KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Een karakterisatie van de speelstijl van voetbalteams op basis van topic modeling

Daan Wildiers

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdoptie Artificiële intelligentie

Promotor:
Prof. dr. J. Davis

Assessoren:
Dr. Ir. M. Van der Hallen
Ir. P. Robberechts

Begeleiders:
Ir. P. Robberechts
Dr. M. Balliauw

Academiejaar 2020 – 2021

Voorwoord

Bij deze zou ik iedereen willen bedanken die mij gesteund heeft bij het tot stand brengen van deze masterproef. In het bijzonder wil ik mijn begeleider Pieter Robberechts bedanken die me wekelijks van nuttige feedback voorzag. Op de momenten dat het even stroef liep, kon hij me steeds terug op het goede pad sturen met handige tips door zijn grondig inzicht in de voetbal- en data-analyse. Ook naast de wekelijkse meetings was hij steeds bereikbaar om snel vragen te beantwoorden. Daarnaast wil ik ook de KBVB bedanken die me toestond om hun platform te gebruiken. In het bijzonder wil ik Matteo Balliauw en Yannick Euvrard bedanken die me tijdens en naast presentaties van nuttige feedback voorzagen en Sven Hermans en Payam Ebrahimi die me hielpen met de toegang tot het platform. Mijn promotor Jesse Davis zou ik willen bedanken om mij de kans te geven om over dit interessante onderwerp mijn masterproef te mogen maken en voor de feedback tijdens de presentaties. Ten slotte zou ik mijn familie en vrienden willen bedanken die me gesteund hebben tijdens het maken van deze thesis.

Daan Wildiers

Inhoudsopgave

Samenvatting

Deze thesis loopt in samenwerking met de Koninklijke Belgische Voetbalbond (KBVB). Om de prestaties van hun nationale teams te verbeteren, doet men tegenwoordig beroep op uitgebreide data-analyse. Het begrijpen van de speelstijl van (potentiële) tegenstanders is daarin een belangrijke component. Ook bij het zoeken van tegenstanders voor oefenmatchen is het bijzonder nuttig om teams met een bepaalde speelstijl makkelijk te kunnen identificeren. Het doel van deze masterproef is om op automatische wijze een begrijpbare representatie van de speelstijl van een team af te leiden uit *ball event data*. Met deze representatie moet het mogelijk zijn om de speelstijlen van teams onderling te vergelijken. De twee voornaamste problemen in voorgaand werk zijn enerzijds de begrijpbaarheid van de representatie en anderzijds het behandelen van de locatie- en tijdscomponent van fases in een wedstrijd. Er wordt in deze masterproef daarom gefocust op deze problemen en meer specifiek wordt de locatie- en tijdscomponent op een *data-driven* manier behandeld.

Verder bouwend op recent werk wordt er een begrijpbare representatie bekomen aan de hand van het *Topic Modeling* algoritme Latent Dirichlet Allocation (LDA). Analoog aan het identificeren van onderwerpen in documenten, worden er typische spelpatronen gezocht waarmee de speelstijl van een team kan beschreven worden. Dit model wordt getraind aan de hand van beschrijvingen van wedstrijden. De keuze van de kenmerken (d.i. *features*) waarmee een wedstrijd beschreven kan worden, bepaalt in grote mate welke typische spelpatronen ermee geïdentificeerd kunnen worden. Informatie zoals de interacties tussen spelers en de locaties van deze interacties worden daarom gecodeerd in de *features*. De *event data* wordt opgesplitst in bewegingskettingen waaruit op een *data-driven* manier een aantal vertegenwoordigers worden gekozen waarmee de andere bewegingskettingen kunnen vergeleken worden. Samen met de bewegingspatronen in de *final third* en een lange ballen statistiek, zijn dit de *features* waarmee een wedstrijd beschreven wordt. Hiermee worden typische spelpatronen geïdentificeerd waarmee een vector kan worden opgesteld die inzicht geeft in de speelstijl van een team.

Een aantal *use cases* en andere evaluatie tools tonen aan dat de vectoren makkelijk te begrijpen zijn en inzichten geven die overeen komen met waarnemingen van sportjournalisten. Met de vectoren is het mogelijk om teams met een gelijkaardige speelstijl te identificeren. De vectoren zijn ideaal om snel een beeld te geven van de speelstijl van een team. Voor een diepere analyse kan er ingezoomd worden om het voorkomen van een *feature* in een bepaald typisch spelpatroon te analyseren.

Lijst van figuren en tabellen

Lijst van figuren

Lijst van tabellen

Lijst van afkortingen en symbolen

Afkortingen

KBVB	Koninklijke Belgische Voetbalbond
PCA	Principal Component Analysis
MRR	Mean Reciprocal Rank
LDA	Latent Dirichlet Allocation
NMF	Non-Negative Matrix Factorization
DTW	Dynamic Time Warping
BoW	Bag of Words
KNN	K-Nearest Neigbors
PPDA	Passes Per Defensive Action

Symbolen

U_i	Uniekheid van de speelstijl van team i
C_i^k	Consistentie van de speelstijl van team i in wedstrijd k

Hoofdstuk 1

Inleiding

Deze thesis verliep in samenwerking met de Koninklijke Belgische Voetbalbond (KBVB). Om de prestaties van hun nationale teams te verbeteren, rekenen ze tegenwoordig op uitgebreide data-analyse. Het begrijpen van de speelstijl van (potentiële) tegenstanders is daarin een belangrijke component. In deze thesis worden de bestaande technieken hiervoor onderzocht en wordt er een model opgesteld en geïmplementeerd dat gebruikt kan worden in een wedstrijdvoorbereiding.

Dit inleidende hoofdstuk definiëert wat we begrijpen onder speelstijl en schetst kort de tekortkomingen van de bestaande methodes. Daarna wordt uitgelegd hoe dit probleem werd aangepakt.

1.1 Situering

De speelstijl van een voetbalteam wordt bepaald door een samenspel van verschillende factoren. De posities van spelers, de manier waarop ze bewegen en hun interacties zijn een aantal belangrijke elementen. Elk team heeft dan ook een unieke combinatie van deze factoren, gestuurd door de coach en de spelers die bepaalde tactieken hanteren. Deze combinatie kan niet achterhaald worden aan de hand van simpele statistieken zoals het aantal schoten naar het doel en het percentage balbezit, maar vergt een grondige analyse door een ervaren voetbalanalist.

Het afleiden van de speelstijl van voetbal teams is dan ook een bijzonder complex maar zeer nuttig proces. Ploegen die de speelstijl van hun tegenstander beter begrijpen, kunnen hun tactiek hieraan aanpassen en dus hun wedstrijd beter voorbereiden. Ook bij het plannen van oefenmatchen zoekt men vaak tegenstanders met een bepaalde speelstijl. Videoanalisten spenderen dan ook veel tijd aan het analyseren van wedstrijden van hun tegenstander. Om een constante te vinden in welke tactieken een tegenstander hanteert, moeten veel verschillende wedstrijden geanalyseerd worden en dan nog liefst door dezelfde personen om het volledige beeld niet uit het oog te verliezen. Dit proces is tijdrovend en daarenboven is het bijzonder complex om meermaals voorkomende tactieken af te leiden uit verschillende wedstrijden. Een automatisatie van dit proces zou een enorme meerwaarde kunnen betekenen voor analisten en coaches.

De laatste jaren zijn twee soorten wedstrijd data meer en meer beschikbaar waaruit men speelstijl automatisch kan afleiden: *ball event data* en *optical tracking data*. *Ball event data* wordt verzameld door menselijke annotators die videobeelden van wedstrijden bekijken en alle acties die spelers uitvoeren op het veld – zoals schoten, passes, dribbels, tackles en stilstaande fases – nauwkeurig beschrijven aan de hand van een aantal attributen. Bijvoorbeeld bij een schot worden de locatie, de speler die op doel schiet, of het een kopbal is, het resultaat en het tijdstip geregistreerd. De posities van spelers die niet in balbezit zijn, worden niet opgenomen in deze data [67]. *Optical tracking data* registreert wel continu de positie van elke speler en de bal aan de hand van hoge-resolutiecamera's en computervisie software. Uit deze data kan veel meer informatie gehaald worden, maar is nog minder wijdverspreid omdat clubs deze data niet altijd willen delen. Doorgaans worden deze *tracking* systemen beheerd op het niveau van de competitie, waardoor clubs enkel toegang hebben tot data van het eigen team of tot de data van teams in dezelfde competitie. Dit maakt *optical tracking data* minder geschikt voor het analyseren van de speelstijl van tegenstanders. In het bijzonder in de context van internationale wedstrijden [67, 41, 7].

Deze thesis focust dus op *ball event data*, waarvan de wijde verspreiding er de laatste jaren toe geleid heeft dat men hier verschillende AI technieken op toegepast heeft. Hier zijn een aantal bruikbare hulpmiddelen uit voortgevloeid, zeker voor het beoordelen van de prestaties van spelers en teams [19, 71, 62, 67]. Ook op het vlak van speelstijl werden heel wat modellen ontwikkeld, voor zowel die van een speler als een team. Het gedrag van een team analyseren is op sommige vlakken makkelijker dan het gedrag van een speler omdat er meer *event data* per team beschikbaar is dan per speler. Langs de andere kant is het moeilijker om die van een team af te leiden uit data omdat er met veel meer factoren rekening gehouden moet worden zoals de opeenvolging van acties en de interacties tussen spelers [67].

De bestaande methodes die de speelstijl van teams analyseren op basis van *ball event data* zijn vrij uiteenlopend. Sommige vatten een speelstijl van een ploeg samen aan de hand van een aantal *features* (meestal simpele statistieken zoals aantal schoten en voorkomen van spelers in bepaalde zones) die men dan clustert om gelijkaardige ploegen te identificeren of omzet naar een vector representatie [25, 21, 29]. Andere extraheren dan weer patronen die veel voorkomende fases of passmotieven beschrijven met een *pattern mining* algoritme of *inductive logic programming* [22, 9, 65]. Daarnaast zijn er ook netwerk-gebaseerde benaderingen die de interacties tussen spelers (voornamelijk passes) beschrijven aan de hand van grafentheorie of Markov modellen [38, 16]. Een recentere aanpak gebaseerd op *mixture models* modelleert dan weer locaties en richtingen van bepaalde acties die bij een ploeg het meest voorkomen [23].

Er zijn twee veel voorkomende problemen met de bestaande benaderingen om de speelstijl van een team af te leiden uit wedstrijd data. Het eerste probleem is de representatie van speelstijl die men gebruikt. In een aantal benaderingen wordt een vector opgesteld voor een team die moeilijk te interpreteren is en waarvan de componenten op zichzelf ook niet eenvoudig te begrijpen zijn [30, 5, 25, 38]. Deze soort vectoren kunnen onderling wel vergeleken worden met elkaar en men kan weten hoe verschillend deze vectoren zijn maar het is minder duidelijk op welk vlak ze

precies verschillen. Deze representatie is meestal wel een goede weergave van unieke speelstijlen maar is dus vaak minder goed begrijpbaar. Dit verhindert dan ook dat coaches en analisten deze representaties in praktijk kunnen gebruiken bij hun wedstrijdvoorbereiding.

Een tweede probleem is meer technisch van aard en gaat over hoe men omgaat met de *spatio-temporele* aspecten van wedstrijd data. In vele benaderingen deelt men, om de zoekruimte van algoritmes te verkleinen, het veld op in zones of verwaarloost men de richting, actie of opeenvolging van acties [23, 21, 66].

1.2 Probleemstelling

Het voornaamste doel van deze thesis is te onderzoeken hoe men op automatische wijze een begrijpbare representatie van speelstijl kan afleiden uit *ball event data*. Meer concreet wordt er een model opgesteld dat bestaande technieken zal combineren om zo voor elk team tot een speelstijlvector met begrijpbare componenten te komen. Deze vector moet bruikbaar en begrijpbaar zijn voor coaches en analisten om hun speelstijlanalyse van tegenstanders te ondersteunen. Daarenboven moeten de speelstijlvectoren van teams onderling vergelijkbaar zijn en het moet duidelijk zijn op welke vlakken hun speelstijl verschilt.

In dit werk gaan we er vanuit dat speelstijl (of de beslissingen die spelers op team-niveau maken) geleid wordt door de locatie op het veld en de opeenvolging van voorgaande acties. Welke actie de speler uitvoerde is minder van belang. Een tweede belangrijk probleem waarop gefocust wordt, zijn de kenmerken waarmee een wedstrijd beschreven wordt. In plaats van deze te beschrijven aan de hand van een aantal gebeurtenissen, wordt een wedstrijd beschreven aan de hand van een aantal opeenvolgingen van acties. De locatie- en tijdscomponent van deze opeenvolgingen van acties wordt op een *data-driven* manier behandeld.

1.3 Benadering

In wezen bestaan er een onbeperkt aantal verschillende speelstijlen die een team kan gebruiken. Deze speelstijlen kunnen echter worden geanalyseerd in termen van een beperkt aantal afzonderlijke en vereenvoudigde stijlcomponenten die elk in verschillende mate kunnen worden geïmplementeerd en in verschillende vormen kunnen worden samengesteld uit *features* waarmee een wedstrijd beschreven wordt. In die zin kan het analyseren van speelstijl vergeleken worden met de analyse van teksten, waarin de verschillende woorden (cfr. *features*) behoren tot een of meerdere onderwerpen (cfr. stijlcomponenten). Op basis van dit inzicht pasten Perdomo Meza en Girela [51, 50] het Latent Dirichlet Allocation (LDA) [36] model toe op de analyse van speelstijl. Dit is een van de meest gebruikte *Topic Modeling* technieken die begrijpbare en logische resultaten oplevert [37]. In deze thesis wordt dit idee verder uitgewerkt.

In eerste instantie werden alle bestaande methodes die trachten de speelstijl van teams te achterhalen uit *ball event data* bekeken om hierop verder te bouwen. In het

model van Perdomo Meza en Girela [51, 50] worden een aantal *features* opgesteld door de *event data* te aggregeren, zoals het pass percentage, aantal schoten in een bepaalde zone, intercepties, etc. *Features* van eenzelfde team die in dezelfde wedstrijd sterk gecorreleerd zijn, worden dan door het LDA model gegroepeerd en vormen een stijlcomponent. Een stijlcomponent die de hoeveelheid countervoetbal van een team weergeeft, wordt dan bijvoorbeeld voorgesteld als 30% progressieve passes, 40% geslaagde doorsteekpasses, 10% dichte schoten en 20% één-op-één pogingen. Aan de hand van deze stijlcomponenten wordt een begrijpbare vector opgesteld voor elk team.

De LDA methode levert goede resultaten op, maar in deze thesis worden vier beperkingen geïdentificeerd en aangepakt. Een eerste probleem is dat er stijlcomponenten worden geïdentificeerd die eerder met de prestatie van een team te maken hebben dan met speelstijl. Een van deze stijlcomponenten is 'Conceding Chances' die is samengesteld uit *features* zoals het aantal reddingen, het aantal doelpogingen in het strafschopgebied, het aantal keren dat een speler voorbij gedribbeld wordt en het aantal intercepties in het strafschopgebied. Deze *features* zeggen weinig over hoe een team speelde maar beschrijven eerder in welke mate een team zijn tegenstander het creëren van gevaarlijke kansen kon beletten. Daarom wordt er onderzocht welke *features* inzicht geven in de manier waarop een team speelt en met welke *features* het LDA model de meest logische en begrijpbare stijlcomponenten kan identificeren.

Een tweede probleem is dat de gebruikte *features* vrij simpel zijn en veel informatie verwaarlozen uit de *ball event data*. We bouwen verder op deze methode maar proberen meer informatieve *features* op te stellen. Bij het opstellen van de *features* houden we in dit onderzoek rekening met de opeenvolging en locaties van acties. In het werk van Decroos et al. [22] worden gelijkaardige fases geclusterd aan de hand van hiërarchische clustering. De gelijkheid van fases wordt bepaald door Dynamic Time Warping (DTW). We bouwen verder op deze methode om *features* op te stellen waarmee een wedstrijd beschreven kan worden zodat deze *feature*-representaties van wedstrijden als invoer kunnen gebruikt worden voor het LDA model. Het doel is om een aantal clusters te bekomen die een goede weergave zijn van de meest voorkomende fases. Een belangrijk aspect hierbij is hoe men de fases opdeelt. Aan de hand van deze clusters kan men voor elk team bepalen in welke mate men bepaalde fases vaker hanteert dan andere.

Een derde probleem is dat Perdomo Meza en Girela [51, 50] het getrainde LDA model en de speelstijlvectoren van teams niet evalueerden met objectieve validatie methodes. Het is niet eenvoudig om een representatie van speelstijl op een objectieve manier te beoordelen omdat de speelstijl van een team een subjectief concept is waar geen *ground truth* van bestaat. Daarom worden er objectieve evaluatie tools onderzocht en ontworpen waarmee de optimale parameters van het LDA model en van de *features* die een wedstrijd beschrijven, kunnen worden bepaald. Deze tools kunnen in de toekomst gebruikt worden om representaties van speelstijl tussen verschillende werken op een objectieve manier te vergelijken.

Een vierde probleem is dat om er voor te zorgen dat een speelstijlvector makkelijk te begrijpen en te vergelijken is, deze slechts uit een beperkt aantal stijlcomponenten bestaat. Dit zorgt ervoor dat een coach of analist snel inzicht kan krijgen in de

speelstijl van een team. Voor een diepere analyse van de speelstijl van een team is deze informatie echter te beperkt. Daarom wordt een manier onderzocht waarmee er ingezoomd kan worden om zo het voorkomen van een *feature* en een stijlcomponent dieper te analyseren.

1.4 Overzicht

Hoofdstuk 2 geeft een overzicht van de verschillende vormen van wedstrijd data in voetbal en bespreekt het bestaande werk rond datagebaseerde karakterisatie van de speelstijl van teams. In hoofdstuk 3 wordt de methode voor het opstellen van speelstijlvectoren en het opstellen van *features* die een wedstrijd beschrijven toegelicht. Deze methode wordt in hoofdstuk 4 geëvalueerd en er worden toepassingen van besproken. Ten slotte worden de belangrijkste resultaten besproken in hoofdstuk 5 en worden er een aantal mogelijke pistes voor verder onderzoek aangereikt.

Hoofdstuk 2

Achtergrond

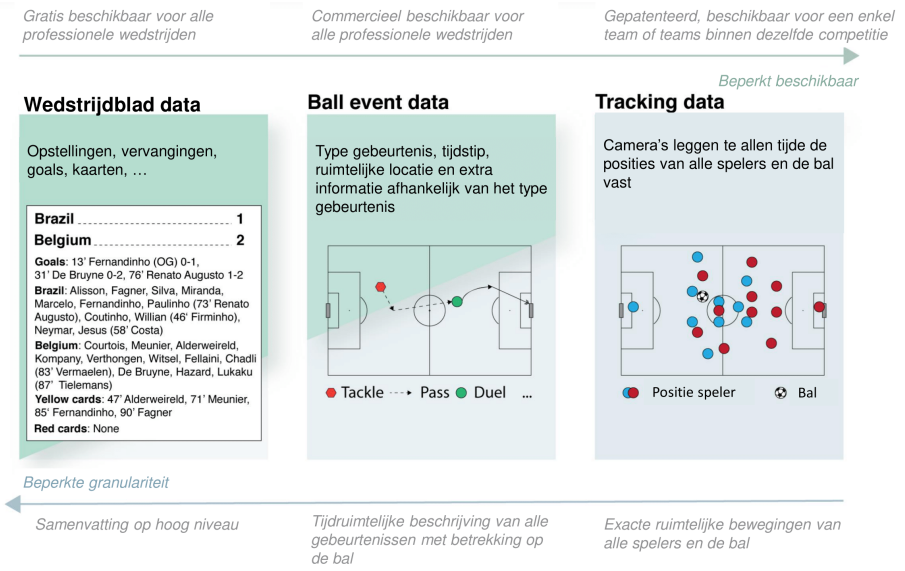
In dit hoofdstuk worden de verschillende vormen van wedstrijddata en de keuze voor *ball event data* uitgelegd. Daarnaast wordt het bestaande werk dat de speelstijl van een voetbalploeg tracht te beschrijven aan de hand van deze data besproken.

2.1 De verschillende vormen van wedstrijddata

De laatste jaren wordt meer en meer data verzameld die het verloop van voetbalwedstrijden beschrijven. Deze data kan grofweg onderverdeeld worden in wedstrijdblad (*matchsheet*) data, *ball event data* en *tracking data* (figuur 2.1). De keuze voor een van deze drie soorten is steeds een afweging tussen enerzijds de beschikbaarheid van de data en anderzijds de hoeveelheid informatie die er mee gereconstrueerd kan worden.

Wedstrijdblad data beperkt zich tot de opstellingen van beide teams, en het vermelden van vervangingen, doelpunten en kaarten. Deze data is voor alle professionele wedstrijden gratis beschikbaar op sites zoals FlashScore.com en wordt soms nog verder aangevuld met enkele geaggregeerde statistieken zoals het percentage balbezit en het aantal schoten op doel. De data bevat echter te weinig details om de speelstijl van teams te analyseren. Ze geeft bijvoorbeeld wel weer hoeveel doelpunten een team kon scoren, maar hoe deze doelpunten tot stand kwamen is niet af te lezen uit de data.

Tracking data bevindt zich helemaal aan de andere kant van het spectrum. Deze data beschrijft de positie van de spelers en de bal op vaste tijdstippen (doorgaans 25Hz), en bevat dus veruit het meeste informatie om een wedstrijd te reconstrueren. Om deze data te bekomen is wel een speciaal *optical tracking system* nodig in het stadion [7, 67], waardoor de data in veel kleinere competities nog niet beschikbaar is. In de grotere competities wordt deze data dan ook nog eens enkel binnen de eigen competitie gedeeld, zodat het analyseren van tegenstanders in bijvoorbeeld de Champions League aan de hand van deze data niet mogelijk is [67]. Een ander nadeel is dat het moeilijker is om inzichten te verwerven uit deze data dan uit *ball event data*, waarbij een menselijke annotator de continue bewegingen van de spelers op het veld opdeelt in de atomaire acties die ook door analisten gebruikt worden



Figuur 2.1: Grafische weergave van de soorten data die een match beschrijven. Ball event data vindt een goed evenwicht tussen de beschikbaarheid en granulariteit. Figuur gebaseerd op Van Haren et al. [67].

(bv. een pass, dribble of tackle). Bijgevolg vereist *tracking data* een uitgebreidere *preprocessing* stap voor deze effectief gebruikt kan worden als invoer voor een model dat deze context in rekening wil brengen. In bepaalde spelsituaties zoals bijvoorbeeld een schot op doel is de locatie van de speler die op doel schiet immers belangrijker dan die van de andere spelers, maar in onbewerkte *tracking data* krijgen de posities van deze spelers evenveel gewicht [41]. Tegenwoordig wordt deze data dan ook vaak gekoppeld aan *ball event data* om deze context makkelijker in rekening te kunnen brengen [59].

Ball event data vindt op vlak van beschikbaarheid en granulariteit een gulden middenweg. Ook voor kleinere competities wordt deze soort data verzameld en aangeboden door verschillende verkopers zoals Opta data van Stats Perform[1], Wyscout[2], en StatsBomb[3] [67]. Deze soort data beschrijft de meest belangrijke gebeurtenissen in een wedstrijd. Dit zijn gebeurtenissen die spelers met de bal verrichten zoals passes, dribbels, tackles, intercepties, voorzetten en schoten; maar ook andere belangrijke gebeurtenissen zoals wissels en kaarten worden vastgelegd. Typisch worden deze gegevens verzameld door personen die TV beelden bekijken en met behulp van speciale annotatie software alle belangrijke events beschrijven met een aantal standaardattributen [11]. Deze standaardattributen zijn meestal de locatie (x,y positie), het tijdstip, het type gebeurtenis (pass, dribbel, voorzet, ...) en de betrokken spelers. Afhankelijk van het type gebeurtenis wordt nog extra informatie geregistreerd zoals

[1] https://www.statsperform.com/opta/
[2] https://wyscout.com/
[3] https://statsbomb.com/

de eind locatie van een pas of het al dan niet slagen van een tackle. Het nadeel van deze soort data is echter dat ze weinig informatie bevat over de spelers die niet aan de bal zijn, wat het analyseren van het defensieve gedrag van een team bemoeilijkt. In 2018 introduceerde StatsBomb met *pressure events* een verbetering op dit vlak. Telkens een speler druk zet wordt dit geregistreerd samen met de speler die onder druk wordt gezet en zijn bijhorende actie (pass, dribbel, ...) en locatie [63].

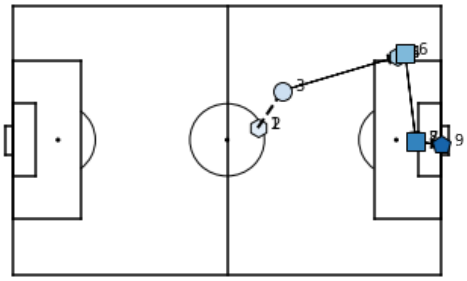
In deze masterproef maken we enkel gebruik van *ball event data* vanwege de belangrijke voordelen dat deze soort data voor bijna alle professionele competities te verkrijgen is en de vele context informatie (zoals de soort gebeurtenis) die ze bevat in vergelijking met *tracking data*. Er wordt gebruik gemaakt van Opta *event data* van Stats Perform [4]. Deze data omvat echter geen *pressure events* zoals bij StatsBomb *event data*.

2.2 Voorverwerking van de data

Ball event data dient vaak meerdere doeleinden (bv. informatie rapporteren aan kranten, omroepen of voetbalclubs) waardoor de structuur ervan meestal niet ideaal is voor data analyse. Elke soort gebeurtenis heeft bijvoorbeeld verschillende attributen. Het bevat soms ook irrelevante informatie voor data-analyses (bv. Wyscout registreert duels tussen twee spelers als twee aparte gebeurtenissen) [19]. Deze twee eigenschappen maken de *preprocessing* stappen in data-analyses met *event data* een stuk complexer. Om de analyses met deze data te vergemakkelijken werd het SPADL-formaat ontwikkeld door Decroos et al. [20]. Dit formaat zet de ongestructureerde *event data* om naar een *attribute-value* representatie dat elke gebeurtenis aan de hand van dezelfde twaalf attributen beschrijft. De *ball event data* representaties van StatsBomb, Wyscout en Opta kunnen met het publieke Python pakket *socceraction* [57] omgezet worden naar SPADL. Het formaat zorgt er dus ook voor dat data analyses met event data van verschillende providers compatibel zijn. Meer informatie betreffende SPADL is te vinden in de paper van Decroos et al. [20] of de documentatie van *socceraction* [39]. In deze masterproef wordt specifiek het Atomic-SPADL formaat [5] gebruikt. Dit is een variatie op het klassieke SPADL formaat waarbij een gebeurtenis en zijn uitkomst opgesplitst worden in twee acties. In het klassieke SPADL formaat wordt de uitkomst van bepaalde gebeurtenissen bijgehouden in een attribuut genaamd 'result'. Bijvoorbeeld bij een succesvolle pass krijgt dit attribuut dan de waarde 'success' en bij een mislukte pass 'fail'. Bij Atomic-SPADL wordt er een aparte '*ball receival*' actie voor het ontvangen van die pass aangemaakt. In plaats van het resultaat van een pass in een attribuut van die actie op te slaan wordt er een aparte actie voor het ontvangen van die pass aangemaakt. In figuur 2.2 wordt een voorbeeld van een fase getoond aan de hand van *ball event data* in Atomic-SPADL formaat [56].

[4] https://www.statsperform.com/opta/

[5] https://dtai.cs.kuleuven.be/sports/blog/introducing-atomic-spadl:-a-new-way-to-represent-event-stream-data



	time	actiontype	player_name	team
1	3m30s	receival	Romelu Lukaku	Belgium
2	3m31s	dribble	Romelu Lukaku	Belgium
3	3m33s	pass	Romelu Lukaku	Belgium
4	3m34s	receival	Nacer Chadli	Belgium
5	3m35s	dribble	Nacer Chadli	Belgium
6	3m36s	cross	Nacer Chadli	Belgium
7	3m36s	receival	Thomas Meunier	Belgium
8	3m37s	shot	Thomas Meunier	Belgium
9	3m37s	goal	Thomas Meunier	Belgium

Figuur 2.2: Visualisatie van de eerste goal in de troostfinale van België tegen Engeland op het WK 2018 in Atomic-SPADL formaat.

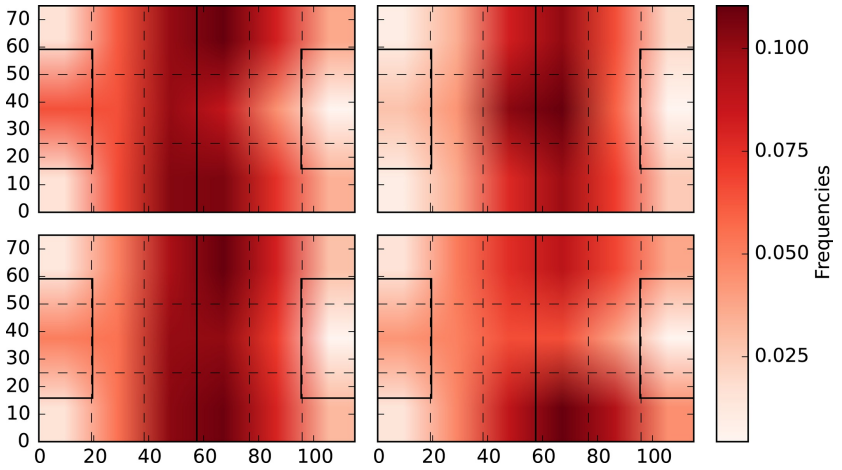
2.3 Datagebaseerde karakterisatie van speelstijl

Er bestaat geen eenduidige definitie van wat de speelstijl van een team nu precies inhoudt. De speelstijl van een team is een samenstelling van zoveel verschillende componenten dat het bijzonder complex is om deze allemaal in rekening te brengen. Bestaande werken die de speelstijl van een team aan de hand van data karakteriseren, focussen dan ook meestal op bepaalde aspecten die inzicht verwerven in de speelstijl van een team. Grofweg genomen zijn er momenteel drie manieren om de speelstijl van voetbal teams te karakteriseren aan de hand van data.

2.3.1 Karakterisatie aan de hand van *features*

Een eerste manier is het beschrijven van teams aan de hand van een aantal kenmerkende eigenschappen (d.i. *features*) en met deze beschrijvingen teams te clusteren tot een aantal stijlen. Een belangrijk voordeel van deze manier is de mogelijkheid om de speelstijlen van teams onderling te vergelijken met een afstands- of gelijkenismaat aangezien ze gekarakteriseerd worden aan de hand van dezelfde *features*. Deze *features* zijn bijvoorbeeld het aantal voorkomens van een actie in bepaalde zones [67]. Een belangrijke ontwerpkeuze is de gedetailleerdheid van de *features*. De locatie van een actie kan bijvoorbeeld voorgesteld worden als de exacte positie, als een vak in een raster over het veld of als een op domein kennis gebaseerde zone. De *feature* 'passes op de linkerflank' is bijvoorbeeld heel begrijpbaar maar zeer algemeen waardoor er informatie verloren gaat. Bovendien kan het een clustering algoritme beperken in het vinden van verrassende patronen omdat er toch al enige *bias* in de *features* zit. Langs de andere kant kan een algemenere beschrijving ervoor zorgen dat gebeurtenissen in bepaalde zones als gelijkaardig worden beschouwd. Ook een mogelijkheid is om de *features* op een *data-driven* manier samen te voegen tot nieuwe *features* die bijvoorbeeld mogelijke speelstijlen voorstellen [30, 3, 21, 23, 12].

Kerr et al. [14] definiëren deze *features* als de verdeling van de start locatie van passes over het veld. Een *heatmap* geeft op een visuele manier de frequentie van het aantal passes in elke zone weer. De *heatmaps* in figuur 2.3 bevestigen bijvoorbeeld



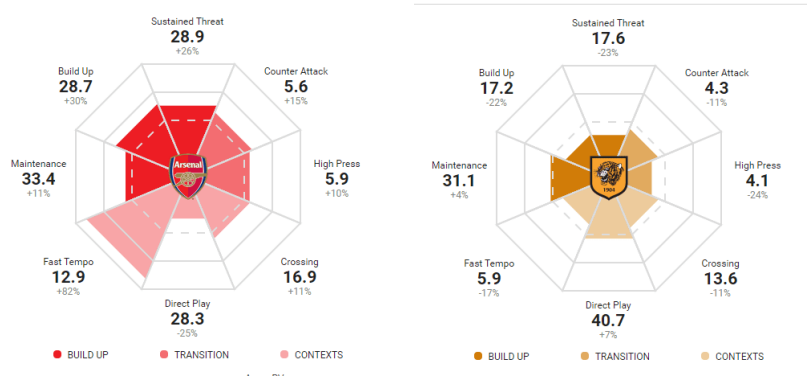
Figuur 2.3: Elke figuur is een 3x6 heatmap die de locatie van de oorsprong van passes weergeeft. Linksboven wordt de heatmap van alle passes van alle teams in seizoen 2012-13 van La Liga getoond. Rechtsboven alle passes van Barcelona, linksonder die van Real Madrid en rechtsonder die van Atletico Madrid. Figuur overgenomen uit Kerr et al. [14]

de publieke opinie dat Barcelona (rechtsboven) meer doorheen het centrum speelt en Atletico Madrid (rechtsonder) meer langs de flanken.

Fernandez-Navarro et al. [30] definiëren hun features dan weer als 19 zorgvuldig uitgekozen prestatie-indicatoren die zowel de aanvallende als defensieve strategie van een team beschrijven. Met een Principal Component Analysis (PCA) groepeerden ze de prestatie-indicatoren (bv. percentage balbezit in de *defensive third*, aantal voorwaartse passes, aantal onderscheppingen in de *final third*) in zes factoren die ieder twee stijlen voorstellen. Één factor bevat bijvoorbeeld prestatie-indicatoren die gelinkt worden aan de directheid van het spel van een team in balbezit. Een positieve waarde voor deze factor wijst op een directe speelwijze en een negatieve waarde op een meer op balbezit gerichte speelwijze.

Stats Perform [3] ontwikkelde een *framework* die de speelstijl van een team kan beschrijven aan de hand van acht verschillende fases in de opbouw van een aanval of een transitie van verdediging naar aanval. De *features* worden hier gedefiniëerd als fases van balbezit in een wedstrijd. Voor elke fase wordt met de relevante informatie gelijktijdig een waarde berekend voor elke component. De sterkte van elke component in de speelstijl representatie is een waarde tussen 0 en 100% en wordt bepaald door de definities en berekeningen van elke component. De waarde voor 'Sustained Threat' wordt bijvoorbeeld bepaald door de tijd dat een ploeg in de *final third* van het veld in balbezit is. Voor elke component werd er een model getraind met een aantal gelabelde fases om de specifieke berekening van component te bepalen. Deze labels en de definities van de componenten werden in een iteratief proces door domein experts bepaald. In figuur 2.4 wordt dit *framework* geïllustreerd met de speelstijlen van Arsenal en Hull City.

Perdomo Meza en Girela [51, 50] visualiseren de speelstijl van een team met



Figuur 2.4: STATS speelstijl representatie van Arsenal (links) en Hull City (rechts). Figuur overgenomen van Stats Perform [3].

een gelijkaardige representatie als het voorgaande werk van Stats Perform. De stijlcomponenten waarmee de speelstijl van een team beschreven wordt, worden echter automatisch geïdentificeerd gebruik makend van LDA terwijl de stijlcomponenten bij Stats Perform worden berekend aan de hand van zelf opgestelde definities en gelabelde fases. De *features* die een wedstrijd beschrijven, werden hier opgesteld door *event data* te aggregeren. Voorbeelden van *features* zijn het aantal schoten in een bepaalde zone, het aantal progressieve passes, het aantal voorzetten van buiten het strafschopgebied, etc. Deze *features* houden enkel rekening met het voorkomen van een enkele gebeurtenis in een wedstrijd terwijl die van Stats Perform opeenvolgingen van gebeurtenissen in beschouwing nemen. Aangezien deze thesis verder bouwt op dit idee, wordt dit werk in meer detail besproken in sectie 3.3.

Bekkers et al. [9] en Laszlo et al. [35] definiëren hun *features* dan weer als *flow motifs*. Een voorbeeld van een *flow motif* is bijvoorbeeld ABAC waarbij iedere letter een andere speler voorstelt en een opeenvolging van letters een pass tussen deze spelers. Voor elk team kan men dan de verschillende voorkomens van een aantal *flow motifs* berekenen om tot een uniek profiel van dat team te komen.

Bojinov en Bornn [13] definiëren hun *features* als de *control* en *disruption surface* van een team. De *control surface* van een team geeft weer in welke gebieden van het veld dat team beter of slechter dan gemiddeld in staat is om in balbezit te blijven. De *disruption surface* toont dan weer de gebieden waar dat team beter of slechter in staat is om de tegenstander het balbezit te doen verliezen en over te nemen.

Bialkowski et al. [12] ontwierpen een methode die met *tracking data* de formaties van teams berekent met een algoritme gelijkaardig aan *k-means*. Samen met de wedstrijd statistieken en *occupancy maps* (hoe vaak een team balbezit heeft in bepaalde zones) vormen de formaties de *features* waarmee een wedstrijd van een team beschreven wordt. Met linear discriminant analysis werden deze beschrijvingen gereduceerd tot vijf stijlen waarmee een compacte maar onderscheidende representatie van een team kan worden opgesteld. De gebruikte *features* laten echter wel belangrijke informatie zoals de interacties tussen spelers weg.

Met Player Vectors ontwierpen Decroos et al. [21] een methode die de speelstijl

van een speler karakteriseert aan de hand van een aantal begrijpbare componenten. Deze componenten werden met Non-Negative-Matrix Factorization (NMF) per actietype (schot, pass, interceptie, dribble) geïdentificeerd met als *features* het aantal voorkomens van de actie in elke cel van het 50x50 grid dat over het veld werd gelegd. Deze componenten stellen ieder een typische locatie van een actie voor en zijn logisch voor een gebruiker. Voor bijvoorbeeld een voetbalscout is het veel eenvoudiger om spelers te vergelijken aan de hand van deze componenten dan aan de hand van de *heatmaps* van elke actie zoals Kerr et al. [14] de locatie van een actie weergeven.

Waar Player Vectors alleen rekening hield met de locatie van de acties, integreerden Decroos et al. ook de richting van deze acties in SoccerMix [23]. Voor elke soort actie werden prototypes opgesteld met een Gaussian Mixture Model (een soft-clustering techniek) [55] voor zowel de locatie als de richting. Voor spelers leveren deze twee methodes een begrijpbare representatie van hun speelstijl op. Voor teams zijn deze methodes echter minder geschikt aangezien de opeenvolging van acties hier niet in rekening wordt gebracht. De speelstijl van een team wordt namelijk in belangrijke mate gekarakteriseerd door de interacties tussen spelers.

2.3.2 Karakterisatie door het vinden van team-specifieke patronen

Een tweede manier om de speelstijl van een team te karakteriseren is het extraheren van patronen uit de data die typisch zijn voor een team. De gevonden patronen kunnen inzicht in de speelstijl van dat team geven. Een grote uitdaging bij deze manier is om de representatie van de patronen zo op te stellen dat ze informatief, intuïtief en logisch voor de gebruiker zijn [67]. In de meeste werken wordt dan ook gefocust op de passes die een team verricht zodat de patronen simpeler en bijgevolg dus ook meer begrijpbaar kunnen worden voorgesteld [66, 22, 68, 69, 33, 65, 70]. Een nadeel van deze benadering is wel dat het moeilijker is om de speelstijlen van teams onderling te vergelijken (bv. met behulp van een gelijkenis- of afstandsmaat) omdat de patronen per team verschillen.

Van Haaren et al. [66] en het vervolg van Decroos et al. [22] hanteren een gelijkaardige vijf stappen benadering om een aantal typische en relevante spelpatronen van een team te vinden. In de eerste stap wordt een wedstrijd opgesplitst in ononderbroken fases van balbezit. In de tweede stap worden deze fases gegroepeerd zodat ruimtelijk overeenkomstige fases in dezelfde cluster zitten. Van Haaren et al. [66] stellen de fases voor met een *possession map* waarin elk element een cel in een raster over het veld voorstelt. Er wordt geteld hoeveel keer er een gebeurtenis in die fase in een cel voorkomt. De *possession maps* worden dan ruimtelijk geclustert met Expectation Maximization en MAP. Decroos et al. [22] clusteren dan weer de exacte locaties van de fases met *agglomerative clustering* en Dynamic Time Warping (DTW) als afstandsmaat tussen de fases. In deze thesis werd het opstellen van de *features* in sectie 3.4 gebaseerd op de clustering methode van Decroos et al. [22]. In elke cluster ontdekt een *pattern mining* algoritme dan een aantal typische spelpatronen. Van Haaren et al. [66] stellen spelpatronen voor als *shot*, *cross* of *run* itemsets waarbij de locatie, richting en afgelegde afstand van de actie in rekening worden gebracht. De locatie is hier beperkt tot een cel in een 5x5 raster over het veld maar doordat

de richting en afstand van de actie ook in de representatie is inbegrepen, kan een spelpatroon hiermee goed gereconstrueerd worden. Decroos et al. [22] beperkten de mogelijke locaties van een gebeurtenis in een spelpatroon tot vijf op domein kennis gebaseerde zones. Dit maakt de gevonden patronen algemener en meer begrijpbaar maar wel minder gedetailleerd.

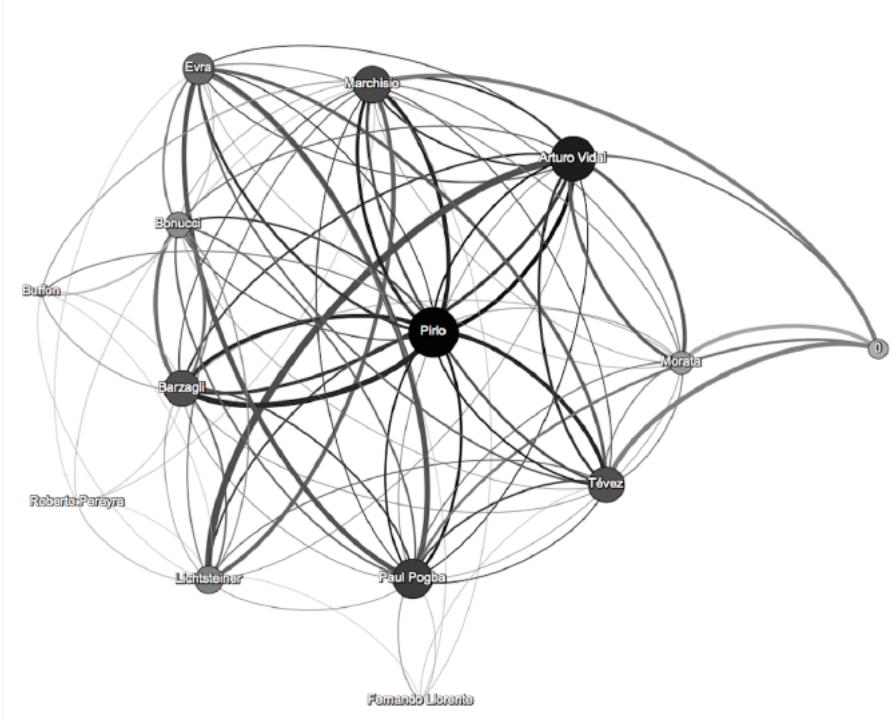
Wang et al. [68] ontwikkelden een variatie op LDA genaamd Team Tactic Topic Model (T^3M) dat de meest voorkomende pass interacties tussen spelers ontdekt. Een opmerkelijk detail is dat niet alleen de interacties tussen de spelers maar ook de locatie ervan tegelijk ontdekt wordt. Een wedstrijd wordt hier opgesplitst in segmenten van enkel passes. In elk segment worden de passes beschreven aan de hand van de eindlocatie en de verzender en ontvanger van de pass. Met de segmenten als invoer identificeert het T^3M model een aantal tactische patronen waarmee de segmenten beschreven kunnen worden.

Van Haaren et al. [65] ontwikkelden dan weer een methode die gebruik maakt van Inductive Logic Programming (ILP). Om meer algemene patronen te ontdekken, integreerde men hiërarchische informatie in de representaties van de locaties en de posities van spelers. Zo wordt de aanvallende helft van het veld bijvoorbeeld onderverdeeld in drie gebieden (bv. penalty gebied) die elk nog eens anders onderverdeeld worden in verschillende zones. Daarnaast worden de posities van de spelers ook onderverdeeld in meer specifieke posities. De positie verdediger wordt bijvoorbeeld onderverdeeld in meer specifieke posities zoals centrale verdediger, *full-* of *wing back*. De relaties tussen de locaties en posities worden als achtergrond kennis doorgegeven aan ILP. Met ILP worden dan de meest frequente patronen gevonden zoals ruimtelijke (bv. pass van de ene naar de andere zone) of speler interactie (bv. pass van de ene naar de andere speler) patronen.

Stats Perform ontwierp met Movement Chains [70] een interessante manier om het gedrag van een team te analyseren. Een *movement chain* beschrijft het patroon van vier opeenvolgende acties (telkens door een andere speler) in een ononderbroken spelfase waarbij enkel de eerste aanraking van elke speler wordt weergeven. Een speler kan meer dan eens in een *movement chain* betrokken zijn. In verdere analyses werden de *movement chains* opgedeeld afhankelijk van in welke zone ze starten en eindigen. In elke groep werden ze dan nog eens ruimtelijk gegroepeerd met een clustering algoritme met de euclidische afstand als gelijkenismaat. De *centroids* van elke cluster geven een aantal mogelijke bewegingspatronen weer waarmee men het gedrag van teams kan bestuderen. In deze thesis werd het opstellen van *features* in sectie 3.4 gebaseerd op dit werk.

Gyarmati et al. [33] extraheren de meest voorkomende opeenvolgingen van passes voor elk team. Dit doen ze door terugkerende patronen in de verschillende pass sequenties van een team te zoeken met een op DTW gebaseerd algoritme. Enkel de locaties van de passes worden in beschouwing genomen in een sequentie.

Wei et al. [69] ontwikkelden een methode die uit *tracking data* de meest voorkomende offensieve en defensieve spelpatronen (de trajecten van alle spelers en de bal) van een team kan identificeren. De methode maakt gebruik van een *role-based representation* om de opstelling van een team voor te stellen in plaats van de spelers zelf. Het voordeel hiervan is dat de opstellingen van teams makkelijker te vergelijken



Figuur 2.5: Een pass netwerk van een wedstrijd van Juventus in de Serie A in het seizoen 2013/14. Figuur overgenomen uit Cintia et al. [16].

zijn en wissels of blessures van spelers geen invloed hebben. Aan de hand van deze *role-based representation* kunnen de trajecten van elke speler en de bal beschreven worden. Na een aantal *feature reduction* strategieën worden de spelpatronen van 10 seconden gegroepeerd met een *agglomerative clustering* algoritme dat $L\infty$ als afstandsmaat tussen de clusters gebruikt. Om de meest waarschijnlijke manieren om te scoren voor een team te vinden wordt deze clustering methode toegepast op alle spelpatronen die tot een doelpoging leiden.

2.3.3 Netwerk-gebaseerde karakterisatie

De derde manier om de speelstijl van een team te karakteriseren, is om het gedrag van een team te modelleren met een soort van netwerk.

Cintia et al. [16] ontwikkelden een methode die een pass netwerk van een team in een bepaalde wedstrijd kan opstellen. In dit netwerk stellen de knooppunten de spelers voor en de verbindingen ertussen een pass tussen deze spelers met een bepaald gewicht, afhankelijk van de frequentie van deze passes. Een visualisatie van dit pass netwerk geeft een duidelijke weergave van de belangrijkste pass interacties tussen spelers van een team (figuur 2.5).

Pẽna et al. [48] modelleren het gedrag van een team dan weer als een Markov Netwerk. Voor een simpel Markov Netwerk bestaande uit de toestanden ‘Recovery’, ‘Keep’, ‘Loss’ en ‘Shot’ worden de frequenties van de overgangen berekend voor elk

team op basis van hun vorige wedstrijden. De overgangsfrequenties tussen deze toestanden bepalen dan eigenlijk het gedrag van een team en kunnen vergeleken worden. Een uitgebreider Markov Netwerk met meer toestanden of een netwerk met als toestanden de mogelijke posities van spelers (keeper, verdediging, middenveld, aanval) geven een betere weergave van het gedrag maar hiervoor zijn de overgangsfrequenties dan weer moeilijker te bepalen.

Hoofdstuk 3

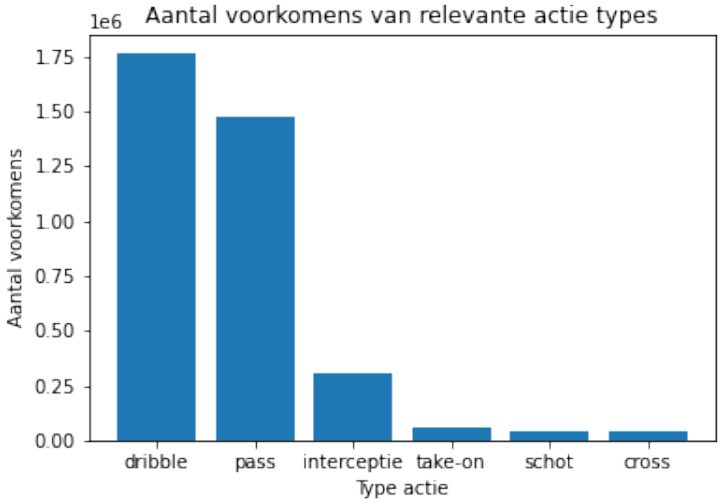
Methodologie voor het opstellen van speelstijlvectoren

In dit hoofdstuk worden de verschillende stappen voor het opstellen van de speelstijlvectoren van een team besproken. Allereerst wordt de algemene benadering toegelicht in sectie 3.1 tot 3.3. In sectie 3.4 wordt de methode voor het opstellen van betekenisvolle *features* die een wedstrijd beschrijven toegelicht. Daarna wordt in sectie 3.5 uitgelegd hoe deze opgestelde *features* gebruikt kunnen worden om speelstijlvectoren te bekomen. In sectie 4.6 wordt besproken hoe de speelstijlvectoren in meer detail kunnen worden geanalyseerd. Tenslotte wordt dit hoofdstuk afgesloten met een overzicht en een bespreking van de gebruikte methodologie in sectie 3.8.

3.1 Identificeren van speelstijl

In deze thesis wordt getracht een methode op te stellen die op automatische wijze een begrijpbare en logische representatie van de speelstijl van elk professioneel voetbalteam kan afleiden uit data. Met de representatie moet het ook mogelijk zijn om de speelstijlen van teams onderling te vergelijken met een afstands- of gelijkenismaat zodat het bijvoorbeeld mogelijk is om teams met gelijkaardige speelstijlen te identificeren voor een oefenmatch. Als we naar het bestaande werk kijken in sectie 2.3, is dit laatste doel eigenlijk enkel te bereiken met een karakterisatie aan de hand van *features* en een netwerk-gebaseerde karakterisatie. Er wordt ook getracht om zoveel mogelijk informatie uit *event data* te integreren die belangrijk is voor het bepalen van de speelstijl van een team. Het opstellen van *features* biedt op dit vlak veel meer flexibiliteit dan een netwerk-gebaseerde oplossing en dus wordt er voor deze aanpak gekozen.

Alvorens een methode wordt opgesteld, is het belangrijk om af te bakenen welke aspecten van de speelstijl van een team af te leiden zijn uit data en in welke mate ze waar te nemen zijn. De speelstijl van een team is namelijk een samenstelling van vele componenten die grofweg op te delen zijn in degene die de offensieve strategie van een team bepalen en degene die de defensieve strategie bepalen. Slechts een aantal van deze componenten kunnen uit *event data* worden afgeleid. De defensieve



Figuur 3.1: De absolute frequenties van de relevante actietypes in Atomic-SPADL formaat voor het seizoen 2019/20 in de vijf grootste voetbal competities.

organisatie van een team valt bijvoorbeeld enkel uit *tracking data* volledig te af te leiden. Voor het bepalen van de defensieve strategie is de enige directe informatie die uit *event data* valt af te leiden het aantal intercepties in een wedstrijd en de locatie en het tijdstip ervan samen met de speler die de bal onderschepte. *Event data* bevat wel veel informatie voor het bepalen van de offensieve strategie van een team. Als we kijken naar de absolute frequentie van elke actie in het seizoen 2019/20 in de vijf grootste competities[1] zien we dat een dribble en een pass veruit het meeste voorkomen in een wedstrijd (figuur 3.1). Een schot heeft slechts een aandeel van 1% van de totale acties in een wedstrijd. Het aantal effectieve kansen (*shot on/off target*) in een wedstrijd is dan ook beperkt en dus is het moeilijk om de offensieve strategie van een team aan de hand hiervan te karakteriseren. De manier waarop men kansen probeert te creëren bepaalt echter in belangrijke mate de kansen die men creëert en *event data* bevat wel veel informatie om dit gedrag af te leiden. De verschillende spelpatronen om een aanval op te bouwen en om in de *final third* te geraken zijn belangrijke elementen die weergeven hoe een team kansen probeert te creëren. Deze spelpatronen worden vooral gekarakteriseerd door opeenvolgingen van passes en dribbels die een team verrichten. Deze acties komen wel veelvuldig voor in *event data* en het is dus makkelijker om de offensieve strategie van een team aan de hand van deze spelpatronen te bepalen.

Aangezien het mogelijk moet zijn om de speelstijlen van teams onderling te vergelijken, dienen een aantal typische spelpatronen voor de offensieve strategie van een team geïdentificeerd te worden. Deze spelpatronen moeten de strategie van elk team kunnen beschrijven en moeten dus algemeen genoeg en herkenbaar zijn voor voetbalanalisten en coaches. Deze typische spelpatronen kunnen door een

[1] Premier League, La Liga, Serie A, Bundesliga, Ligue 1

voetbalanalist of coach opgesteld worden maar deze handmatige aanpak heeft een aantal nadelen. Het bepalen van een typisch spelpatroon is zeer subjectief en het is moeilijk om ze precies te definiëren aan de hand van data. Daarnaast is het ook niet eenvoudig om de spelpatronen uit te kiezen die veelvuldig voorkomen in een wedstrijd maar waarmee de verschillende speelstijlen van teams toch genoeg onderscheiden kunnen worden. Om deze redenen worden de typische spelpatronen die de speelstijl van een team beschrijven op een *data-driven* manier geïdentificeerd.

Een typisch spelpatroon dat beschreven wordt aan de hand van één gebeurtenis in een wedstrijd zal weinig inzicht geven in de speelstijl van een team. Enkel als deze gebeurtenis op een zeer hoog niveau gedefiniëerd is, zal deze inzicht kunnen geven. Het opstellen van zo een gebeurtenis vereist echter dat we eigenlijk al weten wat de mogelijke typische spelpatronen kunnen zijn, wat dus niet het geval is. Een typisch spelpatroon dat beschreven wordt aan de hand van een aantal gebeurtenissen geeft daarentegen wel inzicht in de speelstijl van team. Om de typische spelpatronen te identificeren, moet er dus op zoek worden gegaan naar gebeurtenissen op laag niveau die vaak samen voorkomen in een wedstrijd. Aangezien de typische spelpatronen inzicht moeten geven in de speelstijl van een team dienen deze gebeurtenissen waarmee een typisch spelpatroon dan beschreven wordt, de relevante informatie hiervoor te bevatten. Deze informatie (bv. opeenvolgingen van acties) kunnen we coderen in de kenmerken (d.i. *features*) waarmee een wedstrijd beschreven wordt. Er moet dus op zoek worden gegaan naar de onderliggende structuur van deze *feature* beschrijvingen van wedstrijden door de correlatie tussen *features* te onderzoeken. In grote lijnen valt dit probleem dus te modelleren als een dimensionaliteitsreductie van de *feature* beschrijvingen van wedstrijden van teams. De *feature* beschrijvingen dienen op een *unsupervised* manier omgezet te worden naar een beschrijving in een lagere dimensie waarbij elke nieuwe dimensie overeenkomt met een typisch spelpatroon dat inzicht geeft in de speelstijl van een team. Ook deze lage-dimensie voorstelling moet makkelijk te begrijpen zijn.

Een van de meest gebruikte technieken voor dimensionaliteitsreductie is Principal Component Analysis (PCA) dat de *features* met de hoogste variantie als *principal components* identficeert. Fernandez-Navarro et al. [30] pasten deze techniek al succesvol toe om prestatie-indicatoren die een wedstrijd beschrijven te groeperen in een aantal stijlen. Een nadeel aan deze techniek is echter wel dat de gevonden *principal components* moeilijker te interpreteren zijn aangezien ze een positieve en negatieve waarde kunnen hebben. Bij andere dimensionaliteitsreductie technieken is het interpreteren van de nieuwe samengestelde dimensies ook een probleem. Elke nieuwe dimensie is meestal een functie van alle oorspronkelijke *features* waardoor de lagere dimensie voorstelling enkel volledig kan begrepen worden door alle dimensies samen te beschouwen [53, 17].

3.2 Topic Modeling

Een variatie op de standaard dimensionaliteitsreductie technieken waarbij de interpretatie van de gevonden componenten belangrijk is, is *Topic Modeling*. *Topic Modeling*

wordt gebruikt om onderwerpen te identificeren waarmee documenten (bestaande uit woorden) beschreven kunnen worden. Een document kan dan voorgesteld worden als een samenstelling van deze onderwerpen en een onderwerp als een samenstelling van woorden (uit een op voorhand gedefiniëerd vocabularium). Het voordeel van deze representatie is dat een onderwerp alleenstaand meestal eenvoudiger te interpreteren valt dan de componenten bij standaard dimensionaliteitsreductie technieken. De nieuwe beschrijvingen van de documenten aan de hand van deze onderwerpen zijn ook zeer begrijpbaar [17]. In de volgende sectie 3.3 wordt dit concept en hoe het kan gebruikt worden voor de speelstijl van teams te identificeren verder toegelicht.

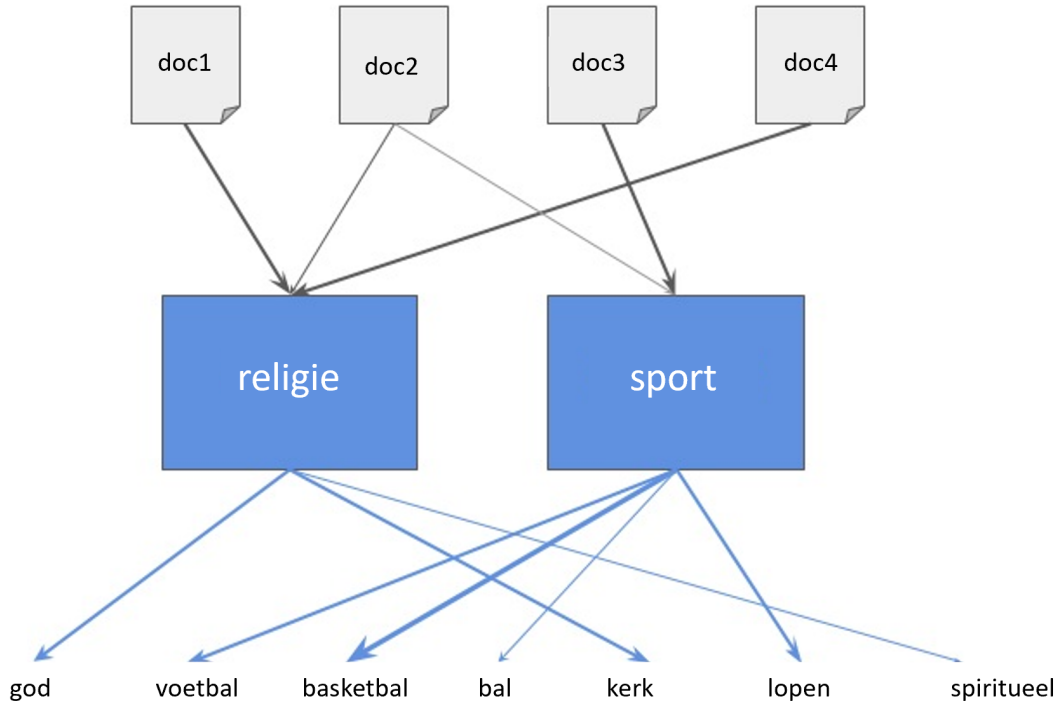
De bekendste *Topic Modeling* technieken waarvan Python implementaties bestaan, zijn Latent Semantic Analysis (LSA), Non-Negative Matrix Factorization (NMF), Probabilistic Latent Semantic Analysis (pLSA) en LDA [17, 40, 72, 37]. Deze technieken vallen op te delen in twee categorieën: matrixfactorisatie (LSA, NMF) en probabilistische (pLSA, LDA) technieken. De matrixfactorisatie technieken hebben als voordeel dat ze efficiënter te berekenen zijn dan de probabilistische technieken. Er bestaan tegenwoordig echter wel benaderende variaties op pLSA en LDA die het trainen van het model versnellen zonder kwaliteitsverlies waardoor dit verschil in praktijk minimaal is [36]. LSA heeft als groot nadeel dat de beschrijvingen van een document en een onderwerp positieve en negatieve waardes bevatten wat het interpreteren van een document of onderwerp moeilijk maakt. Bij NMF, pLSA en LDA zijn deze waardes enkel positief en kunnen ze omgevormd en geïnterpreteerd worden als een kans dat een onderwerp of woord voorkomt.

Terwijl de gewichten van de document-onderwerp en onderwerp-woord relaties bij LSA, pLSA en NMF (complexe) combinaties zijn van de training documenten, zijn de gewichten van LDA een Dirichlet verdeling. Dit zorgt ervoor dat LDA robuuster is tegen *overfitting* en dus beter in staat is om de samenstelling van onderwerpen af te leiden voor een document dat niet gezien werd tijdens de training van het model. Aangezien we een model willen bekomen dat gebruikt kan worden om de speelstijl van alle teams in recente en toekomstige seizoenen af te leiden is dit een zeer belangrijke eigenschap.

3.3 Latent Dirichlet Allocation

Een Dirichlet verdeling codeert de intuïtie dat documenten slechts gerelateerd zijn aan een paar onderwerpen en een onderwerp aan een paar woorden. De kansverdeling van de woord-onderwerp en onderwerp-document relaties is bijgevolg geconcentreerd op een aantal woorden of een aantal onderwerpen. De concentratiesterkte van deze twee kansverdelingen worden ieder bepaald door een parameter α [37].

In LDA wordt een document oorspronkelijk voorgesteld in *Bag-of-Words* (BoW) formaat. Dit houdt in dat voor elk woord in een op voorhand opgesteld vocabularium ($\approx$ *features*) het aantal voorkomens ervan in elk document wordt bijgehouden. De volgorde waarin de woorden voorkomen is dus niet van belang bij deze representatie, enkel het aantal voorkomens. LDA identificeert aan de hand van deze representaties de onderliggende onderwerpen die deze documenten beschrijven. Aangezien



Figuur 3.2: Illustratie van de onderwerp-woord en onderwerp-document relaties in een LDA model. Het gewicht van elke relatie wordt weergeven door de dikte van de lijn ertussen. Enkel de gewichten boven een bepaalde *threshold* worden getoond. Figuur gebaseerd op Ibanez [37].

het aantal onderwerpen op voorhand gegeven is, wijst een LDA model tijdens training eigenlijk enkel de juiste gewichten toe aan alle mogelijke woord-onderwerp en onderwerp-document relaties. Na training kan elk document voorgesteld worden als een geconcentreerde kansverdeling van onderwerpen en elk onderwerp als een geconcentreerde kansverdeling van woorden. Als men naar de woorden met de grootste kans kijkt, is het meestal vrij duidelijk wat het bijhorende onderwerp inhoudt. Figuur 3.2 toont een visualisatie van een hypothetisch LDA model met twee onderwerpen. Het onderwerp waarbij de woorden 'god', 'kerk' en 'spiritueel' een hoge kans hebben (bv. 60%, 30%, 10%), kan bijvoorbeeld duidelijk beschreven worden als 'religie'. Een document kan dan beschreven worden als een kansverdeling van deze onderwerpen (bv. 80% sport en 20% religie).

Het onderliggende idee achter het identificeren van onderwerpen die documenten beschrijven aan de hand van LDA, werd in het werk van Perdomo Meza en Girela [51, 50] succesvol toegepast op het identificeren van begrijpbare componenten die de speelstijl van een voetbalteam beschrijven. Na transformatie met LDA wordt een wedstrijd (document) in hun werk voorgesteld als een kansverdeling van componenten die een bepaalde stijl voorstellen (onderwerpen). Zo een stijl component wordt dan weer voorgesteld als een kansverdeling van *features* die een wedstrijd beschrijven. Het trainen van een LDA model komt hier in grote lijnen neer op het identificeren van

features die vaak samen voorkomen in verschillende wedstrijden en te associëren zijn met een typisch gedrag van een team. Aan de hand van de kansverdeling van deze typische gedragingen kan snel een beeld worden gevormd van hoe een team speelde in een wedstrijd. Deze wedstrijdvectoren kunnen veel afhangen van de tegenstander en het wedstrijdverloop. Het gemiddelde van alle wedstrijdvectoren van dat team in een seizoen kan wel een robuust inzicht geven in hun speelstijl.

De traditionele input van een LDA model is een *Term Frequency Matrix* waarin de rijen overeenkomen met documenten en de kolommen met de mogelijke woorden. Een cel in deze matrix komt overeen met hoeveel een woord in een document voorkomt. Analoog kan een matrix worden opgesteld waarin de rijen wedstrijden voorstellen en de kolommen *features* die een wedstrijd beschrijven. Tabel 3.1 illustreert de analogie die Perdomo Meza en Girela [51, 50] beschreven in hun werk. De *features* die ze in hun werk gebruikten om een wedstrijd te beschrijven is een selectie van Opta's F9 data. Deze data is eigenlijk een uitgebreide vorm van wedstrijdbladdata die het aantal voorkomens van bepaalde gebeurtenissen weergeeft. Het neemt enkel het aantal voorkomens van een enkele gebeurtenis in rekening en niet hoe de verschillende gebeurtenissen elkaar opvolgen. Nochtans zijn de interacties tussen spelers en de opeenvolgingen van gebeurtenissen belangrijke aspecten van de speelstijl van een team. Aangezien de wedstrijdbeschrijvingen (BoW-formaat) geen rekening houden met de volgorde van *features* is het LDA model ook niet in staat om patronen in speler interacties of opeenvolgingen van gebeurtenissen te vinden. In de volgende sectie wordt daarom getracht om *features* op te stellen die de opeenvolging van gebeurtenissen wel in rekening neemt en zoveel mogelijk relevante informatie uit *event data* integreert. Er wordt in deze thesis verder gebouwd op het idee van Perdomo Meza en Girela [51, 50] om met LDA typische spelpatronen te identificeren die de speelstijl van een team kunnen beschrijven.

3.4 Feature constructie

De BoW representatie die als invoer dient voor het LDA model geeft enkel het aantal voorkomens van elke *feature* weer. Er wordt dus geen rekening gehouden met de volgorde waarin een *feature* voorkomt. De typische spelpatronen die geïdentificeerd dienen te worden, zijn echter opeenvolgingen van acties door verschillende spelers. Om deze dus te kunnen identificeren, moet deze informatie gecodeerd worden in de *features*. Daarnaast is het ook belangrijk welke informatie per gebeurtenis er in rekening wordt gebracht. In *event data* zijn er drie hoofdcategoriën van informatie per gebeurtenis: de betrokken speler(s), de locatie en het type van de gebeurtenis [67].

Aangezien er typische spelpatronen dienen geïdentificeerd te worden die de speelstijl van elk team kunnen beschrijven, is het niet wenselijk om de identiteit van de betrokken speler(s) in rekening te nemen. Wel is het belangrijk om te kijken naar het aantal spelers die betrokken zijn in een opeenvolging van gebeurtenissen.

Er bestaan verschillende methodes om de locatie van een gebeurtenis te beschrijven. Een mogelijkheid is bijvoorbeeld om het veld te verdelen in een aantal zones. Dit kan door een raster over het veld te leggen of het veld te verdelen op een *data-*

Tabel 3.1: De analogie tussen de *Term Frequency Matrix* van documenten (a) en die van wedstrijd beschrijvingen (b). Tabel gebaseerd op Perdomo Meza en Girela [51, 50].

(a) Document *Term Frequency Matrix*

	god	voetbal	basketbal	bal	kerk	lopen	spiritueel
Doc 1	3	1	3	2	6	3	5
Doc 2	2	5	0	5	0	4	1
Doc 3	0	0	4	3	1	4	2
Doc 4	4	2	1	3	1	5	3

(b) Wedstrijd *Term Frequency Matrix*

	passes in *final third*	intercepties	schot *off target*	luchtduels	corners
Weds 1	10	1	4	3	1
Weds 2	4	2	1	3	1
Weds 3	3	1	3	2	6
Weds 4	2	5	0	5	0

driven manier of op basis van domeinkennis [66, 22]. Twee opeenvolgingen van acties zijn gelijkaardig als er aantal zones van de acties gelijk zijn. Een nadeel van deze methode is dat twee opeenvolgingen van acties die dezelfde volgorde van naburige locaties hebben toch als verschillend kunnen worden gezien als de locaties net in een andere zone liggen. Een andere mogelijkheid, waar dit probleem niet voorvalt, is om opeenvolgingen van acties voor te stellen met de exacte locatie van elke actie. Twee opeenvolgingen van acties kunnen dan vergeleken worden met DTW [22] of de euclidische afstand [70] als afstandsmaat. Twee opeenvolgingen van acties met nabije locaties zullen dan wel steeds als gelijkaardig worden beschouwd.

Aan de hand van deze voorstellingen kunnen opeenvolgingen van acties ruimtelijk gegroepeerd worden. Er kan dan voor elke groep een vertegenwoordiger gekozen worden waarmee andere opeenvolgingen van acties kunnen vergeleken worden. Bij de eerste voorstelling zullen deze vertegenwoordigers opeenvolgingen zijn van acties waarbij de locaties worden voorgesteld als één van de op voorhand gekozen zones. De keuze van deze zones bepaalt dus de mogelijke vertegenwoordigers. Het is echter niet eenvoudig om te bepalen hoe groot of klein het gebied moet zijn dat een zone omvat. Bij de tweede voorstelling worden de locaties van deze vertegenwoordigers echter op een *data-driven* manier bepaald. De opeenvolgingen van acties waar veel andere opeenvolgingen van acties in een groep gelijkaardig aan zijn, kunnen dan gekozen worden als vertegenwoordiger. De locaties van deze vertegenwoordigers kunnen dus eender welke (x,y) positie zijn die voorkwam in één van de opeenvolgingen van acties. Door deze twee belangrijke voordelen worden opeenvolgingen van acties voorgesteld met de exacte (x,y) posities van de acties.

Voor de laatste categorie van informatie per gebeurtenis is het nuttig om te bekijken welke soort gebeurtenissen relevant zijn voor het beschrijven van typische

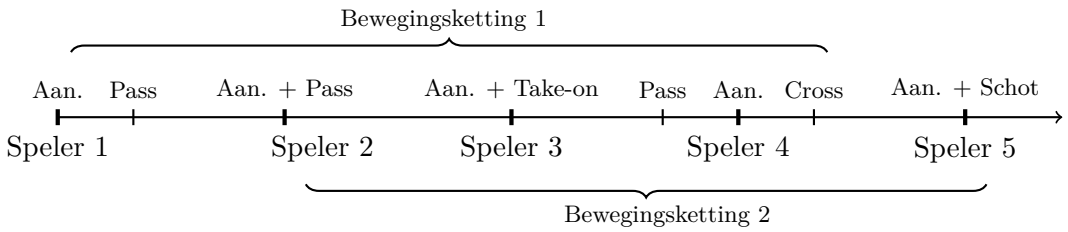
spelpatronen. De verschillende soorten gebeurtenissen zijn op te delen in stilstaande fases en veldspel acties. Stilstaande fases zoals vrije trappen, corners en doeltrappen zijn belangrijk om te analyseren maar zijn minder belangrijk voor het karakteriseren van de speelstijl van een team. Om deze redenen beschouwen we enkel veldspel acties. Van de veldspel acties zijn dribbels en passes veruit het meest frequent in een wedstrijd (figuur 3.1). In SPADL formaat betekent een dribbel met de bal aan de voet vooruit lopen zonder dat daarbij een tegenstander wordt uitgeschakeld. Aangezien deze actie 43.6% van de acties uitmaakt en weinig bijdraagt aan een spelpatroon, wordt deze actie niet in beschouwing genomen. Een pass, een *take-on* (iemand voorbij dribbelen), een schot, een cross en een balaanname worden wel in beschouwing genomen. Er wordt wel geen onderscheid gemaakt tussen de verschillende soorten acties aangezien vooral de bewegingen van een team belangrijk zijn voor het vinden van typische spelpatronen. Enkel de start- en een eventuele eindlocatie worden in rekening gebracht.

3.4.1 Opsplitsen in bewegingskettingen

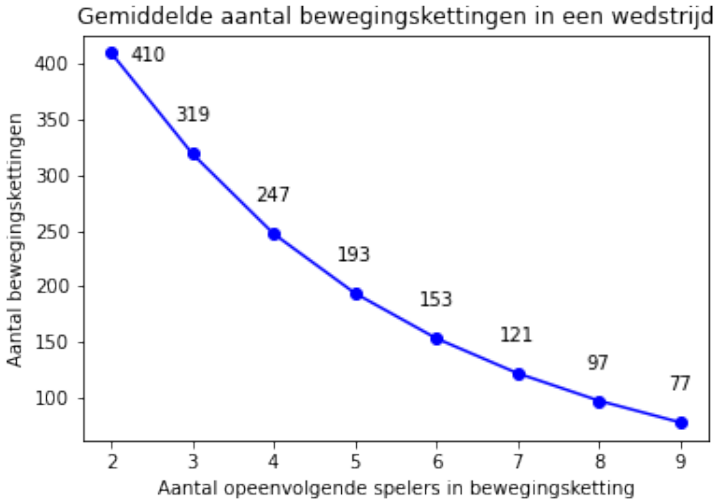
Event data is een sequentie van alle gebeurtenissen in een wedstrijd. Om deze sequentie te kunnen gebruiken om een wedstrijdbeschrijving op te stellen, dient deze opgesplitst te worden in gebeurtenissen die samen horen. Een intuïtieve manier is het opsplitsen in periodes waarin een team ononderbroken balbezit heeft. Een nadeel van deze manier is dat deze fases soms lang kunnen zijn en slechts delen van deze fases relevant zijn voor het beschrijven van een typisch spelpatroon. Om dus de verschillende fases te kunnen vergelijken, dient een fase verder opgesplitst te worden in kleinere delen. Zo een kleiner deel moet lang genoeg zijn zodat het genoeg informatie bevat maar ook niet te lang zodat er genoeg gelijkaardige delen bestaan in andere wedstrijden.

Een belangrijke keuze bij het opsplitsen is of de verschillende delen mogen overlappen of niet. Het nadeel van overlapping is dat sommige gebeurtenissen in meerdere delen aanwezig zijn en hun voorkomens dus meerdere keren worden geteld. Het voordeel is dan weer dat elk mogelijk relevant deel van een fase in rekening wordt genomen wat het vinden van typische spelpatronen bevordert. Aangezien het identificeren van typische spelpatronen een hoofddoel is in onze methode wordt er voor overlapping gekozen.

Een andere keuze die gemaakt moet worden, is of er per gebeurtenis of per betrokkenheid van een speler wordt opgesplitst. De typische spelpatronen van teams worden eerder gekarakteriseerd door het samenspel tussen verschillende spelers en minder door de hoeveelheid gebeurtenissen. Bijvoorbeeld een fase waarbij een speler alleen het veld oversteekt met dribbels onderweg is meer eigen aan de speelstijl van die speler dan die van een team. Geïnspireerd door het werk van Stats Perform [70] wordt een fase daarom opgesplitst per aantal spelers die betrokken zijn in een opeenvolging van acties. In plaats van enkel de eerste aanraking van een speler in rekening te nemen zoals bij Stats Perform worden alle relevante acties van die speler in rekening gebracht: balaannames, passes, crosses, *take-ons* en schoten. Figuur 3.3 illustreert hoe een fase opgesplitst wordt in de verschillende delen die vanaf nu



Figuur 3.3: Illustratie van de opsplitsing van een fase in bewegingsketting met vier spelers. Het kan dat een speler meerdere keren voorkomt in een bewegingsketting. Bijvoorbeeld speler 1 en speler 3 zouden allebei dezelfde speler kunnen voorstellen maar speler 1 en 2 dan weer niet. De aanname van een pass of voorzet wordt afgekort als Aan.



Figuur 3.4: Het gemiddelde aantal bewegingsketting in een wedstrijd per aantal spelers dat beschouwd wordt in een bewegingsketting.

bewegingsketting worden genoemd. Figuur 3.6b toont een visualisatie van een bewegingsketting.

Het laatste wat nog beslist moet worden is hoeveel opeenvolgende spelers er in elke bewegingsketting in beschouwing moeten worden genomen. Hoe meer spelers, hoe meer informatie elke bewegingsketting bevat en hoe duidelijker en unieker deze zal zijn. Langs de andere kant zal het aantal bewegingsketting in een wedstrijd minder zijn als er meer spelers in beschouwing worden genomen. Drie of vier spelers in beschouwing nemen, lijkt een redelijke *trade-off* tussen de hoeveelheid informatie per bewegingsketting en het aantal bewegingsketting in een wedstrijd (figuur 3.4). Het precieze aantal wordt in de volgende sectie bepaald aan de hand van de resultaten bij elke opsplitsing.

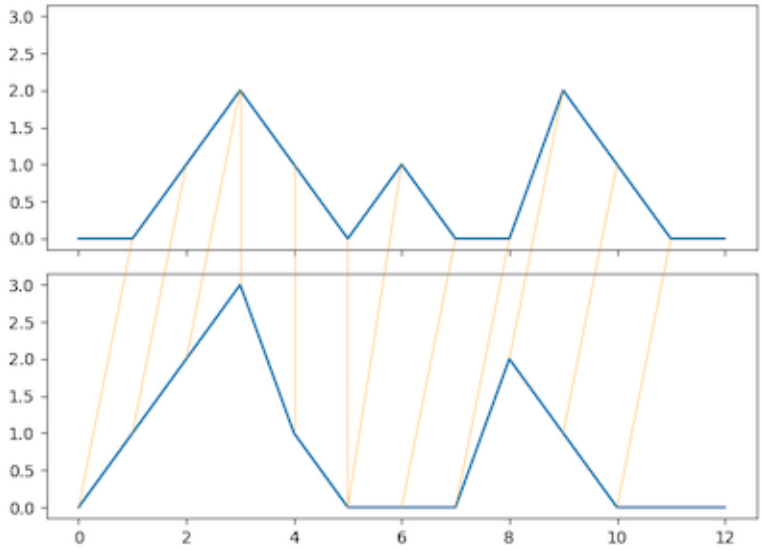
3.4.2 Groeperen van bewegingsketten

Om een wedstrijd te kunnen beschrijven aan de hand van bewegingsketten dienen de ruimtelijk overeenkomstige bewegingsketten gegroepeerd te worden. De bewegingsketten in elke groep kunnen dan voorgesteld worden door een vertegenwoordiger uit die groep die een vergelijkbare vorm en positie dient te hebben als alle andere bewegingsketten in die groep. Deze vertegenwoordigers kunnen op basis van domein kennis worden opgesteld of op een *data-driven* manier. Om een wedstrijdbeschrijving op stellen aan de hand van deze vertegenwoordigers ($\approx$ *features*) moet het mogelijk zijn om een bewegingsketting op automatische wijze te matchen met één van de vertegenwoordigers met behulp van een afstands- of gelijkenismaat. Men kan deze afstandsmaat ook gebruiken om een aantal bewegingsketten te groeperen zodat een vertegenwoordiger in elke groep kan worden gevonden die een kleine afstand heeft ten op zichte van de andere bewegingsketten in die groep.

Dynamic Time Warping (DTW)

Het kiezen van de juiste afstands- of gelijkenismaat voor het vergelijken van bewegingsketten bepaalt dus in belangrijke mate de kwaliteit van de *features* en de wedstrijdsbeschrijvingen aan de hand hiervan. De afstandsmaat moet sequenties die op ruimtelijk en temporeel vlak gelijkaardig zijn, kunnen identificeren. Daarnaast moet het ook kunnen omgaan met het feit dat de bewegingsketten verschillend van lengte zijn. Een bekende techniek die aan deze voorwaarden voldoet is Dynamic Time Warping (DTW) [47]. Decroos et al. [22] gebruikten deze techniek succesvol in een *pre-processing* stap om fases die gelijkaardig zijn op ruimtelijk en temporeel vlak te groeperen. DTW berekent een kost door het vinden van een *one-to-many matching* tussen twee sequenties zoals figuur 3.5 illustreert. Deze soort matching zorgt ervoor dat DTW kan omgaan met kleine mismatches tussen sequenties zoals verschuivingen of vertragingen. Een nadeel van DTW is wel dat het geen echte afstandsmaat is omdat het niet voldoet aan de driehoeksongelijkheid. In de context waarin DTW hier gebruikt wordt, is dit echter geen probleem [22].

De (naïeve) manier om een DTW kost te berekenen en uit te leggen is met Dynamic Programming. Gegeven een bewegingsketting B_1 van lengte m en een bewegingsketting B_2 van lengte n kan de DTW kost berekend worden als:
$D[i,j] = \delta(B_{1_i}, B_{2_j}) + min(D[i-1,j-1], D[i,j-1], D[i-1,j])$
waarbij D een $m \times n$ matrix voorstelt en $\delta(B_{1_i}, B_{2_j})$ de kost om het i^{ste} element van B_1 met het j^{ste} element van B_2 te aligneren. Deze kost wordt berekend als de euclidische afstand tussen de locaties van de i^{ste} en j^{ste} gebeurtenis. De kostfunctie $\delta(B_{1_i}, B_{2_j})$ neemt dus enkel de dichtheid op ruimtelijk vlak in beschouwing. De finale DTW kost bevindt zich in $D[m,n]$ [22]. De implementatie die gebruikt wordt om de DTW kost tussen twee bewegingsketten te berekenen, is die van het Python pakket *dtaidistance* [43].



Figuur 3.5: Een visuele voorstelling van het *warping path* van een *alignment* tussen twee 1-dimensionale sequenties met DTW.

Clustering

Om de tekst leesbaarder te maken, worden *features* (d.i. het aantal typische bewegingskettingen in een wedstrijd) en *medoids* (d.i. een typische bewegingsketting) soms door mekaar gebruikt. Afhankelijk van de context zal het echter steeds duidelijk zijn wat er juist bedoeld wordt.

Nu een afstandsmaat is gekozen, kunnen we deze gebruiken om een aantal bewegingskettingen te groeperen met een clustering algoritme om een vertegenwoordiger van elke groep te identificeren. Deze vertegenwoordiger moet een zo klein mogelijke afstand hebben tot de andere bewegingskettingen in die groep. Daarnaast moet hij ook een realistische bewegingsketting voorstellen zodat de typische spelpatronen die er mee beschreven worden begrijpbaar en logisch zijn. Idealiter is het dus één van de bewegingskettingen in die groep en bijvoorbeeld geen gemiddelde van alle bewegingskettingen in die groep.

Grofweg genomen zijn er vier verschillende clustering manieren: partitionering, model-gebaseerde, hiërarchische en *density-based* clustering [18]. Partitionering clustering leent zich het beste als oplossing voor ons probleem. De twee bekendste algoritmes zijn *K-means* en *K-medoids*. Deze soort clustering algoritmes verdelen een aantal datapunten over een op voorhand gedefiniëerd aantal clusters. In elke cluster trachten ze de som van de afstand van elk datapunt tot een referentie datapunt van die cluster zo klein mogelijk te maken. Bij *K-means* is dit referentie datapunt het gemiddelde van alle datapunten in die cluster. Dit gemiddelde als vertegenwoordiger nemen kan onrealistische bewegingskettingen opleveren en bij testen bleek dit ook het geval. Bij *K-medoids* is dit referentie datapunt één van de datapunten in die cluster [18]. Dit referentie datapunt (een *medoid*) in *K-medoids* voldoet perfect aan

de eisen waar een vertegenwoordiger moet aan voldoen.

Een ander mogelijk partitionering algoritme dat net zoals *K-medoids* een datapunt als referentiepunt neemt, is *Affinity Propagation* [31]. Dit algoritme bepaalt zelf het aantal clusters dat geschikt is om de data te beschrijven. Aangezien er gemiddeld maar 247 (4 spelers) of 319 (3 spelers) bewegingskettingen in een wedstrijd bestaan, is het belangrijk dat het aantal clusters beperkt blijft zodat er niet teveel *features* zijn. Hoe meer *features* er namelijk zijn, hoe minder elke *feature* gemiddeld voorkomt in een wedstrijd en hoe minder makkelijk het LDA model typische spelpatronen kan herkennen in wedstrijden. Langs de andere kant mag dit aantal ook niet te klein zijn zodat er genoeg unieke bewegingspatronen kunnen beschreven worden. Bij *Affinity Propagation* wordt het aantal clusters beïnvloed door de *preference* waarde maar in praktijk bleek het moeilijk om ongeveer het gewenste aantal clusters te bekomen. Het is dus handig om een algoritme als *K-medoids* te gebruiken waarbij het aantal clusters zelf te bepalen is.

Een model-gebaseerde clustering techniek zoals een *Gaussian Mixture Model* [55] zou zich ook goed lenen tot het het vinden van vertegenwoordigers maar dan in de vorm van kansverdelingen. Er bestaan echter geen publieke implementaties waarbij DTW als afstandsmaat kan worden gebruikt en het model zelf implementeren is niet zo eenvoudig. De kansverdeling voorstelling van elke vertegenwoordiger verschilt op het eerste zicht ook niet zoveel met bijvoorbeeld *K-medoids* gecombineerd met een *k-nearest neighbours* aanpak voor het matchen van bewegingskettingen met een vertegenwoordiger. Deze methode werd daarom niet verder getest maar is wel interessant voor verder onderzoek.

De andere twee clustering manieren lenen zich minder tot het bekomen van vertegenwoordigers die aan de opgestelde eisen voldoen. Hiërarchische clustering is een goede manier voor het vinden van compacte clusters [18]. Deze clusters zijn echter niet gecentreerd rond één bepaald punt en dus is het moeilijker om een vertegenwoordiger te vinden die een kleine afstand heeft ten opzichte van de andere bewegingskettingen in die cluster. *Density-based* clustering is dan weer minder geschikt omdat bewegingskettingen in druk bezette zones op het veld samen gegroepeerd worden [18]. Het is juist interessant om verschillende vertegenwoordigers te hebben in deze druk bezette zones waarin waarschijnlijk veel belangrijke bewegingskettingen aanwezig zijn. In praktijk vonden algoritmes zoals DBSCAN [27] en OPTICS [6] (zelfs na optimale parameter *tuning*) steeds maar één of twee clusters.

K-medoids

Omdat *K-medoids* perfect voldoet aan de gestelde eisen en er veel Python implementaties bestaan die met DTW als afstandsmaat kunnen worden uitgevoerd, wordt voor dit algoritme gekozen. De *K-medoids* implementatie die gebruikt wordt, is die van het Python pakket *pyclustering* [49].

K-medoids heeft tijdscomplexiteit $\mathcal{O}(n^2)$ [4] wat het niet mogelijk maakt om de bewegingskettingen van alle wedstrijden in een seizoen in vijf competities ($> 800\,000$) als invoer te nemen van het algoritme. Daarom worden er 5000 bewegingskettingen willekeurig bemonsterd uit deze verzameling van bewegingskettingen. Deze 5000

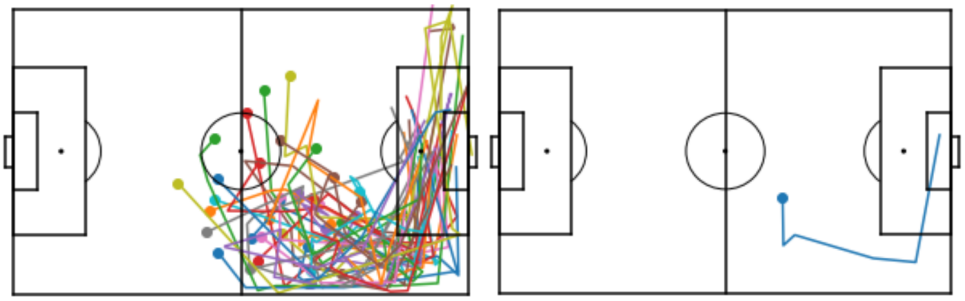
bewegingsketting komen nog steeds overeen met ongeveer 20 wedstrijden dus normaal zou dit weinig invloed mogen hebben op de gevonden clusters. In praktijk werd dit deels bevestigd aangezien de gemiddelde DTW kost om een van de bemonsterde bewegingsketting met zijn dichtstbijzijnde cluster te matchen amper (≈ 2 DTW kost) verschilde met de gemiddelde kost om alle bewegingsketting te matchen.

Een belangrijke parameter bij het uitvoeren van *K-medoids* is het aantal clusters. Deze keuze wordt echter beperkt door het gemiddelde aantal bewegingsketting in een wedstrijd. Bij drie spelers zijn dit er 319 en bij vier spelers 247. Om typische spelpatronen met het LDA model te kunnen identificeren, lijkt het redelijk om het aantal clusters zo te kiezen dat er gemiddeld in een wedstrijd minstens twee voorkomens zijn van elke typische bewegingsketting. Daarnaast is het ook belangrijk dat er genoeg unieke bewegingspatronen zijn. Een cluster aantal van 100 bleek een goede *trade-off* tussen deze twee aspecten. De totale DTW kost om elke bewegingsketting in een cluster met zijn bijhorende *medoid* te matchen nam niet veel meer af bij grotere cluster aantallen.

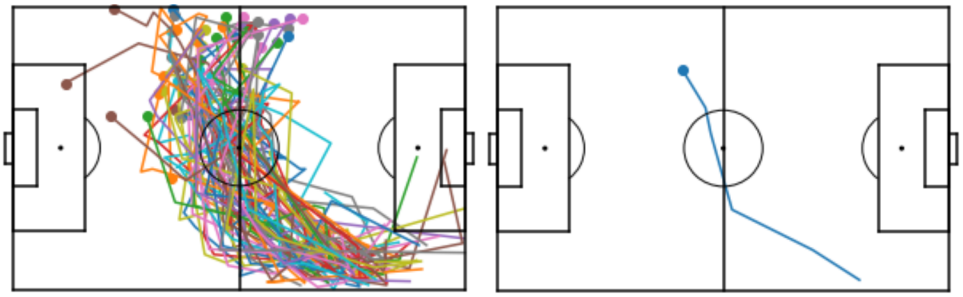
Het algoritme werd een aantal keer met verschillende initiële *medoids* uitgevoerd en de clustering met de meest logische en realistische bewegingsketting als *medoids* werd uiteindelijk gekozen. Er werd ook gekeken of de clusters logisch en compact waren gevormd. Daarnaast werden de bewegingsketting met drie en met vier spelers ieder apart geclusterd om te bekijken welke opsplitsing de beste resultaten geeft. Veel van de bewegingsketting die als *medoid* werden gevonden bij de drie speler opsplitsing waren kort, waardoor ze visueel weinig informatief zijn. Bij de vier speler opsplitsing was dit veel minder het geval. Aangezien de visuele voorstelling van de bewegingsketting enorm belangrijk is bij het interpreteren van geïdentificeerde typische spelpatronen en het verschil in aantal bewegingsketting per wedstrijd bij de drie en vier speler opsplitsing relatief klein is, wordt er voor de vier speler opsplitsing gekozen.

De gevonden clusters zijn compact en geven een duidelijke trend weer (figuur 3.6). De *medoid* van elke cluster komt ook goed overeen met de algemene trend van een cluster en is dus een goede vertegenwoordiger van deze bewegingsketting. Daarbovenop stellen de *medoids* ook logische bewegingen van een team voor op het veld. Wanneer elke bewegingsketting met zijn dichtstbijzijnde cluster wordt gematcht, verschillen deze aantallen soms hard tussen clusters (figuur 3.7). Dit valt te wijten aan het feit dat bewegingen in de eigen helft meer voorkomen dan bewegingen in de helft van de tegenstander.

Bij het analyseren van de gevonden clusters valt het op dat er weinig trends zijn die in of naar de *final third* gaan. De bewegingen in of naar de *final third* zijn echter wel belangrijke elementen die de offensieve strategie van een team bepalen. De initiële typische spelpatronen die met LDA geïdentificeerd werden, waren dan ook enkel opbouwende spelpatronen waarvan veel niet interessant of herkenbaar. Dit is een logisch gevolg van het feit dat een team meer balbezit heeft in de eigen helft en er dus ook meer bewegingsketting in of rond de eigen helft werden gevonden met de clustering. Om meer offensieve spelpatronen te identificeren met LDA dienen er dus meer *features* te zijn die het gedrag in de *final third* beschrijven. Typisch aan de aanvallende patronen van een team in de *final third* is dat er minder spelers

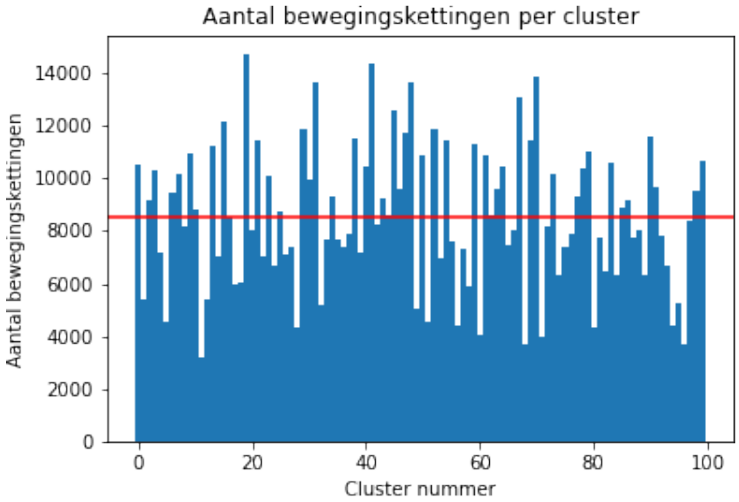


(a) Cluster 18 (b) De *medoid* van cluster 18

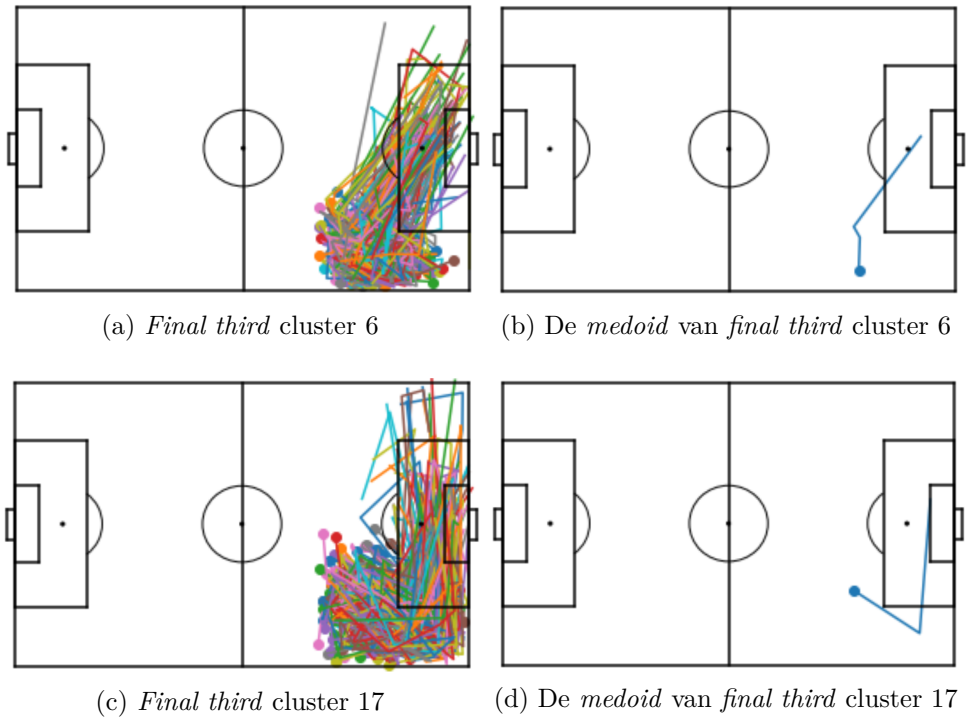


(c) Cluster 22 (d) De *medoid* van cluster 22

Figuur 3.6: Een visualisatie van de bewegingskettingen in de 18^{de} (a) en 22^{ste} (c) cluster die gevonden werden met *K-medoids*. Naast elke cluster wordt telkens de *medoid* van die cluster getoond.



Figuur 3.7: Aantal bewegingskettingen per cluster. De bewegingskettingen van alle wedstrijden in het seizoen 2019/20 in de top vijf competities werden in beschouwing genomen. De rode lijn stelt het gemiddelde aantal bewegingskettingen in een cluster voor.



(a) *Final third* cluster 6 (b) De *medoid* van *final third* cluster 6

(c) *Final third* cluster 17 (d) De *medoid* van *final third* cluster 17

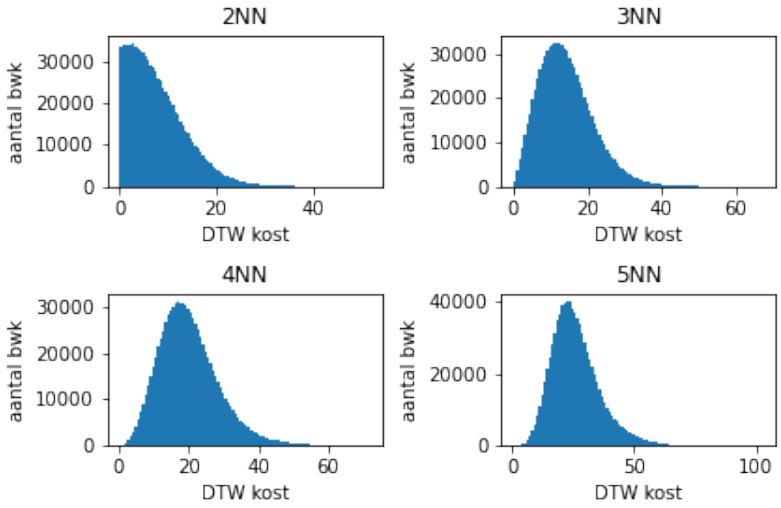
Figuur 3.8: Een visualisatie van de bewegingsketting in de 6^{de} (a) en 17^{de} (c) *final third* cluster die gevonden werden met *K-medoids*. Naast elke cluster wordt telkens de *medoid* van die cluster getoond.

betrokken zijn omdat het moeilijker is om een pass te geven door de hogere druk van de tegenstander in deze zone. Een opsplitsing van fases in bewegingsketting met vier spelers resulteert dus ook in weinig bewegingsketting in de *final third*.

Om unieke bewegingsketting in de *final third* te bekomen, wordt er een aparte clustering uitgevoerd op de bewegingsketting die enkel in de *final third* plaatsvinden. Aangezien er dus meestal weinig spelers betrokken zijn in de *final third* worden er maar twee spelers in beschouwing genomen in een bewegingsketting. Het aantal clusters wordt beperkt tot 25 omdat er slechts een derde van het veld in beschouwing wordt genomen. Veel van de gevonden clusters stellen manieren voor om het strafschopgebied van de tegenstander binnen te geraken wat waardevolle informatie is voor het identificeren van de offensieve strategie van een team (figuur 3.8).

3.5 Opstellen van training data

Nu de *features* zijn geconstrueerd, kunnen de wedstrijdbeschrijvingen aan de hand van deze *features* worden opgesteld om een *Term Frequency Matrix* te bekomen. Deze matrix wordt opgesteld door elke bewegingsketting in een wedstrijd te vergelijken met elke typische bewegingsketting. De standaard manier om dit te doen is *1-nearest-neighbor* (1NN) waarbij de *feature* bijhorend bij de dichtstbijzijnde typische



Figuur 3.9: Het verschil in DTW kost tussen de dichtstbijzijnde *feature* en de 2^{de}, 3^{de}, 4^{de} en 5^{de} dichtstbijzijnde *feature*. De y-as geeft het aantal bewegingsketting weer.

bewegingsketting verhoogd wordt met één. Het gemiddelde verschil tussen de eerste en de tweede dichtstbijzijnde typische bewegingsketting bedraagt bij een groot deel van de bewegingsketting echter een DTW kost tussen nul en vijf (figuur 3.9). In vergelijking met de gemiddelde kost om een bewegingsketting met zijn dichtstbijzijnde *feature* te matchen, 30.2, is dit verschil relatief klein. Daarom werden er drie *distance-weighted k-nearest neighbor* manieren geprobeerd bij het matchen van een bewegingsketting met de *features*. Bij deze drie manieren is de totale waarde die wordt verdeeld over de *features* steeds één zodat elke bewegingsketting evenveel belang heeft. Er wordt telkens rekening gehouden met de DTW kost van een bewegingsketting tot een dichtstbijzijnde *feature*.

De eerste manier is een *optimal weighting scheme* ontworpen door Samworth [58] waarbij de gewichten worden berekend door de volgende formule:

$$w_i = \frac{1}{k}\left[1 + \frac{d}{2} - \frac{d}{2k^{\frac{2}{d}}}\{i^{1+\frac{2}{d}} - (i-1)^{1+\frac{2}{d}}\}\right] \tag{3.1}$$

voor alle $i = 1, ..., k$ en 0 voor alle $i > k$. De parameter d stelt de DTW kost voor en k het aantal dichtstbijzijnde *features* die in beschouwing worden genomen.
De tweede manier berekent het gewicht als de inverse van de DTW kost. De waarde die toegekend wordt aan een *feature* is het genormaliseerde gewicht ten opzichte van de andere *feature* gewichten [28]:

$$w_i = \frac{\frac{1}{d_i}}{\sum_{j=1}^{k} \frac{1}{d_j}} \tag{3.2}$$

De derde manier is ontworpen door Dudani [26] en wordt geformuleerd als:

$$w_i = \frac{\frac{d_k - d_i}{d_k - d_1}}{\sum_{j=1}^{k} \frac{d_k - d_j}{d_k - d_1}} \tag{3.3}$$

Bij de tweede manier worden de gewichten enkel bepaald door de DTW kost van de dichtstbijzijnde *features*. Als er dus een groot verschil is tussen de dichtstbijzijnde *features* zal dit zich ook hard uiten in de gewichten. Bij de eerste en derde manier wordt er ook rekening gehouden met de DTW kosten maar wordt er onafhankelijk van deze kosten wel meer gespreid over de dichtstbijzijnde *features*. Bij de eerste manier zijn de gewichten zelfs meer afhankelijk van het aantal dichtstbijzijnde *features* dan van de DTW kost ervan.

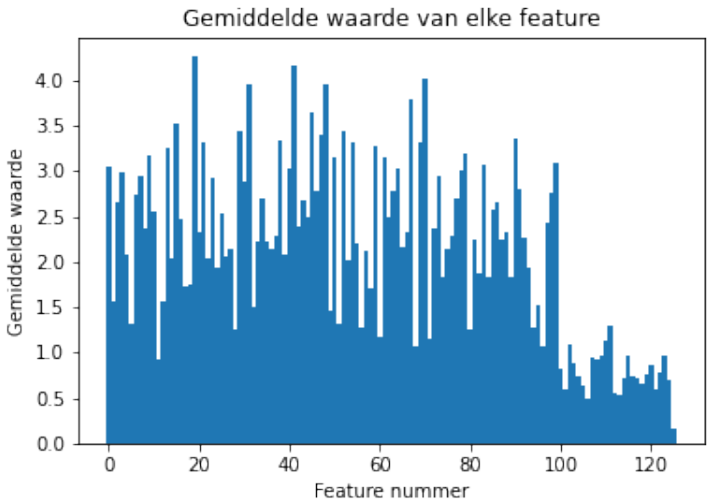
Een belangrijke parameter die bij deze manieren moet bepaald worden, is hoeveel dichtstbijzijnde *features* er in beschouwing worden genomen. Om dit te bepalen is het nuttig om te kijken naar het gemiddelde verschil in afstand tussen de dichtstbijzijnde *feature* en de daarna dichtstbijzijnde *features*. Het gemiddelde verschil tussen de derde dichtstbijzijnde *feature* en de dichtstbijzijnde *feature* bedraagt slechts bij een klein deel van de bewegingskettingen een kost tussen nul en tien. De derde manier wordt met drie dichtstbijzijnde *features* getest omdat de formule niet werkt met twee dichtstbijzijnde *features*. De eerste manier wordt ook getest met drie dichtstbijzijnde *features* omdat de derde dichtstbijzijnde *feature* altijd weinig gewicht krijgt. Vanaf de vierde dichtstbijzijnde *feature* is het verschil bij de overgrote meerderheid groot genoeg om te besluiten dat de deze *feature* niet interessant genoeg is om in beschouwing te nemen.

Appendix B bevat de evaluatie van de gemiddelde *feature*-beschrijving van een wedstrijd voor een team. Deze evaluatie toont aan dat de *features* geschikt zijn voor het karakteriseren van de speelstijl van een team.

Feature normalisatie

De gemiddelde waardes van de *features* in een wedstrijd lopen hard uiteen (figuur 3.10). Vooral de waardes van de *final third features* zijn beduidend lager dan de rest van de *features*. Dit is een logisch gevolg van het mindere balbezit dat een team heeft in de *final third* dan op de rest van het veld. Door de grote verschillen van de gemiddelde waarde kunnen de *features* die gemiddeld meer voorkomen sneller geïdentificeerd worden als een belangrijke *feature* in een typisch spelpatroon, ook al is deze *feature* eigenlijk minder belangrijk. Om de *features* gelijkwaardig te kunnen vergelijken, dienen ze gestandaardiseerd te worden.

Bij het identificeren van onderwerpen die documenten beschrijven met LDA wordt er regelmatig met een *term frequency-inverse document frequency* matrix (tf-idf) gewerkt om de resultaten te verbeteren [8]. De bedoeling van tf-idf is om het relatief belang van een woord in een document voor te stellen door de frequentie van een woord te wegen met het aantal documenten waarin een woord voorkomt. Een woord zoals ‘de’ dat weinig betekenis heeft maar wel heel vaak voorkomt, krijgt hierdoor een lager gewicht. In de context van wedstrijdbeschrijvingen is deze voorstelling echter



Figuur 3.10: De gemiddelde waarde van elke *feature* in een wedstrijd. De laatste 25 *features* zijn de *final third features*.

minder geschikt. Het aantal *features* om een wedstrijd te beschrijven is namelijk veel kleiner dan het aantal woorden in een vocabularium. Daarnaast zijn er ook veel *features* die in bijna elke wedstrijd voorkomen maar daarom niet minder belangrijk zijn. Om het voorkomen van elke *feature* op dezelfde schaal te bekijken, wordt er daarom met de z-score van een *feature* gewerkt. Perdomo Meza en Girela [51, 50] slaagden er met deze *feature* transformatie succesvol in om begrijpbare stijlen te identificeren. Een *feature* geeft zo niet meer het voorkomen van die *feature* weer, maar hoeveel standaardafwijkingen die *feature* meer voorkomt dan gemiddeld. Zonder deze transformatie was het niet mogelijk om typische spelpatronen te identificeren waarbij een *final third feature* belangrijk was.

Een andere ontwerpkeuze is hoe de wedstrijdbeschrijvingen te aggregeren voor ze aan het LDA model worden doorgegeven. Aangezien we op zoek zijn naar een speelstijl beschrijving van een team zou de gemiddelde wedstrijdbeschrijving van een specifiek team over een volledig seizoen gebruikt kunnen worden. Het LDA kon met deze voorstelling echter veel minder begrijpbare en logische typische spelpatronen identificeren. Ook bij de evaluatie presteerde het model beduidend minder goed. Dit resultaat is niet onlogisch omdat een team in wedstrijden tegen sterkere tegenstanders vaak anders speelt dan tegen zwakkere tegenstanders. Door deze fluctuatie van de speelstijl in verschillende wedstrijden is het makkelijker voor een LDA model om op wedstrijdniveau typische spelpatronen te identificeren.

Daarnaast werd er een extra *feature* toegevoegd aan de wedstrijdbeschrijvingen die de lange ballen in een wedstrijd telt. Dit is een aspect dat de bewegingskettingen niet genoeg kunnen weergeven. Een lange bal wordt gedefiniëerd als een pass van minstens 25 meter die in de *defensive third* vertrekt en eindigt op de helft van de tegenstander. Met deze toevoeging was het LDA model in staat om een typisch

spelpatroon te identificeren bestaande uit deze statistiek en bewegingskettingen die het spelen van een lange bal illustreren.

Andere statistieken die uitgetest werden, zorgden niet voor een verbetering van de gevonden spelpatronen en maakten de spelpatronen vaak zelfs minder logisch en begrijpbaar. Voorbeelden van deze statistieken zijn het aantal intercepties (al dan niet in bepaalde zones), de gemiddelde tijd of afstand om op de helft van de tegenstander te geraken na een interceptie, de gemiddelde tijd van een bewegingsketting en passes per defensieve actie (PPDA) [1]. Het doel van deze statistieken was om het defensief en het counter gedrag van een team te integreren maar via *features* was dit niet mogelijk.

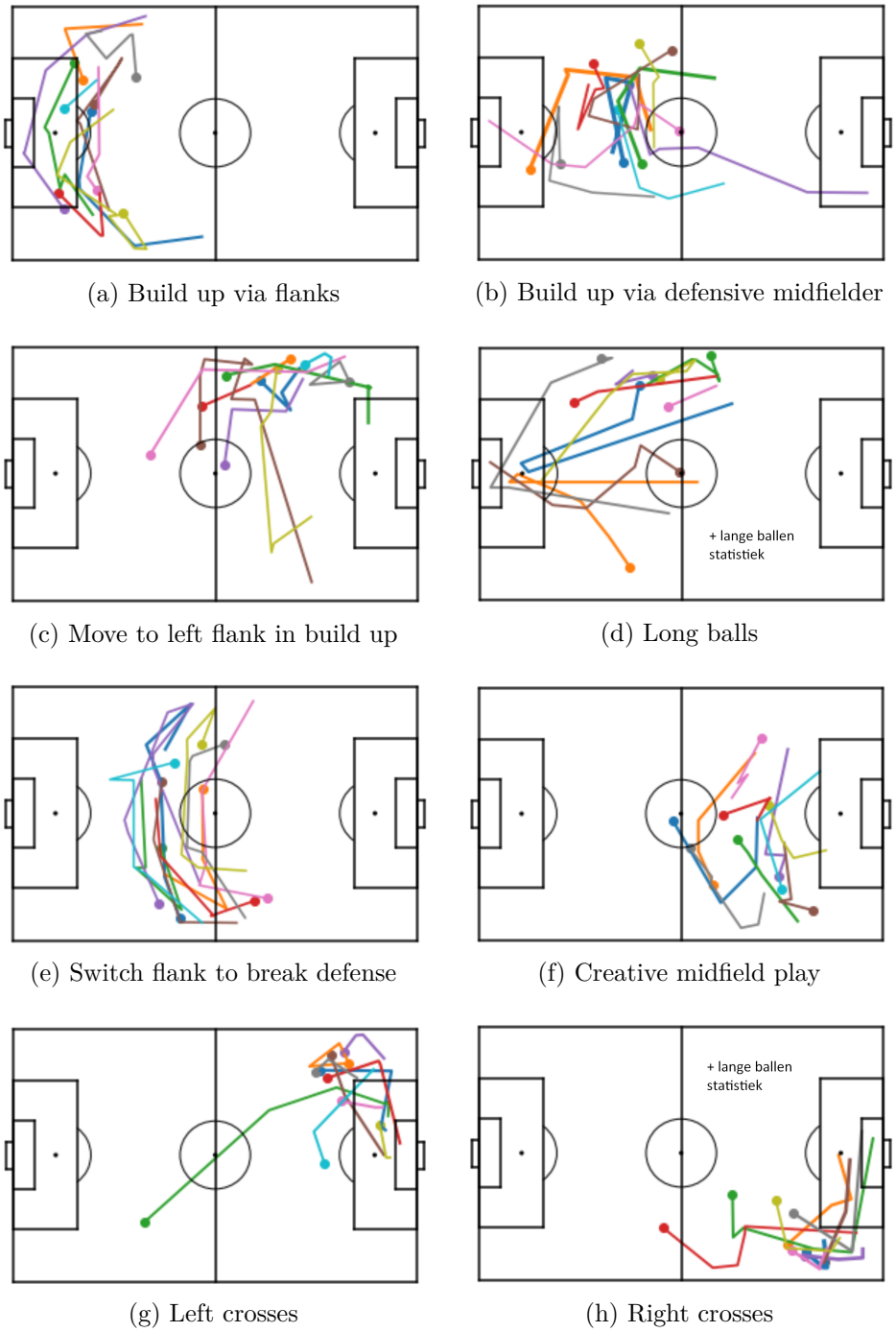
3.6 Identificatie van typische spelpatronen

Het LDA model kan met de opgestelde wedstrijdbeschrijvingen nu getraind worden zodat er typische spelpatronen geïdentificeerd worden. Voor het LDA model wordt er gebruik gemaakt van de implementatie van het Python pakket *gensim* [54]. Een LDA model vereist dat het aantal typische spelpatronen op voorhand gegeven is. Aangezien er slechts 126 *features* zijn en we een makkelijk begrijpbare representatie willen, mag dit aantal niet te hoog liggen. In de bestaande werken die de speelstijl van een team beschrijven aan de hand van een aantal stijlkenmerken ligt dit aantal tussen de acht [3, 51] en twaalf [30]. De aantallen die getest werden zijn bijgevolg die tussen zes en twaalf. In praktijk bleek acht het ideale aantal. Bij grotere aantallen werden er geen nieuwe typische spelpatronen geïdentificeerd maar enkel opgesplitste typische spelpatronen die minder informatief waren. Bij kleinere aantallen werden sommige typische spelpatronen dan weer samengevoegd tot een patroon.

Het testen van een configuratie van het LDA model is niet eenvoudig omdat het model niet deterministisch getraind wordt. Elke configuratie moet een aantal keer getraind worden waarbij de geïdentificeerde typische spelpatronen steeds grondig geänalyseerd moeten worden. Aangezien de spelpatronen vooral begrijpbaar en logisch moeten zijn, is dit een manueel proces dat veel tijd vereist. Figuur 3.11 illustreert de gevonden typische spelpatronen die uiteindelijk werden gekozen. Enkel de tien belangrijkste *features* van elk typisch spelpatroon worden weergeven. De configuratie van het uiteindelijke model gebruikt de 1NN matching manier.

3.7 Uitbreiding van speelstijlvector

De typische spelpatronen die geïdentificeerd werden, stellen een aantal logische en begrijpbare opbouw- en aanvalspatronen voor. Een speelstijl kenmerk dat echter wel verwacht maar niet geïdentificeerd werd, is degene die een counter attack voorstelt. Een poging om dit op te lossen, was om een statistiek gebaseerd op de ‘Counter Attack’ stijl van Stats Perform [3] toe te voegen als typisch spelpatroon. Elke keer dat een team de bal herovert in de eigen helft en in dezelfde balbezit fase een locatie 50 meter verder op het veld bereikt, wordt deze fase als een transitie beschouwd. De



(a) Build up via flanks

(b) Build up via defensive midfielder

(c) Move to left flank in build up

(d) Long balls

(e) Switch flank to break defense

(f) Creative midfield play

(g) Left crosses

(h) Right crosses

Figuur 3.11: Een visualisatie van de met LDA geïdentificeerde typische spelpatronen. Bij elk typisch spelpatroon is een label toegevoegd die een goede beschrijving van het patroon geeft. Er wordt van links naar rechts gespeeld. Een bolletje stelt het begin van een bewegingsketting voor. Hoe dikker de lijn van een bewegingsketting, hoe belangrijker deze is in een typisch spelpatroon.

counter attack waarde wordt dan berekend als de gemiddelde tijd van een transitie in een wedstrijd.

Daarnaast zijn de geïdentificeerde patronen enkel gericht op het beschrijven van de offensieve strategie van een team. De defensieve strategie wordt bepaald door de organisatie van een team en de manier waarop men druk zet. Daarom werd er een statistiek toegevoegd die meet hoe hoog op het veld druk wordt gezet, gebaseerd op de 'High Press' stijl van Stats Perform [3]. Telkens wanneer een team de bal herovert vanaf de 47^{ste} meter van de eigen goal bekeken, wordt een waarde tussen nul en één toegekend die toeneemt naarmate deze balherovering verder op het veld plaatsvindt.

De toevoeging van deze kenmerken aan de speelstijlvector levert slechts een kleine verbetering in de validatie test in sectie 4.2 op. In het volgende hoofdstuk worden deze twee kenmerken dan ook niet in beschouwing genomen zodat de focus van de evaluatie volledig op het LDA model ligt. Het is echter wel nuttig om deze mogelijke uitbreiding in gedachten te houden aangezien het wel meer inzicht kan geven in de speelstijl van een team.

Een reden waarom deze kenmerken geen significante verbetering in resultaten opleveren is dat het aantal balheroveringen in een wedstrijd beperkt is. Dit is eigenlijk de enige informatie over het defensief gedrag van een team dat waar te nemen is in de standaard *event stream data*. *Tracking data* dat het gedrag van het team dat niet in balbezit is ook beschrijft, zou veel extra informatie opleveren. De manier van druk zetten valt echter ook deels af te leiden uit StatsBomb *event data* die *pressure events* bevat [63]. In de Opta data die in deze thesis gebruikt wordt, zijn deze *pressure events* niet beschikbaar.

3.8 Besluit

In dit hoofdstuk werd een nieuwe methode voorgesteld die uit *event data* een begrijpbare vector representatie kan afleiden die inzicht geeft in de speelstijl van een team. Er werden op automatische wijze met LDA acht typische spelpatronen geïdentificeerd die begrijpbaar en logisch zijn en waarmee een vector kan worden opgesteld. Deze vectoren kunnen onderling vergeleken worden met behulp van een afstandsmaat.

Bij de zelf opgestelde *features* die gebruikt worden om een wedstrijd te beschrijven, werd de locatie component van een actie op een *data-driven* manier behandeld. In de *features* werd ook belangrijke informatie voor het bepalen van de speelstijl van een team zoals de opeenvolgingen van acties gecodeerd. Het is ook mogelijk om de *features* te visualiseren wat het eenvoudiger maakt om de typische spelpatronen die worden bepaald door een aantal *features* te interpreteren. Daarnaast viel het ook op hoeveel invloed een *feature* heeft op de typische spelpatronen die geïdentificeerd worden. Coaches en voetbalanalisten die deze typische spelpatronen naar een eigen voorkeur willen sturen, kunnen dit dus best doen door bepaalde *features* wel of niet in beschouwing te nemen.

Hoofdstuk 4

Evaluatie en toepassingen

Dit hoofdstuk bespreekt de toepassingen van het LDA model dat in het vorige hoofdstuk werd getraind. Daarnaast wordt het model geëvalueerd aan de hand van een aantal testen en use cases.

4.1 Visualisatie speelstijlvector

Het getrainde LDA model kan *event stream data* transformeren naar een vector representatie met de geïdentificeerde typische spelpatronen als componenten. Elk van deze componenten stelt de frequentie van een typisch spelpatroon voor en de volledige vector beschrijft dus welke spelpatronen een team het vaakst gebruikt. Een radardiagram is een handige voorstelling die toelaat om deze componenten te visualiseren. Hiervoor kan men zowel relatieve frequenties als absolute frequenties weergeven. Met relatieve frequenties is het makkelijk om teams onderling te vergelijken (figuur 4.1a). Figuur 4.1a toont de visualisaties van de speelstijlvectoren van Paris Saint-Germain, Barcelona en Burnley. Als de vectoren gebruikt dienen te worden om de absolute frequentie van een typisch spelpatroon te weergeven, kan de niet-genormaliseerde vector gebruikt worden (figuur 4.1b). De speelstijlvectoren van een team worden berekend als het gemiddelde van de speelstijlvectoren van elke wedstrijd van dat team in een seizoen.

4.2 Validatie

4.2.1 Vergelijking van speelstijl in verschillende periodes

De methode die gebruikt wordt voor het opstellen van speelstijlvectoren heeft veel parameters: de matching manier, het aantal dichtstbijzijnde *features* bij de matching, de *features* die in beschouwing worden genomen, de geïdentificeerde typische spelpatronen en de afstandsmaat waarmee speelstijlvectoren vergeleken kunnen worden. Normaal gezien is er geen manier om deze parameters op een objectieve manier te evalueren aangezien de speelstijl van een team een subjectief concept is waar geen *ground truth* van bestaat. Decroos et al. [21, 23] ontwikkelden een methode voor

(a) De genormaliseerde speelstijlvector

(b) De niet-genormaliseerde speelstijlvector

Figuur 4.1: De visualisaties van de genormaliseerde (a) en de niet-genormaliseerde (b) speelstijlvectoren van Barcelona (blauw), Paris Saint-Germain (rood) en Burnley (groen).

het evalueren van speelstijlvectoren van spelers. Een belangrijke assumptie bij het gebruiken van deze methode is dat er vanuit gegaan wordt dat de speelstijlen van spelers genoeg verschillen dat spelers aan de hand van hun speelstijl kunnen gedifferentiëerd worden. Aangezien teams uit verschillende spelers bestaan en beïnvloed worden door verschillende trainers met elk hun eigen filosofie wordt verondersteld dat deze assumptie ook voor teams van toepassing is.

De methode kan beschreven worden als volgt. Er worden eerst twee verzamelingen van opeenvolgende wedstrijden opgesteld waarvan één verzameling de training set en de andere de test set voorstelt. Beide verzamelingen bevatten evenveel wedstrijden van elk team. Voor elke verzameling kunnen vervolgens speelstijlvectoren voor elk team worden opgesteld. De speelstijlvectoren van de test set worden dan vergeleken met elke speelstijlvector van de training set met behulp van een afstandsmaat. Er wordt voor elke speelstijlvector van de test set een rangschikking opgesteld met de meest vergelijkbare speelstijlvectoren van de training set. De positie van de speelstijlvector van het overeenkomstige team in de rangschikking bepaalt dan de kwaliteit van een model. Als de meeste teams dus bovenaan in hun eigen rangschikking verschijnen, kan er besloten worden dat de opgestelde vectoren de speelstijl van teams goed karakteriseren. Een handige statistiek om de positie van elk team in beschouwing te nemen is de Mean Reciprocal Rank (MRR). Deze wordt gedefeniëerd als het gemiddelde van elke inverse rangschikking: $MRR = \frac{1}{T}\sum_{i=0}^{T}\frac{1}{R_i}$ met T het aantal teams.

Decroos et al. [21, 23] gebruiken als verzamelingen twee opeenvolgende seizoenen. De training set stelt dan bijvoorbeeld de verzameling van wedstrijden in het seizoen 2018/19 voor en de test set die van het seizoen 2019/20. Deze verzamelingen bleken voor spelers een goede evaluatie tool. Het is dan ook aannemelijk dat de speelstijl van een speler over twee seizoen niet enorm veel veranderd. Bij teams verandert er tussen twee seizoenen wel enorm veel. Slechts 39 van de 84 teams hadden tijdens het seizoen 2018/19 dezelfde trainer als tijdens het seizoen 2019/20. Daarnaast gebeuren er elke zomer een aantal transfers van belangrijke spelers. Aangezien de speelstijl van een team enorm afhangt van het gedrag van zijn spelers heeft het verdwijnen en bijkomen van een aantal belangrijke spelers waarschijnlijk een grote impact. De evaluatie methode met de seizoenen 2018/19 en 2019/20 als training en test set bevestigt dit vermoeden. Slechts 7.1% van de teams staan op de eerste plaats in hun eigen rangschikking, wat bijzonder weinig is (tabel 4.1a).

De speelstijl van een team binnen een seizoen is daarentegen wel meer consistent. De drukke opeenvolging van wedstrijden leent zich er niet toe om nieuwe tactische patronen in het team te slepen. Transfers van belangrijke spelers vinden er in het midden van het seizoen meestal niet plaats, die gebeuren eerder in de zomer. De vergelijking van de speelstijlvectoren met als training set de eerste helft van het seizoen en als test set de tweede helft bevestigen dit vermoeden (tabel 4.1b en figuur 4.2). De teams die uit de top 10 van hun eigen rangschikking vallen, zijn AFC Bournemouth, Everton, Wolverhampton Wanderers, Marseille, Amiens, Metz, Toulouse, Torino, Hoffenheim, Freiburg en Real Betis. De meeste van deze teams eindigden onderaan in het klassement en Everton wisselde dan weer in het midden van het seizoen van trainer. Het seizoen van de Ligue 1 teams Marseille, Amiens,

Tabel 4.1: De evaluatie tabellen van de vergelijking tussen de speelstijlvectoren in verschillende periodes.

(a) Vergelijking van de speelstijlvectoren van het seizoen 2018/19 met die van 2019/20.

	MRR	Top 1	Top 2	Top 3	Top 5	Top 10
Alle teams	0.163	7.1%	11.9%	15.5%	22.6%	34.5%
Zelfde coach	0.172	7.7%	12.8%	15.4%	23.1%	41.0%

(b) Vergelijking van de speelstijlvectoren van de eerste helft van het seizoen 2019/20 met die van de tweede helft.

	MRR	Top 1	Top 2	Top 3	Top 5	Top 10
1NN	0.548	37.8%	56.1%	66.3%	74.5%	88.8%
2NN (vgl. 3.1)	0.620	47.9%	61.2%	70.4%	81.6%	87.7%
2NN (vgl. 3.2)	0.583	44.9%	55.1%	66.3%	75.5%	88.8%
3NN (vgl. 3.1)	0.607	44.8%	63.3%	70.4%	79.6%	89.8%
3NN (vgl. 3.3)	0.580	43.9%	58.2%	66.3%	73.5%	87.7%

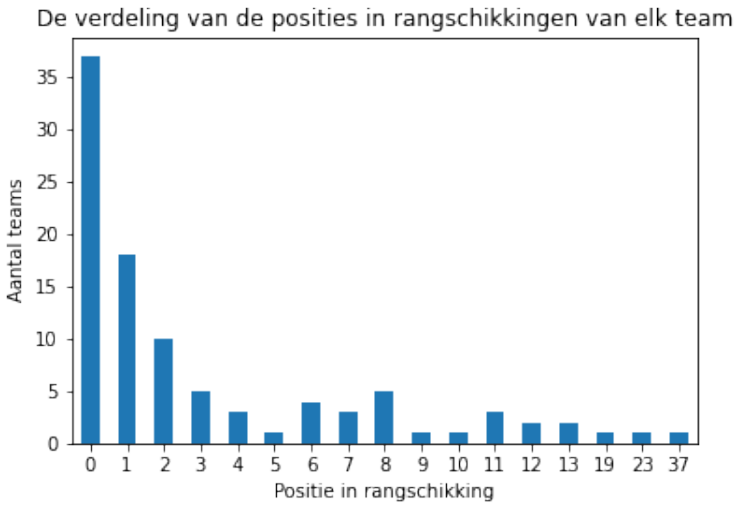
Metz en Toulouse werd vanwege de Covid-pandemie na 27 wedstrijden beëindigd waardoor er maar 13 wedstrijden per seizoenshelft in beschouwing werden genomen.

Daarnaast kan deze evaluatie methode ook gebruikt worden om de afstandsmaat te bepalen voor het identificeren van vergelijkbare teams. Drie veelgebruikte afstandsmaten voor het vergelijken van vectoren zijn de manhattan, de euclidische en de mahalanobis afstand [18]. De afstandsmaat die veruit de beste rangschikkingen opleverde is de manhattan afstand. Deze afstand werkt goed omdat elke waarde in een speelstijlvector betekenisvol is. De manhattan afstand wordt dan ook berekend als de som van de absolute verschillen in elke component. Grote verschillen worden niet afgestraft zoals bij de euclidische afstand wel het geval is [21].

Om de verschillende parameters van het model te bepalen, werden steeds de speelstijlvectoren van de eerste helft van het seizoen 2019/20 vergeleken met die van de tweede helft met de manhattan afstand. De 1NN configuratie scoort iets lager dan de andere configuraties op de validatie test (tabel 4.1b). De typische spelpatronen die met de 1NN configuratie werden gevonden, zijn wel het meest begrijpbaar en logisch (figuur 3.1). Appendix A bevat de typische spelpatronen die met de andere configuraties werden gevonden. Omdat de verschillen in waardes beperkt zijn, werd er voor de 1NN configuratie gekozen.

4.2.2 Visualisatie van afstand tussen typische spelpatronen

Een andere manier om de geïdentificeerde typische spelpatronen te evalueren, is door ze te visualiseren met pyLDAvis [42, 61]. Deze visualisatie (figuur 4.3) bestaat uit twee componenten: een *intertopic distance map* en een staafdiagram. De *intertopic distance map* is een visualisatie van de typische spelpatronen in een tweedimensionale ruimte.

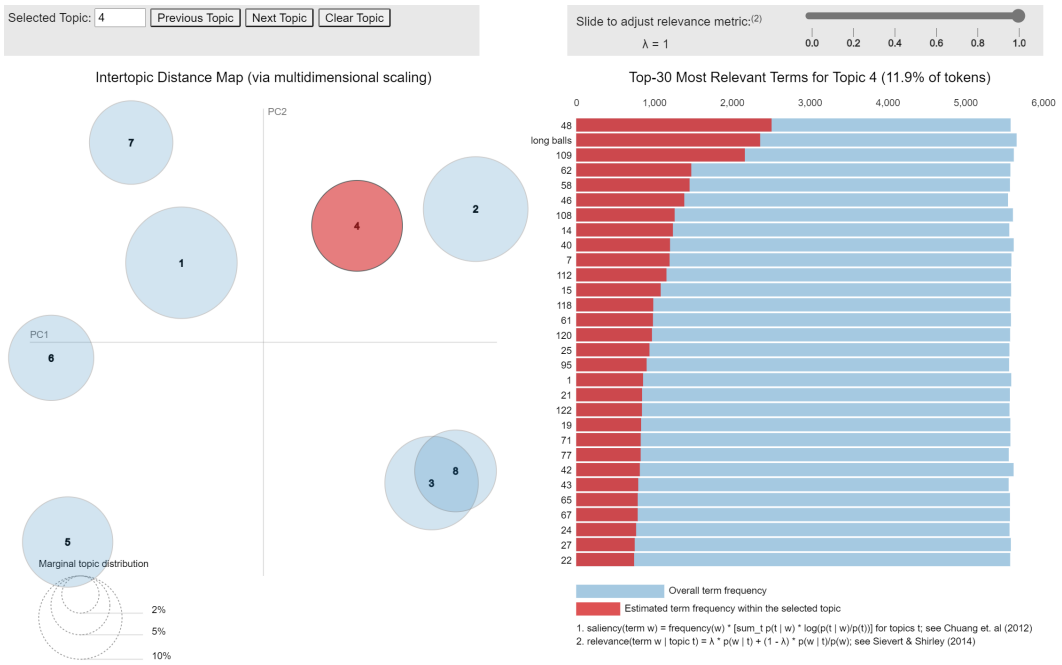


Figuur 4.2: De verdeling van de rangschikkingen bij het vergelijken van de speelstijlvectoren van de eerste helft van het seizoen 2019/20 met die van de tweede helft. De bijhorende MRR waarde is 0,548.

De twee dimensies stellen de *principal components* voor van een dimensionaliteitsreductie met PCA, gebaseerd op de *features* die een typisch spelpatroon omvat. Hoe dichter twee cirkels dus staan, hoe meer *features* ze gemeenschappelijk hebben. Hoe groter de oppervlakte van de cirkels, hoe meer *features* ze omvatten.

De typische spelpatronen van het finale LDA model liggen op deze kaart ver uiteen wat betekent dat ze verschillende delen van de *feature* ruimte beschrijven (figuur 4.3). Enkel het derde en het achtste typische spelpatroon overlappen. Het derde spelpatroon stelt een opbouw langs de flanken voor (figuur 3.11a). Het achtste spelpatroon stelt voorzetten langs de linkerflank voor (figuur 3.11g). Dit zijn allebei spelpatronen die typerend zijn voor een flank gerichte speelwijze dus is het niet onlogisch dat ze een aantal *features* gemeenschappelijk hebben. Daarnaast omvat elk typisch spelpatroon iets meer dan 10% van de *features* wat af te lezen valt aan de oppervlaktes van de cirkels. Aangezien de typische spelpatronen (behalve drie en acht) niet overlappen beschrijven ze samen dus een groot deel van de *feature* ruimte.

Het staafdiagram rechts in figuur 4.3 geeft de meest relevante *features* voor het vierde typisch spelpatroon weer die vooral geassociëerd worden met het spelen van lange ballen. De relevantie van een *feature* voor een typisch spelpatroon is de waarschijnlijkheid dat deze voorkomt in een typisch spelpatroon gewogen met de waarschijnlijkheid dat de *feature* in het algemeen voorkomt. Omdat de frequenties van alle *features* ongeveer gelijk zijn, komen de relevante *features* ongeveer overeen met de meest waarschijnlijke *features* voor een typisch spelpatroon die in figuur 3.11 werden getoond.



Figuur 4.3: Visualisatie van de afstanden tussen de verschillende typische spelpatronen. Rechts worden de meest relevante termen van het vierde typisch spelpatroon 'Long balls' getoond.

4.3 Meest vergelijkbare teams op vlak van speelstijl

In de voorbereiding van een seizoen of een toernooi spelen teams meestal een aantal oefenmatchen om hun systeem op punt te zetten. Het liefst spelen ze tegen tegenstanders die een gelijkaardige speelstijl hebben als de teams die ze treffen in het komende toernooi of seizoen. Aan de hand van de speelstijlvectoren van elk team kunnen een aantal vergelijkbare tegenstanders worden geïdentificeerd. In de vorige sectie werd het duidelijk dat de speelstijlvectoren van teams het beste kunnen vergeleken worden met de manhattan afstand. In tabel 4.2 worden de teams met de meest gelijkaardige speelstijl aan die van Manchester City en RB Leipzig getoond. Opvallend is dat Sassuolo op de tweede plaats staat in de rangschikking van Manchester City tussen allemaal top teams.

Een andere handige manier om teams met gelijkaardige speelstijlen te identificeren is door de speelstijlvectoren te clusteren met *k-means*. Er werden zeven groepen van teams geïdentificeerd die een gelijkaardige speelstijl hebben maar dit aantal kon ook meer of minder zijn geweest [46]. Met behulp van PCA kan deze clustering gevisualiseerd worden in een twee-dimensionale ruimte (figuur 4.4).

Veel van de top teams bevinden zich links bovenaan in figuur 4.4. Een buitenbeentje dat zich tussen deze top teams bevindt, is Sassuolo. Het team onder leiding van Roberto Di Zerbi speelt dominant voetbal met een op balbezit gerichte speelwijze. Met veel beweging en korte passes richt zijn team zich op het opbouwen van achteruit

Tabel 4.2: De tien teams met de meest vergelijkbare speelstijl als Manchester City en RB Leipzig. De gebruikte afstandsmaat is de manhattan afstand.

(a) Manchester City

Team	Afstand
Paris Saint-Germain	0.2868
Sassuolo	0.3578
Liverpool	0.3876
Lazio	0.3880
AS Roma	0.4312
Barcelona	0.4875
AC Milan	0.4923
Manchester United	0.5269
Napoli	0.5437
Chelsea	0.5608

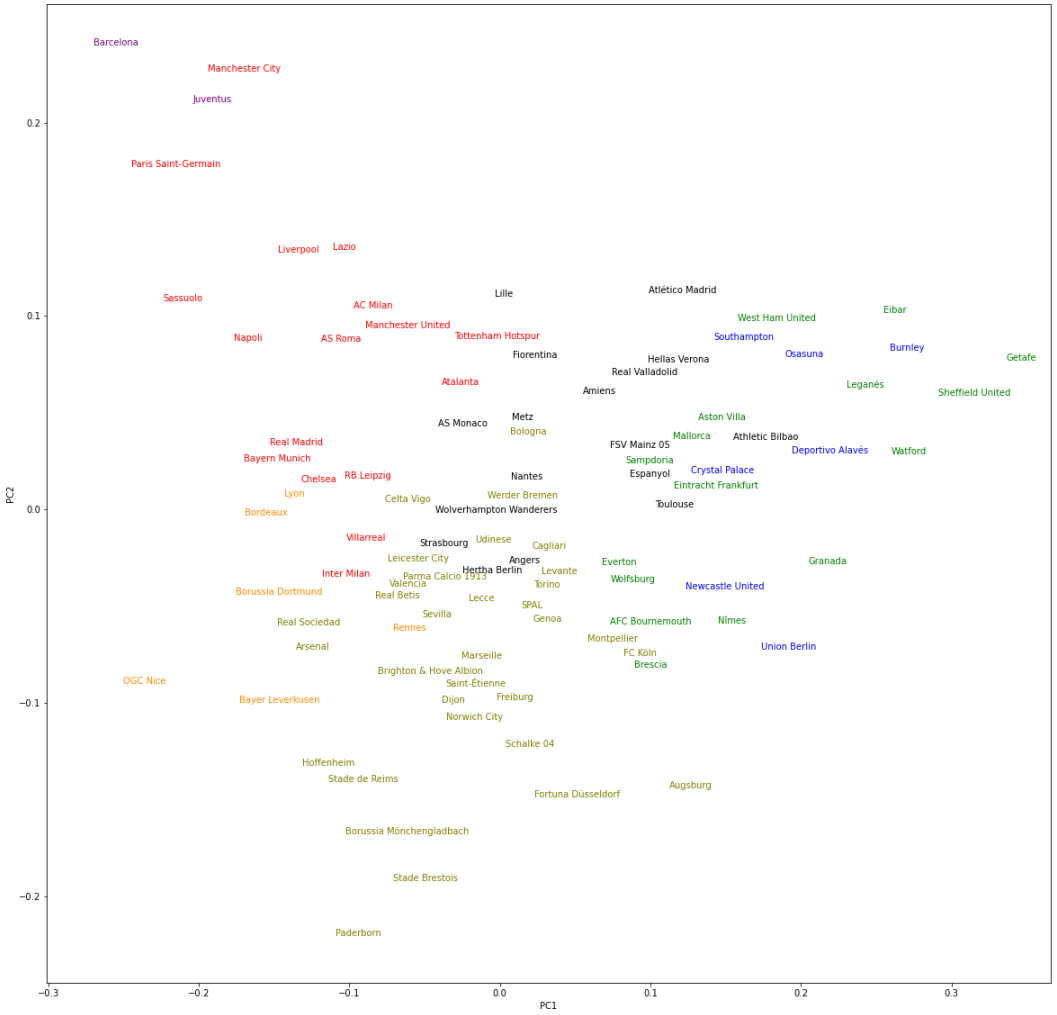
(b) RB Leipzig

Team	Afstand
Leicester City	0.2101
Inter Milan	0.2289
Chelsea	0.2511
B. & Hove Albion	0.2525
Tottenham Hotspur	0.2554
Celta Vigo	0.2746
AC Milan	0.2759
Lecce	0.2771
Udinese	0.2797
Real Sociedad	0.2880

via een sterk bezet centraal middenveld. Centraal is Manuele Locatelli één van de sleutelfiguren in de opbouw om de bal in het centrum van de *middle third* te krijgen. Dit valt af te lezen uit het hoge aandeel van het typisch spelpatroon 'Build up via defensive midfielder' in de speelstijlvector van Sassuolo (figuur 4.5). Op de flanken zakken de vleugelspelers laag terug op het veld om te helpen met de opbouw zodat de breedte van het veld volledig wordt benut. De flanken worden enkel gebruikt in de opbouw want hierna focussen ze hun acties op de centrale zones. Het gebeurt zelden dat ze de bal voor het doel proberen te krijgen door middel van een voorzet [60]. Deze aspecten zijn dan weer duidelijk terug te vinden in het hoge aandeel van 'Build up via flanks' en 'Creative Midfield Play' en het lage aandeel van 'Left Crosses' en 'Right Crosses'.

Een andere uitschieter is Paderborn die op het einde van het seizoen 2019/20 degradeerde. In tegenstelling tot de meeste andere teams onderaan het klassement kozen ze er niet voor om veel lange ballen te spelen maar om op te bouwen van achteruit. De meeste tegenstanders zetten met drie spelers druk op hun viermans-defensie waardoor er één vrije man is. Meestal is dit één van de vleugelverdedigers waardoor ze hun aanvallen dan ook vaak opbouwen via de flanken. Dit valt duidelijk af te lezen uit het hoge aandeel van 'Build up via flanks' in de speelstijlvector van Paderborn (figuur 4.5). In de *final third* proberen ze niet enkel met voorzetten voor het doel te geraken maar ook via dribbels. Na Bayern München verrichten ze dan ook het meeste 1v1 dribbels per 90 minuten. Aangezien ze niet de beste spelers hadden, leidde deze tactiek echter niet tot veel doelkansen wat dan ook resulteerde in weinig balbezit in de *final third*. Deze onsuccesvolle tactiek verklaart dan ook het hoge aandeel van 'Build up via flanks' en het lage aandeel van 'Right Crosses', 'Left Crosses' en 'Creative midfield play' in de speelstijlvector [10].

Getafe helemaal rechts in figuur 4.4 is dan weer een team dat zo snel mogelijk de goal probeert te bereiken. Dit doen ze met behulp van snelle flank middenvelders

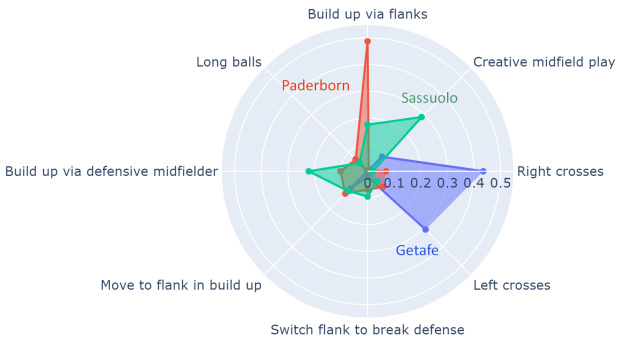


Figuur 4.4: Visualisatie van de speelstijlvectoren van elk team

die vaak voorzetten geven naar één van de aanvallers in de *box* waar ze minstens met drie proberen te zijn [15]. Dit resulteert dan ook in een hoog aandeel van 'Left Crosses' en 'Right Crosses' in de speelstijlvector van Getafe (figuur 4.5).

4.4 Uniekheid en consistentie van speelstijl

Naast het identificeren van de meest vergelijkbare teams op vlak speelstijl is het interessant om te kwantificeren hoe uniek en consistent de speelstijl van een team is. Gyarmati en Hefeeda [34] definiëerden daarom twee interessante statistieken: uniekheid en consistentie.



Figuur 4.5: De gevisualiseerde speelstijlvectoren van Paderborn (rood), Sassuolo (groen) en Getafe (blauw).

4.4.1 Uniekheid

De uniekheid geeft weer hoe moeilijk het is om een team te vinden met een vergelijkbare speelstijlvector. Om deze te berekenen wordt gekeken naar de M meest vergelijkbare speelstijlvectoren van een team. De uniekheid van de speelstijl van team i wordt gedefiniëerd als:

$$U_i = \sum_{j=0}^{M} d_{ij}$$

waarbij d_{ij} de afstand tussen de speelstijlvector van team i en team j voorstelt. Hoe hoger deze waarde, hoe unieker de speelstijl van een team. Met $M = 5$ valt het op dat de teams met de meest unieke speelstijl veelal topteams zijn terwijl de teams met de minst unieke speelstijl vooral zwakkere teams zijn (figuur 4.6).

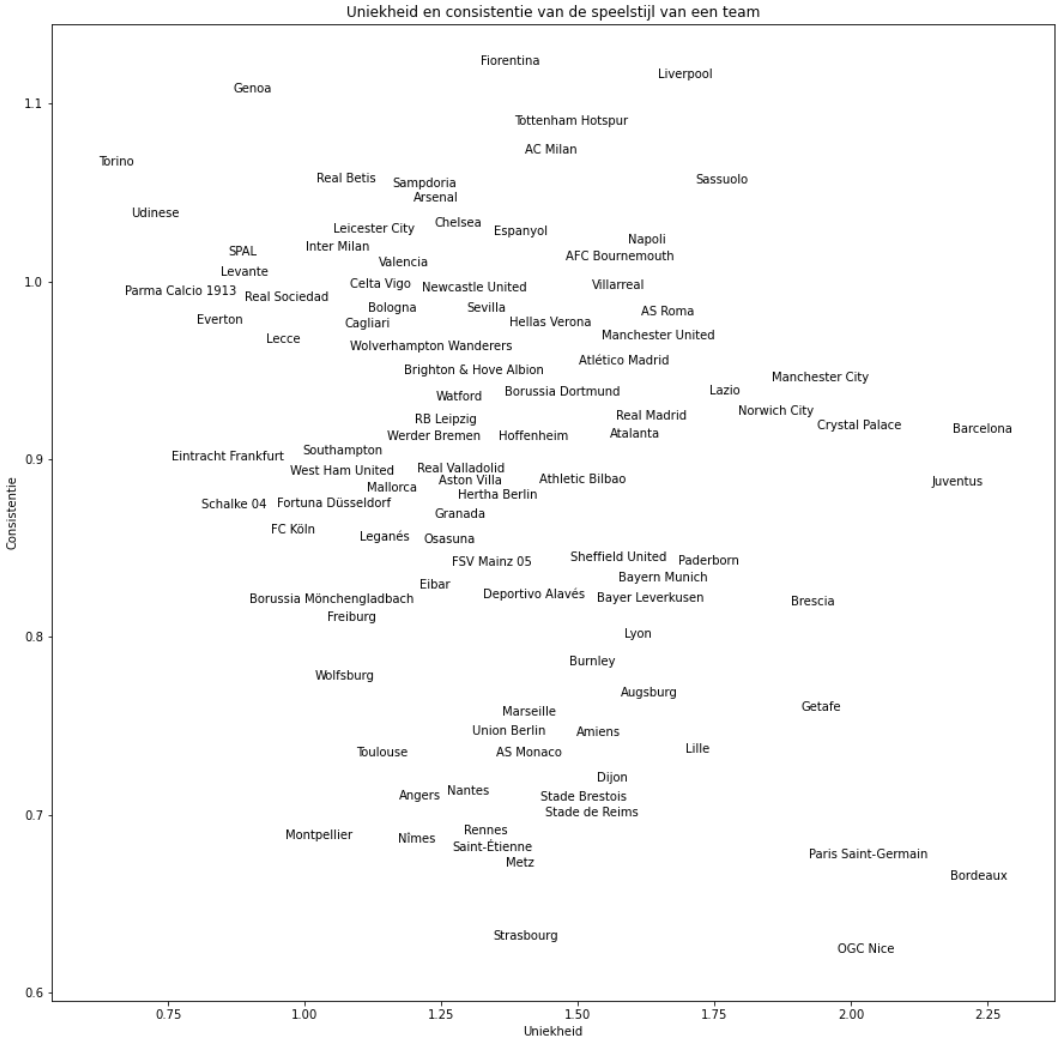
4.4.2 Consistentie

De consistentie van team i voor wedstrijd k wordt gedefiniëerd als:

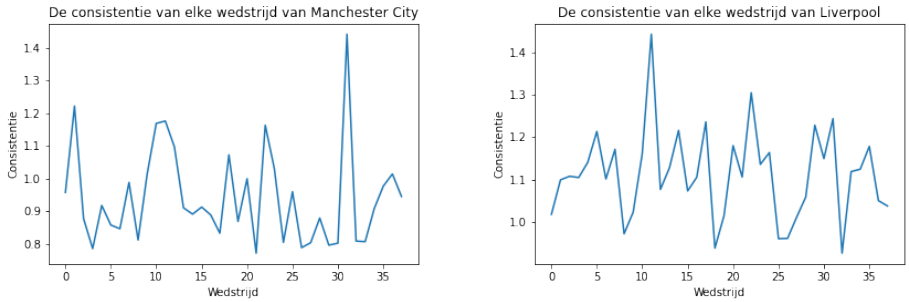
$$C_i^k = \frac{1}{N} \sum_{t=1}^{N} D(v_i^k, v_i^t)$$

met $D(v_i^k, v_i^t)$ de afstand tussen de speelstijlvectoren van team i voor wedstrijden k en t, en N het aantal wedstrijden van team i in een seizoen. Hoe lager deze waarde, hoe gelijkaardiger de speelstijl in die wedstrijd is aan de speelstijlen van de andere wedstrijden van dat team in een seizoen. In figuur 4.6 wordt de gemiddelde consistentie waarde van elk team in een seizoen getoond.

Als we naar de consistentie waarde van elke wedstrijd kijken van Manchester City en Liverpool steken er twee wedstrijden bovenuit (figuur 4.7). Bij Manchester City is dit de wedstrijd op speeldag 32 wanneer het Liverpool als tegenstander thuis ontving. Liverpool ontving thuis dan weer Manchester City op speeldag 12. Deze teams waren de twee belangrijkste titelkandidaten in 2019/20 en staan bekend om hun dominant voetbal en counterpressing. Het is dan ook niet onlogisch dat ze



Figuur 4.6: De uniekheid en consistentie van de speelstijl van elk team in het seizoen 2019/20.



(a) Manchester City (b) Liverpool

Figuur 4.7: De consistentie waardes van elke wedstrijd in het seizoen 2019/20 voor Manchester City en Liverpool.

hun tactiek aanpassen aan elkaar zodat de ander niet zijn bekend spel kan spelen. Wanneer twee teams het balbezit opeisen, resulteert dit ook vaak in een wedstrijd met veel eindeloos rond getik zonder echt tot doelkansen te komen wat de aparte speelstijl in deze twee wedstrijden kan verklaren. Daarnaast valt het ook op dat de speelstijl van Manchester City consistenter is doorheen het seizoen dan die van Liverpool.

4.5 Use cases

4.5.1 Sarriball

Één van de opvallendste figuren in het Europees voetbal de afgelopen jaren is Maurizio Sarri. De voormalige bankier werd door velen voor het eerst opgemerkt toen hij met Napoli open, aanvallend en snel voetbal bracht dat door de media tot Sarriball werd gedoopt. De sleutelspeler in Sarriball is de verdedigende middenvelder die naast het afstoppen van aanvallers ook in staat moet zijn om de middenvelders en aanvallers van aanvoer te voorzien. Kenmerkend aan Sarriball is ook het passen in één tijd, het bewegen van het team in blok en een eigen invulling van *Gegenpressing*. Sarri is geen fan van het opbouwen via de flanken en dwingt zijn team dan ook om door het centrum op te bouwen. Tegen tegenstanders die de bus parkeren is dit echter al vaak een groot probleem gebleken. Het weinige gebruik van de flanken in de opbouw maakt zijn teams dan ook voorspelbaarder en dus makkelijker af te stoppen [2]. De invloed van Sarri is ook duidelijk te zien in de speelstijlvectoren van de twee laatste teams die hij onder zijn hoede had (figuur 4.8). Na het vertrek van Sarri evolueerde de speelstijl van Chelsea duidelijk meer richting het gebruik van de flanken in de opbouw. Sinds de komst van Sarri bij Juventus werd hun speelstijlvector meer geconcentreerd op 'Creative midfield play'. Dit wijst er op dat er minder gevariëerd wordt in de opbouw van een aanval en in de meerderheid van de gevallen door het centrum wordt gespeeld. De speelstijlvectoren van Juventus en Chelsea onder Sarri zijn heel gelijkaardig met een duidelijke uitschieter als 'Creative midfield play' terwijl

(a) Chelsea en Juventus onder Sarri

(b) Chelsea onder Lampard en Juventus onder Allegri

Figuur 4.8: Vergelijking van de speelstijlvectoren van Chelsea en Juventus onder Sarri

de speelstijlvectoren onder Lampard en Allegri redelijk veel verschillen.

4.5.2 Inter Milaan: voor en na Antonio Conte

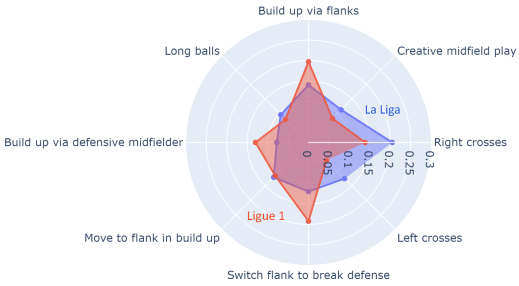
Dit seizoen (2021/21) doorbrak Inter Milaan de hegemonie van Juventus door het Italiaanse kampioenschap te winnen. Één van de sleutelfiguren van dit succes is trainer Antonio Conte die eind mei 2019 het roer overnam van Luciano Spalletti. Met zijn bekende systeem transformeerde hij Inter Milaan in een machine die 74% van zijn wedstrijden kon winnen dit seizoen. Een bekend kenmerk van zijn systeem is het veelvuldig bezetten van de flanken met onder andere aanvallende vleugelverdedigers [24]. De spelers op deze posities, Kwadwo Asamoah en Ashley Young, zijn eigenlijk vleugelmiddenvelders die ook kunnen verdedigen. In de speelstijlvectoren is dan ook een duidelijke verschuiving te zien van 'Creative midfield play' en 'Build up via defensive midfield' naar 'Build up via flanks' en 'Move to flank in build up' (figuur 4.9). Inter Milaan staat sinds Antonio Conte ook bekend om zijn snelle omschakeling met speerpunten Lukaku en Martinez wat deels het verminderde aandeel van 'Creative midfield play' kan verklaren [24].

4.5.3 Competities

Naast de speelstijl van een team kan ook de speelstijl van een competitie worden beschreven als de gemiddelde speelstijlvector van alle wedstrijden in die competitie. De wedstrijden in de Serie A hebben het grootste aandeel van 'Creative midfield play' en het laagste aandeel van 'Long balls'. Heel wat teams in de Serie A hadden dat seizoen dan ook een trainer die bekend staat om zijn modern en aanvallend voetbal zoals Sarri (Juventus), Conte (Inter Milaan), Fonesca (AS Roma), Gampioli (AC Milaan), Di Zerbi (Sassuolo) en Gasperini (Atalanta) [64, 32]. Daarnaast valt het op dat de wedstrijden in de Bundesliga en de Ligue 1 een laag aandeel van 'Creative midfield play' hebben maar een hoog aandeel van 'Build up via flanks' en 'Build up via defensive midfielder' en 'Switch flank to break defense'. Dit kan er op wijzen dat er meer wedstrijden plaatsvonden waarbij één van de twee teams de

Figuur 4.9: De speelstijlvectoren van Inter Milan in het seizoen 2018/19 en in het seizoen 2019/20. In 2018/19 was Luciano Spalletti coach en in 2019/20 Antonio Conte.

(a) Bundesliga (blauw), Serie A (rood), Premier League (groen)



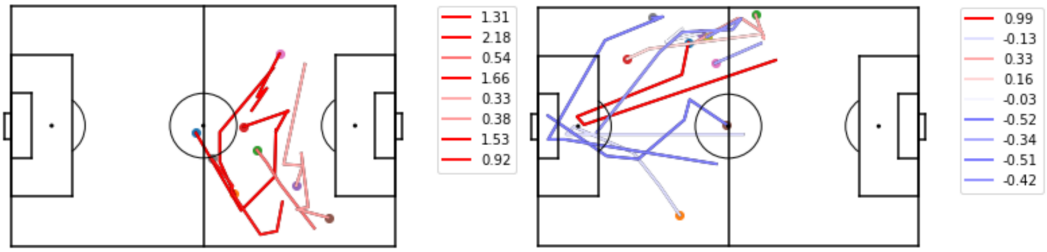
(b) La Liga (blauw) en Ligue 1 (rood)

Figuur 4.10: De speelstijlvectoren van alle wedstrijden in een competitie in het seizoen 2019/20

bus parkeerde waardoor het andere team verplicht werd om geduldig de bal rond te spelen om toch een opening in de defensie te vinden. De teams in de Ligue 1 en de Bundesliga kwamen ook het minste tot het nemen van voorzetten. Dit kan een bewuste keuze zijn of simpelweg omdat ze niet genoeg mogelijkheden tot een voorzet konden creëren.

4.6 Analyse van *features* in typische spelpatronen

De *features* die in sectie 3.4 werden opgesteld, werden in sectie 4.6 gegroepeerd tot typische spelpatronen waarmee de speelstijl van een team kan worden beschreven. Het identificeren van typische spelpatronen was nodig om een vector te bekomen die snel een beeld geeft van de speelstijl van een team zodat ze ook makkelijk vergeleken kunnen worden. Deze speelstijlvector op hoog niveau is eenvoudiger te begrijpen dan een vector bestaande uit de oorspronkelijk opgestelde *features*. Wanneer men



(a) Paris Saint-Germain
'Creative midfield play'

(b) Burnley
'Long balls'

Figuur 4.11: Een ingezoomde weergave van een typisch spelpatroon. Een rode kleur betekent dat de *feature* meer dan gemiddeld voorkomt en een blauw kleur minder dan gemiddeld. Hoe donkerder de kleur hoe meer of minder de *feature* voorkomt. Er wordt van links naar rechts gespeeld. De start van een bewegingsketting wordt aangegeven door een punt.

de speelstijl van een team in meer detail wil analyseren, is het echter nuttig om te kijken naar het voorkomen van elke *feature*. Men kan als het ware inzoomen op een component en het voorkomen van elke belangrijke *feature* in een typisch spelpatroon. Op deze manier kunnen de *features* op een gestructureerde wijze geanalyseerd worden.

Om het voorkomen van een *feature* naar waarde te kunnen schatten, wordt de waarde vergeleken met de gemiddelde waarde van die *feature*. Concreet wordt de z-score berekend voor de gemiddelde *feature* waarde van alle wedstrijden van een team. Als voorbeeld worden de *features* van het typisch spelpatroon 'Creative midfield play' beschouwd voor Paris Saint-Germain. Het valt op dat de *features* die een bewegingsketting van de rechterflank naar het midden voorstellen, minder voorkomen dan de rest van de *features* (figuur 4.11a). Niet toevallig spelen supersterren Neymar en Mbappé op de linkerflank terwijl Di Maria de rechterflank voor zijn rekening neemt. Het is dan ook aannemelijk dat de andere spelers van Paris Saint-Germain eerder aanvallen met Neymar en Mbappé proberen op te zetten dan met Di Maria.

Een ander voorbeeld zijn de *features* van Burnley in het typisch spelpatroon 'Long balls'. Het valt op dat de *feature* die het terugspelen vanaf de linkerflank naar de keeper of centrale verdediger die dan een lange bal verstuurt, ≈ 1 standaarddeviatie boven het gemiddelde voorkomt terwijl de rest van de *features* bijna allemaal onder het gemiddelde zitten. Het kan dus nuttig zijn om als tegenstander meer druk te zetten op de linkervleugelverdediger van Burnley om deze lange ballen te vermijden.

4.7 Besluit

In dit hoofdstuk werd het getrainde LDA model geëvalueerd met behulp van een aantal validatie technieken en door een aantal voorbeelden te bespreken. Uit deze evaluatie kunnen we besluiten dat de speelstijlvectoren die met het model afgeleid kunnen worden een goed beeld geven van de speelstijl van een team. Deze speelstijlvectoren kunnen ook in verdere analyses gebruikt worden om de speelstijl van een team in

rekening te brengen. Daarnaast werden er een aantal toepassingen van het model besproken zoals het identificeren van de meest vergelijkbare teams op vlak van speelstijl of een diepere analyse van een typisch spelpatroon voor een team.

Hoofdstuk 5

Conclusies

Het doel van deze masterproef was om op automatische wijze een begrijpbare representatie van de speelstijl van een team af te leiden uit *ball event data*. Met deze representatie moest het ook mogelijk zijn om de speelstijlen van teams onderling te vergelijken. In de eerste sectie wordt de voorgestelde oplossing en de resultaten hiervan besproken. Vervolgens worden er suggesties voor verder onderzoek voorgesteld.

5.1 Samenvatting resultaten

Om een begrijpbare representatie te bekomen die onderling vergeleken kan worden, dienden een aantal typische spelpatronen geïdentificeerd te worden die inzicht geven in de speelstijl van een team. Aan de hand van deze typische spelpatronen kan er voor elk team dan een speelstijlvector worden opgesteld. Belangrijk hierbij was dat de typische spelpatronen en de vector die hiermee opgesteld kan worden begrijpbaar en logisch zijn, wat bij veel bestaande werken niet het geval is. Gebaseerd op het werk van Perdomo Meza en Girela [51, 50] werden er typische spelpatronen geïdentificeerd met behulp van het *Topic Modeling* algoritme Latent Dirichlet Allocation (LDA). Analoog aan het vinden van onderwerpen waarmee documenten beschreven kunnen worden, werden er typische spelpatronen gezocht waarmee de speelstijl van een team beschreven kan worden.

Om dit model te trainen dienden er *features* opgesteld te worden waarmee een wedstrijd beschreven kan worden. Er werd hierbij getracht om zoveel mogelijk relevante informatie uit de *event data* te integreren. Om de offensieve strategie van een team te bepalen zijn de interacties tussen de spelers en de locatie hiervan belangrijke aspecten. Om deze informatie te coderen werd de *event data* opgedeeld in bewegingskettingen die ieder een opeenvolging van gebeurtenissen beschrijven waarbij vier opeenvolgende spelers betrokken zijn [70]. De locatie component werd hierbij op een *data-driven* manier behandeld. Concreet werden de bewegingskettingen gegroepeerd met Dynamic Time Warping (DTW) als afstandsmaat [22]. Voor iedere groep werd er een vertegenwoordiger gekozen. Een *feature* wordt dan voorgesteld als het aantal bewegingskettingen in een wedstrijd die met een vertegenwoordiger matchen. Op deze manier is het mogelijk om de verschillende bewegingen van een

team in een wedstrijd te beschrijven. Er werden nog een aantal *features* toegevoegd die de bewegingen in de *final third* beschrijven en één die het aantal lange ballen in een wedstrijd weergeeft.

Er werden uiteindelijk acht typische spelpatronen geïdentificeerd met LDA met de wedstrijdbeschrijvingen als training data. Dit zijn herkenbare en logische spelpatronen die de opbouw van een aanval beschrijven. De speelstijlvector die aan de hand van deze typische spelpatronen kan worden opgesteld is ook begrijpbaar en logisch. Bij de evaluatie van het model werd er vooral gelet op hoe herkenbaar en begrijpbaar de spelpatronen waren zodat de vectoren inzicht geven in de speelstijl van een team. Met behulp van een aantal *use cases* en evaluatie tools werd aangetoond dat de vector effectief inzicht geeft in de speelstijl van een team. Met een radar diagram visualisatie zijn de speelstijlvectoren makkelijk te begrijpen en te vergelijken. Een coach of analist kan met deze speelstijlvectoren snel een beeld krijgen van de manier waarop een team speelt. Als ze de speelstijl van het team dieper wensen te analyseren, kan er ingezoomd worden in de typische spelpatronen om het individueel voorkomen van een *feature* te onderzoeken. De speelstijlvectoren kunnen ook gebruikt worden om teams met een bepaalde speelstijl te identificeren, wat nuttig kan zijn voor het zoeken van een tegenstander in een oefenmatch.

5.2 Toekomstig werk

Voor zowel het opstellen van *features* als het identificeren van typische spelpatronen zijn er aantal alternatieven en uitbreidingen. De verschillende pistes die gevolgd kunnen worden in verder onderzoek worden hier besproken.

Momenteel worden de *features* opgesteld door met K-medoids bewegingsketten te groeperen met DTW als afstandsmaat en de *medoid* van elke cluster als vertegenwoordiger te kiezen. Een alternatief clustering algoritme dat Decroos et al. [23] gebruikten om hun prototype acties op te stellen is een Gaussian Mixture Model (GMM). Met een aanpassing van dit model zodat het opeenvolgingen van locaties als datapunten accepteert en DTW als afstandsmaat tussen datapunten gebruikt, zouden er een aantal kansverdelingen kunnen worden geïdentificeerd die als *features* kunnen gebruikt worden. Net zoals *K-means* wordt er ook een gemiddelde bewegingsketting berekend bij GMM. Het gemiddelde van een aantal bewegingsketten vinden is echter niet triviaal maar kan bijvoorbeeld met DTW Barycenter Averaging [52] berekend worden.

Indien er gebruik gemaakt zou kunnen worden van *event data* die *pressure events* bevat, zou de defensieve component van de speelstijl van een team beter gemodelleerd kunnen worden. Het aantal intercepties in een wedstrijd is namelijk te beperkt om een aantal typische defensieve gedragingen te modelleren. Daarnaast zou met een toevoeging van *tracking data* ook meer informatie zoals de opstelling van een team in een aanval of tijdens een tegenaanval in rekening kunnen worden gebracht. Een typisch spelpatroon zou dan voorgesteld kunnen worden als een combinatie van een aantal looplijnen die vaak samen plaatsvinden in een aanval in plaats van enkel het traject van de bal te beschouwen [69]. Miller en Bornn [44] ontwikkelden een

gelijkaardige methode om de verschillende looplijnen in basketbal te groeperen.

Hoewel het LDA model in staat is om een aantal typische spelpatronen te identificeren, zijn de assumpties van het model gericht op het verwerken van tekst documenten. Wang et al. [68] toonden aan dat het model aangepast kan worden om tactische patronen van een team te identificeren waarbij de identiteit van de betrokken spelers bij een pass in rekening wordt genomen. De resultaten in dit werk zijn veelbelovend dus het aanpassen van het LDA model is zeker een interessante piste voor verder onderzoek. Zo kan men bijvoorbeeld de opeenvolgingen van acties proberen te verwerken in het model. Aangezien een team anders kan gaan spelen afhankelijk van het score verloop in een wedstrijd zou het ook interessant zijn om te verwerken in het model of een team aan het winnen is wanneer een actie wordt verricht.

In plaats van een LDA model zou *lda2vec* [45] kunnen gebruikt worden om typische spelpatronen te identificeren. Dit model is een combinatie van LDA en *word2vec* die tegelijk globale document onderwerpen en lokale woord patronen kan vinden. Het model zou dus eigenlijk ook in rekening kunnen brengen welke *features* er regelmatig na elkaar voorkomen in een wedstrijd wat met LDA niet mogelijk is. Dit model is echter nog redelijk nieuw en experimentëel dus werd deze methode niet verder onderzocht.

Uit feedback van de experts van de KBVB bleek dat het handig zou zijn moesten coaches en analisten de typische spelpatronen kunnen sturen naargelang de aspecten die ze interessant vinden. Een eerste idee om dit te doen was om Guided LDA[1] te gebruiken waarbij men een aantal *seed topics* kan opgeven bij het trainen van een model. De typische spelpatronen die dan geïdentificeerd worden, worden naar de richting van de *seed topics* begeleid. De typische spelpatronen die hiermee geïdentificeerd werden, kwamen inderdaad goed overeen met de opgegeven *seed topics* maar de rest van de typische spelpatronen waar geen *seed topics* van waren opgegeven bleken niet logisch en begrijpbaar. De manier die het beste werkte om het model te sturen was het toevoegen of verwijderen van bepaalde *features*. Er werden een aantal extra *features* uitgeprobeerd maar niet elke *feature* had het gewenste effect. Verder onderzoek naar het gebruik van Guided LDA en het ontwerpen van nieuwe *features* en het effect van het toevoegen van deze *features*, kan dus interessant zijn.

De DTW kost die nu gebruikt wordt om bewegingskettingen te vergelijken neemt de locatie en de volgorde van gebeurtenissen in rekening. De tijd tussen twee gebeurtenissen of van de gehele bewegingsketting wordt niet in rekening gebracht. Een variatie op DTW of een andere afstandsmaat die dit wel in rekening kan brengen, zou de matching procedure kunnen verbeteren aangezien er zo onderscheid gemaakt kan worden tussen snelle en trage bewegingen.

De bewegingskettingen die nu als *feature* gebruikt worden stellen geen conceptueel patroon voor, maar zijn reële sequenties van acties. Deze sequenties bevatten vaak heel wat ruis, zoals een kleine knik veroorzaakt door het kort drijven met de bal. De typische spelpatronen die voorgesteld worden aan de hand van deze

[1] Deze versie van Guided LDA werd gebruikt: https://gist.github.com/scign/2dda76c292ef76943e0cd9ff8d5a174a

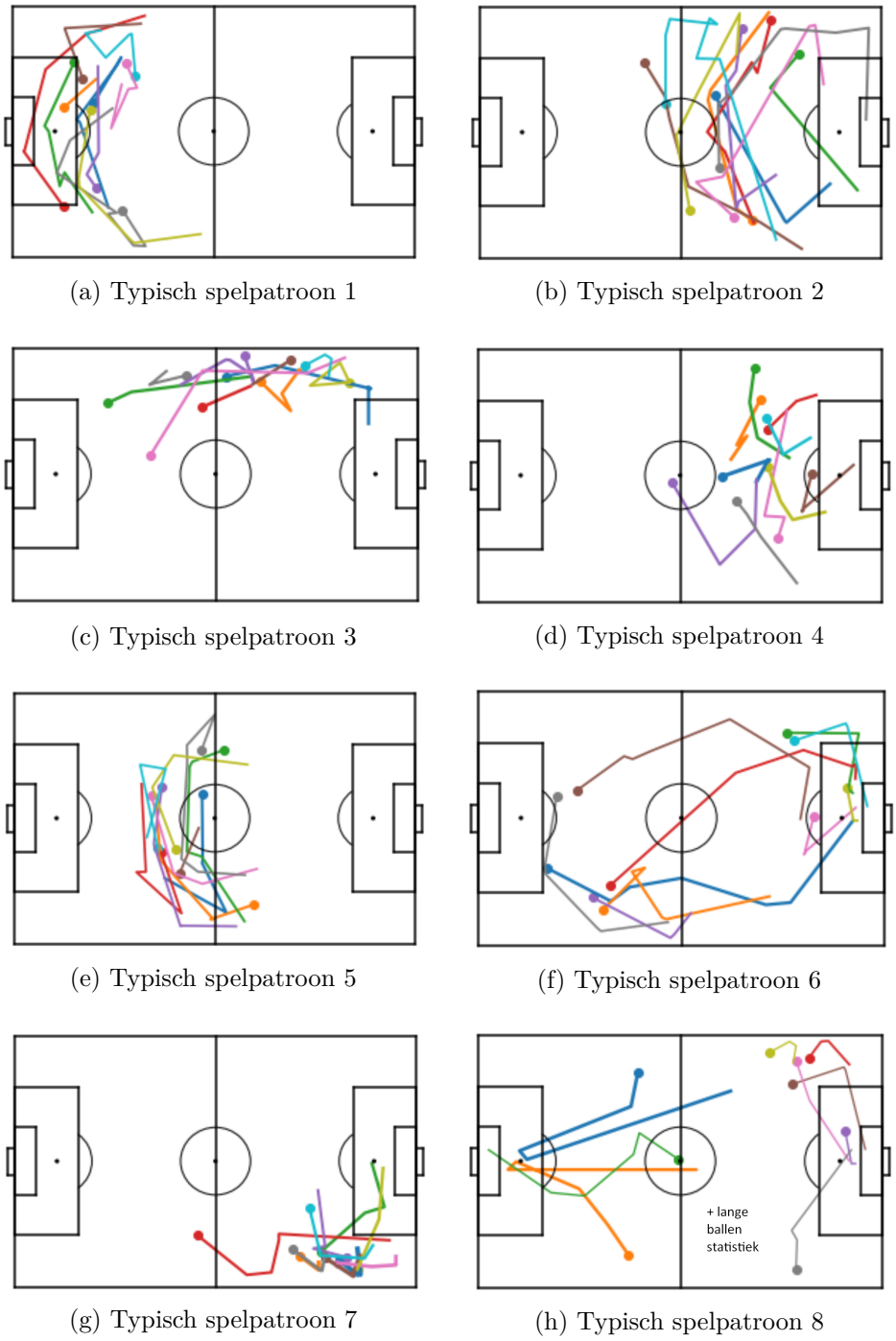
bewegingsketten zijn daardoor soms minder begrijpbaar. Een oplossing zou er uit kunnen bestaan om de bewegingsketten via een *post-processing* stap te vereenvoudigen of de clustering uit te voeren op een *feature*-representatie in plaats van op de ruwe data. De passes in een bewegingsketting zouden bijvoorbeeld beschreven kunnen worden door hun locaties in een vooraf gedefiniëerd grid, hun richting en relatieve tijdstip ten opzichte van elkaar. Een clustering op basis van deze features zou *centroids* kunnen opleveren die bewegingspatronen op hoger conceptueel niveau voorstellen. Een nadeel van deze oplossing is dan wel weer dat het aantal mogelijke *centroids* beperkter is en volledig afhangt van hoe de *features* gedefiniëerd zijn.

Bijlagen

Bijlage A

Verschillende configuraties

Deze appendix bevat de typische spelpatronen die met de verschillende configuraties van het LDA model werden geïdentificeerd. Figuur A.1 bevat de typische spelpatronen van de 2NN (vgl 3.1) configuratie. Figuur A.2 die van de 2NN (vgl 3.2) configuratie. Figuur A.3 die van de 3NN (vgl. 3.1) configuratie. Figuur A.4 die van de 3NN (vgl. 3.3) configuratie.



(a) Typisch spelpatroon 1

(b) Typisch spelpatroon 2

(c) Typisch spelpatroon 3

(d) Typisch spelpatroon 4

(e) Typisch spelpatroon 5

(f) Typisch spelpatroon 6

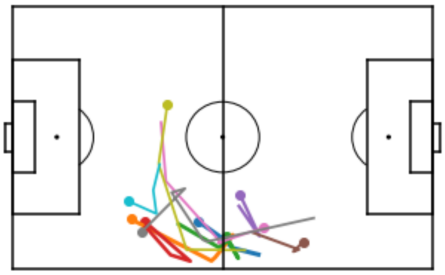
(g) Typisch spelpatroon 7

(h) Typisch spelpatroon 8

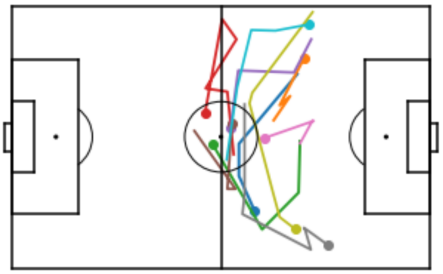
Figuur A.1: Een visualisatie van de met LDA geïdentificeerde typische spelpatronen met de 2NN (vgl. 3.1) configuratie. Bij elk typisch spelpatroon is een label toegevoegd die een goede beschrijving van het patroon geeft. Er wordt van links naar rechts gespeeld. Een bolletje stelt het begin van een bewegingsketting voor. Hoe dikker de lijn van een bewegingsketting, hoe belangrijker deze is in een typisch spelpatroon.

(a) Typisch spelpatroon 1

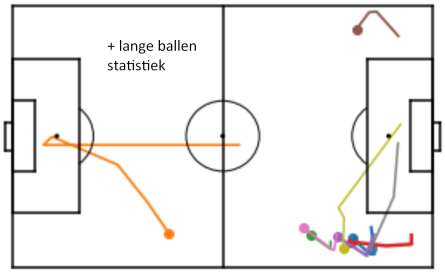
(b) Typisch spelpatroon 2



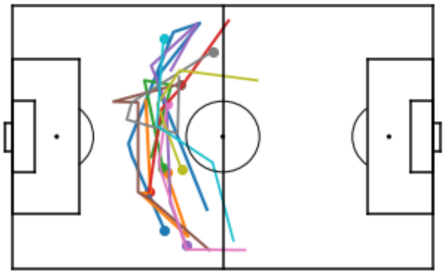
(c) Typisch spelpatroon 3



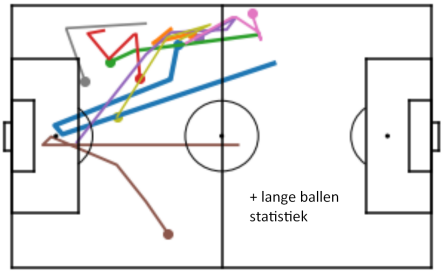
(d) Typisch spelpatroon 4



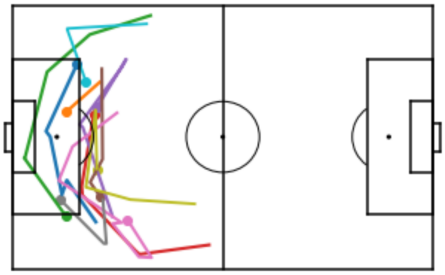
(e) Typisch spelpatroon 5



(f) Typisch spelpatroon 6

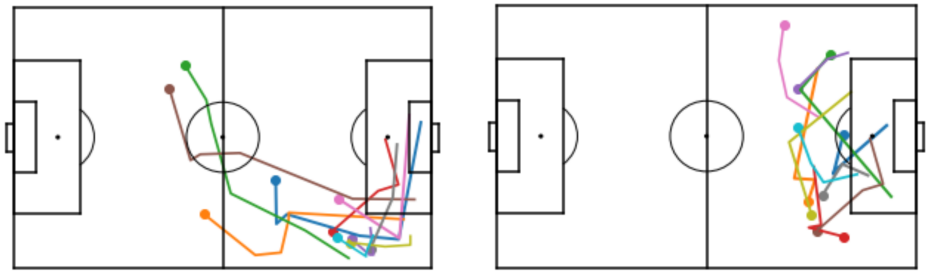


(g) Typisch spelpatroon 7



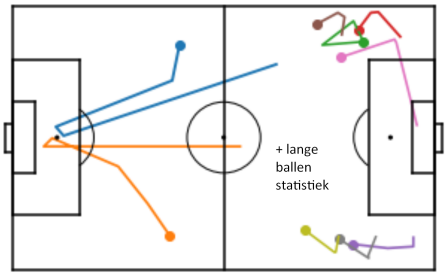
(h) Typisch spelpatroon 8

Figuur A.2: Een visualisatie van de met LDA geïdentificeerde typische spelpatronen met de 2NN (vgl. 3.2) configuratie. Bij elk typisch spelpatroon is een label toegevoegd die een goede beschrijving van het patroon geeft. Er wordt van links naar rechts gespeeld. Een bolletje stelt het begin van een bewegingsketting voor. Hoe dikker de lijn van een bewegingsketting, hoe belangrijker deze is in een typisch spelpatroon.

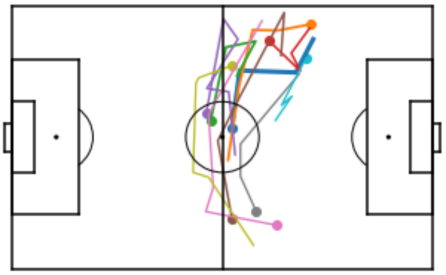


(a) Typisch spelpatroon 1

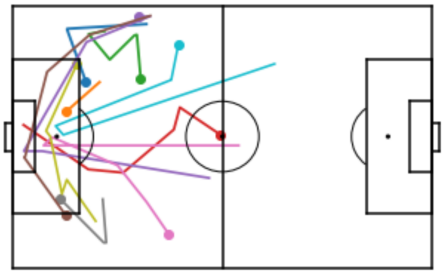
(b) Typisch spelpatroon 2



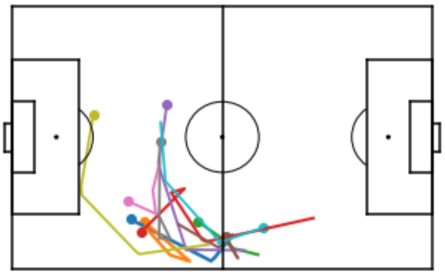
(c) Typisch spelpatroon 3



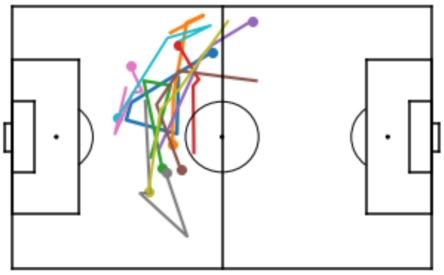
(d) Typisch spelpatroon 4



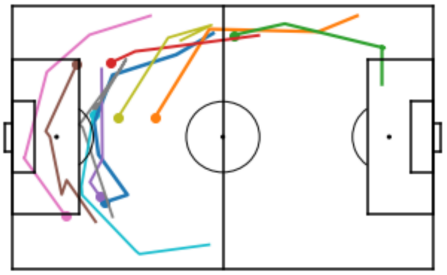
(e) Typisch spelpatroon 5



(f) Typisch spelpatroon 6

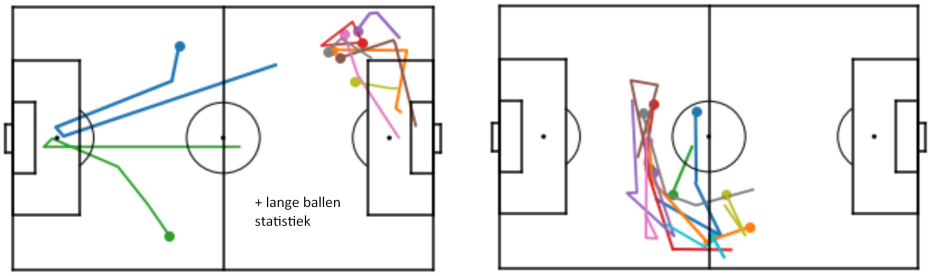


(g) Typisch spelpatroon 7



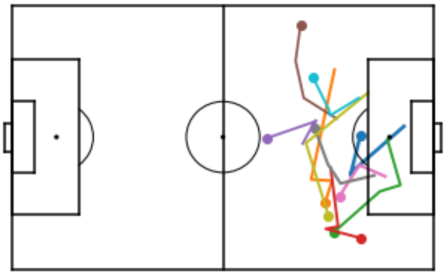
(h) Typisch spelpatroon 8

Figuur A.3: Een visualisatie van de met LDA geïdentificeerde typische spelpatronen met de 3NN (vgl. 3.1) configuratie. Bij elk typisch spelpatroon is een label toegevoegd die een goede beschrijving van het patroon geeft. Er wordt van links naar rechts gespeeld. Een bolletje stelt het begin van een bewegingsketting voor. Hoe dikker de lijn van een bewegingsketting, hoe belangrijker deze is in een typisch spelpatroon.

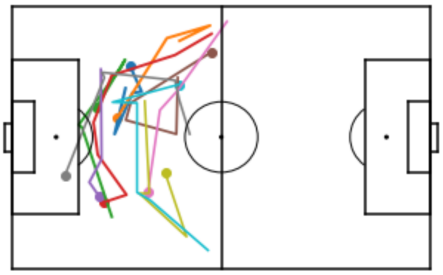


(a) Typisch spelpatroon 1

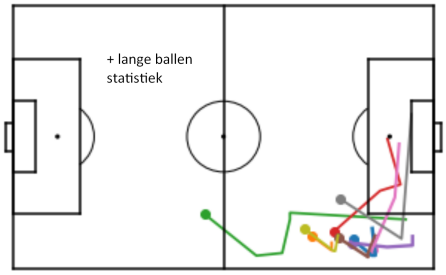
(b) Typisch spelpatroon 2



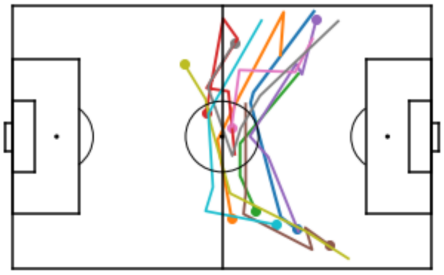
(c) Typisch spelpatroon 3



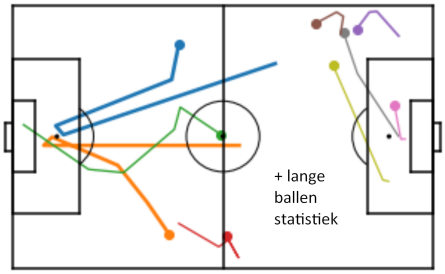
(d) Typisch spelpatroon 4



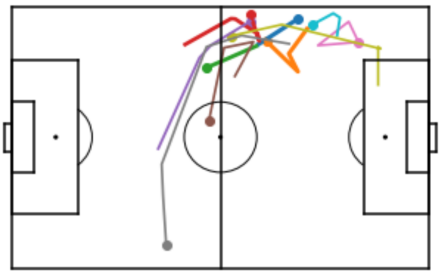
(e) Typisch spelpatroon 5



(f) Typisch spelpatroon 6



(g) Typisch spelpatroon 7



(h) Typisch spelpatroon 8

Figuur A.4: Een visualisatie van de met LDA geïdentificeerde typische spelpatronen met de 3NN (vgl. 3.3) configuratie. Bij elk typisch spelpatroon is een label toegevoegd die een goede beschrijving van het patroon geeft. Er wordt van links naar rechts gespeeld. Een bolletje stelt het begin van een bewegingsketting voor. Hoe dikker de lijn van een bewegingsketting, hoe belangrijker deze is in een typisch spelpatroon.

Bijlage B

Validatie van *feature* vectoren

Deze appendix bevat de evaluatie van de vectoren opgesteld als de gemiddelde *feature*-beschrijving van een wedstrijd voor een team.

Tabel B.1: Vergelijking van de vectoren opgesteld met de *feature*-beschrijvingen van wedstrijden van de eerste helft van het seizoen 2019/20 met die van de tweede helft.

	MRR	Top 1	Top 2	Top 3	Top 5	Top 10
1NN	0.758	65.3%	78.6%	83.7%	89.8%	91.8%
2NN (vgl. 3.1)	0.784	69.4%	80.6%	85.7%	89.8%	91.8%
2NN (vgl. 3.2)	0.761	65.3%	81.6%	87.7%	89.8%	92.8%
3NN (vgl. 3.1)	0.778	68.4%	79.6%	86.7%	89.8%	92.9%
3NN (vgl. 3.3)	0.783	68.4%	81.6%	87.7%	89.8%	92.8%

Bijlage C

Wetenschappelijke poster

Identifying a football team's playing style

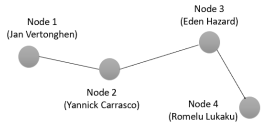
Daan Wildiers

Identifying a football team's playing style can be modelled as an NLP topic extraction problem

INTRO:

- A team's playing style is a complex combination of player interactions guided by strategies
- Nowadays the main instrument is video analysis which is time consuming and subjective
- **Interpretable representation** of playing style is needed to use in practice by coaches and analysts
  - To adapt tactics to an opponent
  - Find teams with a specific playing style for friendly games

METHODS



Ball event data
All ball involvements are registered (location, type, player)

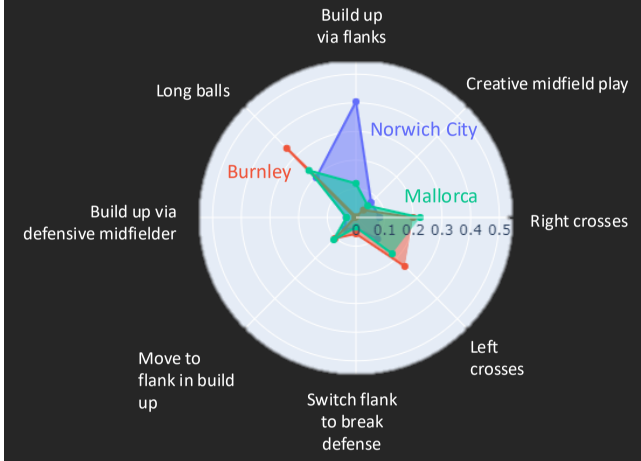
Split into movement chains
- 3 or 4 player involvements
- Final third movement chains

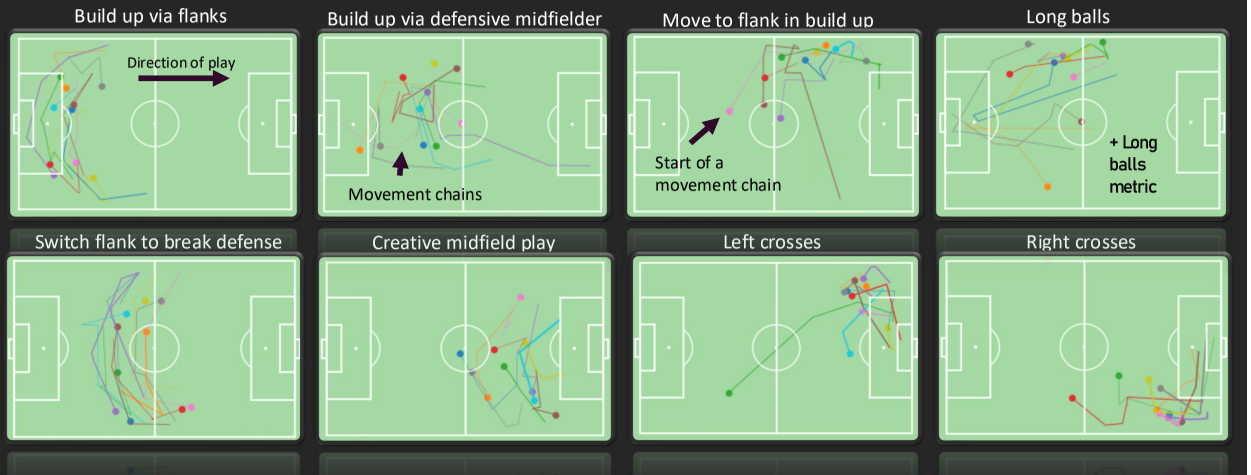
Data-driven feature construction
- **K-medoids** with **Dynamic Time Warping** as similarity
- 125 clusters which centroids are the prototypes

Construct feature matrix
- 1 nearest neighbour
- Dynamic Time Warping
- Z-score
- Other metrics (e.g. long balls)

Train Latent Dirichlet Allocation model with features
- **8 styles** identified
- Each **style** consists of a **mixture of features**
- Each **game** consists of a **mixture of styles**

Team vectors which consist of a mixture of styles
- Average game vector per team
- **Interpretable** style components



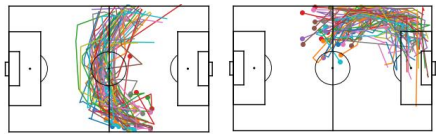


RESULTS

Can we identify a team's playing style given the styles of the previous season?

In top-k of most similar teams	K = 1	K = 5	K = 10
All teams	24 %	36 %	52 %
Teams with same coach in both seasons	33 %	46 %	72 %

Mostly top teams. Mediocre teams have a less consistent playing style



2 examples of clusters found with k-medoids with Dynamic Time Warping

	Cluster 1	Cluster 2	...	Cluster 125	Long balls
Game 1	3	1		10	5
Game 2	1	5		9	1
Game 3	4	3		4	2

Feature matrix as input for LDA model

Most similar teams to Manchester City

		Manhattan distance
1	Barcelona	0.379860
2	Paris Saint-Germain	0.401751
3	Liverpool	0.434694
4	Napoli	0.440281

Daan Wildiers
Promotor:
Prof. dr. ir. Jesse Davis
Supervisors:
Ir. Pieter Robberechts
Dr. Matteo Balliauw

KU LEUVEN

Bibliografie

[1] Analytics - Wyscout footballdata. https://footballdata.wyscout.com/advanced-metrics/. (Geraadpleegd op 16/08/2021).

[2] Sarriball and the difference of play in Napoli, Chelsea and Juventus. https://www.sports-nova.com/2020/04/14/sarriball-and-the-difference-of-play-in-napoli-chelsea-and-juventus/. (Geraadpleegd op 11/08/2021).

[3] Stats perform playing styles – An introduction. https://www.statsperform.com/resource/stats-playing-styles-introduction/. (Geraadpleegd op 14/07/2021).

[4] *Partitioning Around Medoids (Program PAM)*, chapter 2, pages 68–125. John Wiley & Sons, Ltd, 1990.

[5] M. Ángel Gómez, M. Mitrotasios, V. Armatas, and C. L.-P. nas. Analysis of playing styles according to team quality and match location in greek professional soccer. *International Journal of Performance Analysis in Sport*, 18(6):986–997, 2018.

[6] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. volume 28, pages 49–60, 06 1999.

[7] G. M. Arastey. Artificial intelligence (AI) in sports. https://www.sportperformanceanalysis.com/article/artificial-intelligence-ai-in-sports. (Geraadpleegd op 22/05/2021).

[8] J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17:305–338, 2015.

[9] J. Bekkers and S. Dabadghao. Flow motifs in soccer: What can passing behavior tell us? *Journal of Sports Analytics*, 5:1–13, 07 2019.

[10] M. Bergmann. Steffen Baumgart at SC Paderborn 2019/20 – tactical analysis. https://totalfootballanalysis.com/head-coach-analysis/steffen-baumgart-sc-paderborn-2019-20-tactical-analysis-tactics. (Geraadpleegd op 10/08/2021).

[11] C. Bialik. The people tracking every touch, pass and tackle in the world cup. https://fivethirtyeight.com/features/the-people-tracking-every-touch-pass-and-tackle-in-the-world-cup/. (Geraadpleegd op 08/07/2021).

[12] 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 Workshops, ICDMW*, 2015:9–14, 01 2015.

[13] I. I. Bojinov and L. Bornn. The pressing game : Optimal defensive disruption in soccer. 2016.

[14] J. Brooks, M. Kerr, and J. Guttag. Using machine learning to draw inferences from pass location data in soccer. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(5):338–349, 2016.

[15] R. Carvahlo. Tactical analysis: José Bordalás's Getafe. https://breakingthelines.com/tactical-analysis/tactical-analysis-jose-bordalass-getafe/. (Geraadpleegd op 10/08/2021).

[16] P. Cintia, S. Rinzivillo, and L. Pappalardo. A network-based approach to evaluate the performance of football teams. 09 2015.

[17] S. P. Crain, K. Zhou, S.-H. Yang, and H. Zha. *Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond*, pages 129–161. Springer US, Boston, MA, 2012.

[18] J. Davis. Data mining b-kul-h02c6a, 2021.

[19] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis. Actions speak louder than goals: Valuing player actions in soccer. 02 2018.

[20] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pages 1851–1861, New York, NY, USA, 2019. ACM.

[21] T. Decroos and J. Davis. Player vectors: Characterizing soccer players' playing style from match event streams. Machine Learning and Knowledge Discovery in Databases, pages 569–584. Springer International Publishing.

[22] T. Decroos, J. Van Haaren, and J. Davis. *Automatic Discovery of Tactics in Spatio-Temporal Soccer Match Data*. 2018.

[23] T. Decroos, M. Van Roy, and J. Davis. Soccermix: Representing soccer actions with mixture models. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020.

[24] R. Desmond. Antonio Conte – Inter Milan – Tactical analysis (2020-21 edition). https://themastermindsite.com/2021/03/19/antonio-conte-inter-milan-tactical-analysis-2020-21-edition/. (Geraadpleegd op 10/08/2021).

[25] J. Diquigiovanni and B. Scarpa. Analysis of association football playing styles: An innovative method to cluster networks. *Statistical Modelling*, 19(1):28–54, 2019.

[26] S. A. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):325–327, 1976.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[28] G.-f. Fan, Y.-H. Guo, J.-M. Zheng, and W.-C. Hong. Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. *Energies*, 12:916, 03 2019.

[29] J. Fernandez-Navarro, L. Fradua, A. Zubillaga, P. Ford, and A. McRobert. Attacking and defensive styles of play in soccer: analysis of spanish and english elite teams. *Journal of sports sciences*, 34:1–10, 2016.

[30] J. Fernandez-Navarro, L. Fradua, A. Zubillaga, P. R. Ford, and A. P. McRobert. Attacking and defensive styles of play in soccer: analysis of spanish and english elite teams. *Journal of Sports Sciences*, 34(24):2195–2204, 2016. PMID: 27052355.

[31] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[32] F. Fusi. The best Serie A team that you (probably) don't know are Hellas Verona. https://statsbomb.com/2020/02/the-best-serie-a-team-that-you-probably-dont-know-are-hellas-verona/. (Geraadpleegd op 11/08/2021).

[33] L. Gyarmati and X. Anguera. Automatic extraction of the passing strategies of soccer teams, 2015.

[34] L. Gyarmati and M. Hefeeda. Analyzing in-game movements of soccer players at scale, 2016.

[35] L. Gyarmati, H. Kwak, and P. Rodriguez. Searching for a unique style in soccer, 2014.

[36] M. Hoffman, D. Blei, and F. Bach. Online learning for latent dirichlet allocation. volume 23, pages 856–864, 11 2010.

[37] D. Ibanez. Topic modeling with latent dirichlet allocation. https://www.baeldung.com/cs/latent-dirichlet-allocation. (Geraadpleegd op 30/07/2021).

[38] C. Lago-Peñas, M. Gómez-Ruano, and G. Yang. Styles of play in professional soccer: an approach of the chinese soccer super league. *International Journal of Performance Analysis in Sport*, 17(6):1073–1084, 2017.

[39] D. K. Leuven. Data representation – socceraction 1.0.2 documentation. https://socceraction.readthedocs.io/en/latest/documentation/SPADL.html. (Geraadpleegd op 11/07/2021).

[40] S. Li. Topic modeling quora questions with lda & nmf. https://towardsdatascience.com/topic-modeling-quora-questions-with-lda-nmf-aff8dce5e1dd. (Geraadpleegd op 29/07/2021).

[41] P. Lucey. AI in soccer. https://vimeo.com/515977363/be3de09fc1. (Geraadpleegd op 22/05/2021).

[42] B. Mabey. Python library for interactive topic model visualization. port of the R LDAvis package. https://github.com/bmabey/pyLDAvis. (Geraadpleegd op 09/08/2021).

[43] W. Meert, K. Hendrickx, and T. V. Craenendonck. wannesm/dtaidistance v2.0.0, Aug. 2020.

[44] A. C. Miller and L. Bornn. Possession sketches : Mapping NBA strategies. 2017.

[45] C. Moody. Introducing our hybrid lda2vec algorithm. https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/#topic=38&lambda=1&term=. (Geraadpleegd op 12/08/2021).

[46] J. Muller. The seven styles of soccer. https://spacespacespaceletter.com/the-seven-styles-of-soccer/. (Geraadpleegd op 20/07/2021).

[47] M. Müller. Dynamic time warping. *Information Retrieval for Music and Motion*, 2:69–84, 01 2007.

[48] J. L. P. na. A markovian model for association football possession and its outcomes, 2014.

[49] A. Novikov. PyClustering: Data mining library. *Journal of Open Source Software*, 4(36):1230, apr 2019.

[50] D. Perdomo Meza. Automated playing style detection of soccer teams and players using latent dirichlet allocation. Technical report, MIT Sloan Sports Analtyics Conference, 2018.

[51] D. Perdomo Meza and D. Girela. Optapro analytics forum 2020 - tactical insight through team personas, 2020.

[52] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[53] J. T. Raj. A beginner's guide to dimensionality reduction in machine learning. https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e. (Geraadpleegd op 28/07/2021).

[54] R. Rehurek and P. Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[55] D. Reynolds. *Gaussian Mixture Models*, pages 659–663. Springer US, Boston, MA, 2009.

[56] P. Robberechts, T. Decroos, and J. Davis. Introducing Atomic-SPADL: A new way to represent event stream data. https://dtai.cs.kuleuven.be/sports/blog/introducing-atomic-spadl-a-new-way-to-represent-event-stream-data. (Geraadpleegd op 20/05/2021).

[57] M. V. Roy. Convert existing soccer event stream data to SPADL and value player actions. https://github.com/MaaikeVR/socceraction. (Geraadpleegd op 18/05/2021).

[58] R. J. Samworth. Optimal weighted nearest neighbour classifiers. *The Annals of Statistics*, 40(5), Oct 2012.

[59] SciSports. The power of combining tracking and event data. https://www.scisports.com/the-power-of-combining-tracking-and-event-data/. (Geraadpleegd op 24/05/2021).

[60] L. Shushkova. Roberto De Zerbi: Sassuolo. https://totalfootballanalysis.com/article/roberto-de-zerbi-at-sassuolo-2020-21-tactical-analysis-tactics. (Geraadpleegd op 10/08/2021).

[61] C. Sievert and K. Shirley. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[62] K. Singh. Introducing expected threat (xt). https://karun.in/blog/expected-threat.html. (Geraadpleegd op 15/08/2021).

[63] StatsBomb. Statsbomb data launch – New pressure events. https://www.youtube.com/watch?v=JlXpOTVJUQw. (Geraadpleegd op 11/07/2021).

[64] L. Uomo. The tactical revolution of Italian Serie A. https://blog.wyscout.com/serie-sarri-conte-giampaolo-fonseca/. (Geraadpleeg op 11/08/2021).

[65] J. Van Haaren, V. Dzyuba, S. Hannosset, and J. Davis. Automatically discovering offensive patterns in soccer match data. Advances in Intelligent Data Analysis XIV, pages 286–297. Springer International Publishing, 2016.

[66] J. Van Haaren, S. Hannosset, and J. Davis. Strategy discovery in professional soccer match data. pages 1–4, 2016.

[67] J. Van Haaren, P. Robberechts, T. Decroos, L. Bransen, and J. Davis. Analysing performance and playing style using ball event data. In *Football Analytics: Now and Beyond. A Deep Dive into the Current State of Advanced Data Analytics.*, pages 36–47. Barça Innovation Hub, 2019.

[68] Q. Wang, H. Zhu, W. Hu, Z. Shen, and Y. Yao. *Discerning Tactical Patterns for Professional Soccer Teams: An Enhanced Topic Model with Applications*, pages 2197–2206. Association for Computing Machinery, New York, NY, USA, 2015.

[69] X. Wei, L. Sha, P. Lucey, S. Morgan, and S. Sridharan. Large-scale analysis of formations in soccer. In *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2013.

[70] J. Whitmore. Introducing movement chains. https://www.statsperform.com/resource/introducing-movement-chains/. (Geraadpleegd op 17/07/2021).

[71] J. Whitmore. What are expected goals (xg)? https://www.theanalyst.com/eu/2021/06/what-are-expected-goals-xg/. (Geraadpleegd op 15/08/2021).

[72] J. Xu. Topic modeling with lsa, psla, lda & lda2vec. https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05. (Geraadpleegd op 29/07/2021).