |d(o,t_{n+1})|}\right) \tag{6}$$

5 EVENT DETECTION

In following section, we present our approach to recognize events based on the features already introduced in the previous section. The most central object of a soccer match is the ball. The ball is the object that shows the most and highly rapid movements on the pitch. Therefore, we computed all features based on the spatio-temporal data of the ball and created a vector for every time t containing all corresponding feature values.

Velocity and acceleration describe the current momentum. Acceleration peaks are a indicator for interactions with the ball. The direction change feature covers ball interactions with high intensity (e.g. passes) as well as ball interactions with little intensity (e.g. ball touches during dribbling). Each vector describes an instant of the soccer match and consecutive vectors can represent a certain event. Depending on the type of the event, features become more or less important and have characteristic values. To determine the specific events, we trained and used an artificial neural network.

Neural networks are biologically inspired models, that can model complex non-linear functions [15]. They consist of several connected layers of artificial neurons as shown in Figure 3. A neuron is a single computational function that maps several x_i of an input vector $\vec{x}$ (and a bias term b) to a single output value a called activation. It computes a weighted linear combination z of the inputs by using different weights w_i for each input and transforms it using a non-linear

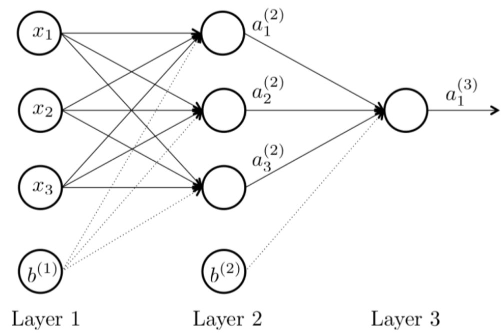

Figure 3: Neural Network consisting of 3 layers with a single output neuron.

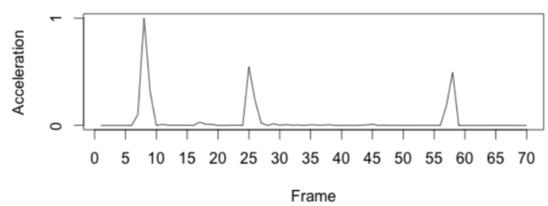

Figure 4: Acceleration of the ball (squared and normalized, 10 Hz data).

activation function h as shown in Equation 7. A commonly used activation is the *sigmoid* function which is defined in Equation 8.

$$a = h(z) = h\left(\sum_i w_i x_i + b\right) \tag{7}$$

$$\sigma(z) = \frac{1}{1 + exp(-z)} \tag{8}$$

As shown in Figure 3, the neurons are arranged in layers, where the outputs or activations of a layer serve as inputs for the following layers. The term hidden layer denotes all layers between the input and output layer. The network maps the inputs $\vec{x}$ to outputs y based on the weights W and bias terms B of its neurons (cf. Equation 9). To compute the output for a given input, the activation values of each layer are computed beginning with the input layer. The next layer uses the activation of the previous layer (e.g. the input layer) as input. This process is called forward propagation.

$$y\left(\vec{x}, W, B\right) = \vec{y} \tag{9}$$

The classification of various event types requires different output functions. For the multiclass classification of k mutual exclusive classes, we used the *softmax* function [5]. The *softmax* function for k classes is defined in Equation 10. In contrast to other activation function, the output of the *softmax* function is the posterior probability of each class [3].

$$y_k\left(\vec{x}, W, B\right) = \frac{exp(y_k)}{\sum_j exp(y_j)} \tag{10}$$

To train the artificial neural network in a supervised manner, we used a gradient-descent algorithm [1]. As first step, we initialized the network with randomly setting weight. Afterwards, the training input data with the associated outputs is used to train the neural network. Based on the corresponding error function the gradient-descent algorithm of that function by iteratively updating the weights for W and B in the direction of the negative gradient of W and B. However, this problem is not convex and the found minimum could be only a local minimum [3].

5.1 System Architecture

The core of the event detection system is an artificial neural network. To train the system, the computed feature data is transferred from the database and preprocessed for each match-period. As a first step, the data of each feature is normalized. Based on the data of the computed features, the windows are determined accordingly to the labels of the gold standard (see Section 3). By analyzing the acceleration peaks (see Figure 4) and the corresponding video sequences, we determine the window size as 7 frames for the 10 Hz data sets (A_{10}, C_{10}) and 18 frames for the 25 Hz data sets (A_{25}, B_{25}). In the next step, the three-dimensional feature windows are flattened to form a one-dimensional training instance, which is used to train the network.

The network consists of three layers. The size of the input layer depends on the window size of the used data set. For the data sets $A_1 0$ and $C_1 0$ the input layer has 21 neurons to account for the size of the flattened training instances. In the case of 25 Hz data the input layer has 54 neurons. The number of neurons in the hidden layer can be set to an arbitrary number. However, the number of hidden neurons can effect the detection performance significantly [15]. For that reason, we attempt to optimize this aspect of the network design (see Section 6). As mentioned in the pervious section, the input layer and hidden layer use the *sigmoid* activation function and for the output layer we used the *softmax* function. After the training phase the system can be used to determine the posterior probabilities for each class by analyzing the computed feature data windows.

The system is implemented in Python using the libraries for scientific computing numpy[1], scipy[2] and pandas[3]. The neural network implementation is based on the machine learning frameworks scikit-neuralnetwork[4] and scikit-learn[5].

5.2 Model Parameters

There are several parameters that influence the detection accuracy of the underlying neural network model. Therefore, we tried to increase the performance by optimizing the configuration of the parameters for the soccer data.

[1] http://www.numpy.org/
[2] http://www.scipy.org/
[3] http://pandas.pydata.org/
[4] https://github.com/aigamedev/scikit- neuralnetwork
[5] http://scikit- learn.org/

In general, there are several parameters in a neural network, which have an effect on the learning outcome. One of these is the architecture of the network. The number of neurons in the hidden layer is not specified and can be adjusted to the given data set. Neural Networks with a higher number of neurons have the ability the represent the data characteristics more precise, but they also have the risk of over fitting the training data [15]. Therefore we tried to find a number of hidden neurons that produces the highest general accuracy.

Another factor that has to be taken into account is the learning rate. It controls the rate at which the weights are updated on the basis of new information in the learning process. Low values result in a network that adopts very slowly. However, if these values are too high, the learning process may not converge [15]. To avoid over fitting we implemented a technique called dropout. Hereby, a random number of activations is set to zero for each training instance. This mechanism helps to prevent co-adaption of neurons on the training data [18]. A too high dropout rate complicates the effective learning of the network.

To find an optimal model we used a grid-search approach to test multiple parameter configurations. Here, we list the parameters and the specified values we used:

- **Number of Hidden Units**
  We used different values for 10 Hz and 25 Hz to account for their different sizes of input to the network.
  Values (25 Hz): 1 to 50
  Values (10 Hz): 1 to 20
- **Learning Rate**
  Values: 0.1, 0.05, 0.01, 0.005, 0.001
- **Dropout**
  Values: 0, 0.01, 0.05, 0.1, 0.2

These parameters are augmented by the fixed parameter for the window size as described in the previous section. The grid-search implementation uses parallel processing to test different configurations in parallel and speed up the process. The results of the grid-search for the presented parameters are evaluated in following section.

6 EVALUATION

In this section, we present the evaluation results of our approach. Based on the presented data (see Section 3), we optimized the configuration of your neural network and compared the accuracy of the different settings. For the evaluation, we focused on pass events, which occur most frequently in the gold standard. A pass event consists of two consecutive actions – *kick* and *reception*.

6.1 Preliminaries

To evaluate the quality of our event detection, we used the gold standard to generate test and training instances. These instances are labeled windows over the feature data at specific time points. The system is trained on the training instances and then presented with the unknown test instances, which it has to label. Based on the assigned label by the system

Table 2: Optimal parameter configurations for the different temporal resolutions

Parameter	25 Hz	10 Hz
Number of Hidden Units	50	20
Learning Rate	0.01	0.01
Dropout	0.05	0.01

and the true event we can compute the precision and recall scores to quantify the detection quality.

$$precision = \frac{true\_positives}{true\_positives + false\_positives} \quad (11)$$

$$recall = \frac{true\_positives}{true\_positives + false\_negatives} \quad (12)$$

We also computed the F_1-score, which is the harmonic mean of precision and recall, and use it as our main evaluation metric to compare different network settings.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (13)$$

6.2 Model Optimization

Since we have data sets with different temporal resolutions, we conducted the parameter search for each data set separately. For the grid-search approach we selected the labeled data of the first minutes of the game Berlin vs. Mainz (A_{25}, A_{10}). Based on the data sets, we generated the training and testing instances. We tested each configuration by using a five-fold cross validation with a 60/40 split for training and testing instances. To compare the accuracy of the different configuration the F_1-score was used. Table 2 shows the configurations that achieved the highest scores for the given data set.

6.3 Model Comparison

By using the configurations presented in the previous section, we analyzed in more detail the accuracy of the different models. To compare the performance of the 10 Hz and 25 Hz model, we tested each one using a 100-fold cross validation. Analogous to the configuration computation, we used a 60/40 split between training and testing instances. Afterwards, we calculated an overall precision, recall and F_1-score for each iteration and averaged them over all iterations. The results are shown in Table 3. The evaluation shows that the model trained and tested on the 10 Hz data performs much better than the 25 Hz model with an averaged F_1-score of 0.89, and averaged precision and recall scores of 0.89 and 0.90 respectively. The 25 Hz model only achieves averaged precision, recall and F_1-scores of 0.52, 0.52 and 0.49.

In the next step, we compared the performance of the two models per class. Therefore, we calculated the precision, recall and F_1-scores for each class separately over all iterations. The results are shown in Table 4. As expected, we observed that the 10 Hz have a higher accuracy for both classes compared to the 25 Hz model. For kick events, both models achieved

Table 3: Precision, recall and F_1-score, averaged over all classes

Data set	Precision	Recall	F_1-Score
25 Hz	0.52	0.52	0.49
10 Hz	0.89	0.90	0.89

Table 4: Precision, recall and F_1-score per class

Data set	Event	Precision	Recall	F_1-Score
25 Hz	Kick	0.73	0.91	0.81
	Reception	0.32	0.14	0.18
10 Hz	Kick	0.95	0.92	0.93
	Reception	0.82	0.89	0.85

Table 5: Comparison of overall results for different training and testing strategies for two matches.

Strategy	Precision	Recall	F_1-Score
Merged Matches	0.81	0.75	0.75
Across Matches	0.65	0.73	0.66
Single Match	0.89	0.90	0.89

a similar recall value of 0.91 or 0.92 respectively. However, the 10 Hz model had a better precision score of 0.95 than the 25 Hz model, which had a precision score of 0.73. That results in a F_1-score of 0.93 for the 10 Hz model, and one of 0.81 for the 25 Hz model. Both models detect most of the true kick events in the data. However, the lower precision of the 25 Hz model implies that this model is more likely to detect a false kick event. Next to that, the comparison for the reception event showed a more diverging picture. The 10 Hz model achieved precision, recall and F_1 scores of 0.82, 0.89 and 0.85. The 25 Hz model however performed not as well with only 0.32 for precision and 0.14 for recall, with an averaged F_1-score of 0.18.

This great difference in performance could be due to the fixed size of the gold standard and the fact that the 25 Hz model is more complex to train due to its larger structure. In this experiment we used the data of match A_{10} and A_{25}, for which the gold standard holds 85 labeled kick events but only 34 reception events. One reason for the performance differences could be the unequal distribution of kick and reception events.

To summarize the previous experiment we can state that a model trained and tested on 10 Hz data achieved a higher accuracy compared to one trained and tested on the 25 Hz data, using the given gold standard. This could be due to the fact that the 10 Hz data has been smoothed and has therefore fewer outliers.

6.4 Model Evaluation

In this section, we evaluate the effects on the performance of the event detection when the training and testing data derived from different matches (merged matches strategy) or the model was trained with data form one soccer match and tested with data from another (across matches strategy). For this evaluation, we focused on the 10 Hz model, because it produced the best results in the previous experiments. The motivation for this evaluation is that the characteristics (e.g. playing speed, team tactics) of a game could vary between different matches and teams.

First, we trained and tested the model on the merged data of two different matches (A_{10} and C_{10}) to set a baseline. We merged the data sets of the matches of Berlin vs. Mainz and Berlin vs. Braunschweig. Afterwards, we extracted training and testing instances based on a 60/40 split. The presented results for the merged matches strategy are averaged over 100 iterations with a random selection of training and testing values.

The second evaluated strategy is the across matches strategy. In this case, the training instances for the model were randomly selected from data set A_{10} and afterwards tested on instances of the data set C_{10}. The results of the two strategies are listed in Table 5 together with the single match results of the previous section.

In general, we observed that the scores for the merged matches strategy have a higher accuracy compared to the results of the across matches strategy. However, both have a lower performance in comparison to the single match strategy, where only one single match was used for training and testing. This suggests that we have to consider differences in playing style between different matches.

When we drill down and evaluate the performance per class, the results show that the scores for kick events are generally higher than those for reception events in all evaluated strategies. For the kick events both strategies produced results comparable to these of the model, which was evaluated on a single match. One exception to that was the recall of the across matches strategy, which was slightly lower with a value of 0.72 compared to 0.93 and 0.92 for the other strategies. The implication of this is that applying the across matches strategy will not be able to detect as many of the real kick-events. In comparison to the scores of the kick class, the scores of the reception class are much lower. While for the merged matches strategy the receptions have a precision of 0.75 and recall of 0.56, for the across matches strategy they have a precision of 0.39 and a recall of 0.75.

To conclude, the results showed that neural networks present a viable model to detect events in soccer data. Our experiments showed that keeping the complexity of the model low in combination with smoothed data helps to achieve better results. The best results were achieved if the model was trained and tested on data of a single match or mixed matches. For that reason, we recommend to use merged data of different matches as training instances to classify completely new matches.

7 FUTURE WORK

Our experiments have shown that neural networks are generally a suitable model to perform event detection on spatio-temporal soccer data. However, since we used two-dimensional data without height information, the features we calculated

Table 6: Comparison of results by event for different training/testing strategies for two matches

Strategy	Event	Precision	Recall	F_1-Score
Merged Matches	Kick	0.86	0.93	0.90
	Reception	0.75	0.56	0.61
Across Matches	Kick	0.92	0.72	0.81
	Reception	0.39	0.75	0.51
Single Match	Kick	0.95	0.92	0.93
	Reception	0.82	0.89	0.85

cannot capture ball movements on the z-axis. This fact leads to small inaccuracies in the computed features. For that reason, the incorporation of the z values could improve the accuracy of the features and consequently lead to an improvement of the results. As our experiments have shown, there is a difference in detection quality between the models working with the smoothed 10 Hz and 25 Hz data. We have to further analyze, if this is due to the fact that the gold standard includes only manageable number of events or if the data smoothing supports the learning capabilities. Accordingly, the effects of different smoothing function on the accuracy of the results could be evaluated.

8 CONCLUSION

In this paper we presented a system that is able to detect events from spatio-temporal soccer data. Using two-dimensional positional data, we computed velocity, acceleration and change of angle features to capture time-dependent movement information from the data. On these features, we then trained a neural network to detect kick and reception events and optimize its parameters through a grid-search approach. We evaluated and compared the event detection performance on raw 25 Hz data and smoothed 10 Hz data. Our experiments showed, that the neural network trained and tested on 10 Hz data achieved an F_1-score of 0.89 whereas a network for 25 Hz data achieved only a score of 0.49. Both models achieved high scores for kick events, however the 10 Hz model performed substantially better on reception events with a F_1-score of 0.85, compared to 0.18 for the 25 Hz model. The evaluation of the precision, recall, and F_1-score showed that neural networks are a viable model to detect events in spatio-temporal soccer data. Further experiments showed that training and testing on different matches have a significant effect on the accuracy of the results. This indicates that different matches and teams have different game characteristics, which influence the detection performance. To minimize those effects, the training data should consist of data from different matches.

REFERENCES

[1] Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. 1996. A new learning algorithm for blind signal separation. *Advances in neural information processing systems* (1996), 757–763.

[2] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining*. IEEE, 725–730.

[3] Christopher M Bishop. 2006. Pattern recognition. *Machine Learning* 128 (2006), 1–58.

[4] Christopher Carling, A Mark Williams, and Thomas Reilly. 2005. *Handbook of soccer match analysis: A systematic approach to improving performance.* Psychology Press.

[5] Alexandre de Brébisson and Pascal Vincent. 2015. An Exploration of Softmax Alternatives Belonging to the Spherical Loss Family. *arXiv preprint arXiv:1511.05042* (2015).

[6] Peter Dizikes. 2013. Sports analytics: a real game-changer. *Massachusetts Institute of Technology, MIT News Mar* 4 (2013).

[7] Wenchao Jiang and Zhaozheng Yin. 2015. Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks. In *Proceedings of the 23rd ACM international conference on Multimedia.* ACM, 1307–1310.

[8] Payal Khanwani, Susmita Sridhar, and Mrs K Vijaylakshmi. 2010. Automated Event Detection of Epileptic Spikes using Neural Networks. *International Journal of Computer Applications* 2, 4 (2010).

[9] Ho-Chul Kim, Oje Kwon, and Ki-Joune Li. 2011. Spatial and spatiotemporal analysis of soccer. In *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems.* ACM, 385–388.

[10] Kihwan Kim, Matthias Grundmann, Ariel Shamir, Iain Matthews, Jessica Hodgins, and Irfan Essa. 2010. Motion fields to predict play evolution in dynamic sport scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 840–847.

[11] Wen-Nung Lie, Ting-Chih Lin, and Sheng-Hsiung Hsia. 2004. Motion-based event detection and semantic classification for baseball sport videos. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, Vol. 3. IEEE, 1567–1570.

[12] Patrick Lucey, Dean Oliver, Peter Carr, Joe Roth, and Iain Matthews. 2013. Assessing team strategy using spatiotemporal data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 1366–1374.

[13] Rob Mackenzie and Chris Cushion. 2013. Performance analysis in football: A critical review and implications for future research. *Journal of sports sciences* 31, 6 (2013), 639–676.

[14] Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. 2014. Factorized Point Process Intensities: A Spatial Analysis of Professional Basketball.. In *ICML*. 235–243.

[15] Thomas M Mitchell. 1997. Machine learning. *New York* (1997).

[16] Keven Richly, Max Bothe, Tobias Rohloff, and Christian Schwarz. 2016. Recognizing Compound Events in Spatio-Temporal Football Data. In *International Conference on Internet of Things and Big Data (IoTBD).*

[17] Sami Saalasti. 2003. *Neural networks for heart rate time series analysis.* University of Jyväskylä.

[18] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[19] Chinh T Vu, Raheem A Beyah, and Yingshu Li. 2007. Composite event detection in wireless sensor networks. In *2007 IEEE International Performance, Computing, and Communications Conference.* IEEE, 264–271.

[20] Kasun Wickramaratna, Min Chen, Shu-Ching Chen, and Mei-Ling Shyu. 2005. Neural network based framework for goal event detection in soccer videos. In *Seventh IEEE International Symposium on Multimedia (ISM'05).* IEEE, 8–pp.

[21] Zengyuan Yue, Holger Broich, Florian Seifriz, and Joachim Mester. 2008. Mathematical analysis of a soccer game. Part I: Individual and collective behaviors. *Studies in applied mathematics* 121, 3 (2008), 223–243.