A Dockerized Big Data Architecture for Sports Analytics

Yavuz Melih Özgüven
Kocaeli University: Kocaeli Universitesi

Utku Gönener
Kocaeli University: Kocaeli Universitesi

Süleyman Eken (✉ suleyman.eken@kocaeli.edu.tr)
Kocaeli University https://orcid.org/0000-0001-9488-908X

Research Article

Keywords: Big data, sports analytics, Apache Spark, containers, wearable devices, IoT

Posted Date: June 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-524005/v1

Noname manuscript No.
(will be inserted by the editor)

A Dockerized Big Data Architecture for Sports Analytics

Yavuz Melih Özgüven · Utku Gönener ·
Süleyman Eken[*]

Received: date / Accepted: date

Abstract The revolution of big data has also affected the area of sports analytics. Many big companies have started to see the benefits of combining sports analytics and big data to make a profit. Aggregating and processing big sport data from different sources becomes challenging if we rely on central processing techniques, which hurts the accuracy and the timeliness of the information. Distributed systems come to the rescue as a solution to these problems and the MapReduce paradigm is promising for large-scale data analytics. In this study, we present a big data architecture based on Docker containers in Apache Spark. We demonstrate the architecture on four data-intensive case studies including structured analysis, streaming, machine learning methods, and graph-based analysis in sport analytics, showing ease of use.

Keywords Big data · sports analytics · Apache Spark · containers · wearable devices · IoT

1 Introduction

Statistics in sports is a growing field in statistics that provides specialized methodology for collecting and analyzing sports data in order to make decisions for successful planning and implementation of new strategies. Decision making in sports based on the information acquired by observation has

Y. M. Özgüven
Department of Computer Engineering, Kocaeli University, 41001 İzmit, Turkey

U. Gönener (ORCID: 0000-0002-6152-3353)
Faculty of Sports Sciences, Kocaeli University, 41001 İzmit, Turkey
E-mail: utku.gonener@kocaeli.edu.tr

[*]Corresponding author
S. Eken (ORCID: 0000-0001-9488-908X)
Department of Information Systems Engineering, Kocaeli University, 41001 İzmit, Turkey
E-mail: suleyman.eken@kocaeli.edu.tr

changed with technological advances. Sports analytics has been more popular in recent days [1].

Sports analytics is broadly described as the process of data management, predictive model implementation, and the use of information systems for decision making to gain a competitive advantage on the field of play [2]. Since sports analytics has the concept of using sports data to create valuable statistics for analysis the need for proper data models is present. There are different approaches on how to treat sports data in combination with data analytics to create statistics and other beneficial information. Sports analytics is also in need of getting correct data from matches to analyze. To collect match data, different kind of technologies are applied to provide large amounts of data, many in which a lot of detailed statistics could be constructed. The big collection of sports data then benefits the analysis and decision making from the sports games [3].

Sports analytics is the concept of treating sports data through analytic methodologies to help make valuable conclusions [4]. Analytics is applied to conclude advantages in the exercising of sports. The conclusions need to be originated from established data analytics, according to mathematical models that the sports industry has evaluated and are used in some manner. These changes in sports science and analytics have also taken place in the academic field. Some research journals have been created that are totally devoted to analytics in sports such as Journal of Quantitative Analysis in Sports (JQAS)[1] and Journal of Sports Analytics (JSA)[2]. In addition to journals and books, sports analysts publish their work on blog sites as well. Sports analytics conferences are also a platform for professionals, researchers and students to discuss related topics in sports [5]. Besised all these, a group of SFU faculty, coaches and students who have a passion for sports and analytics formed the Sports Analytic Group (SAG) in 2015 at Simon Fraser University (SFU)[3].

Since the data rate has gone up in the latest years the need for efficient big data analytics has become more and more important. The increasing smart devices being carried have made data rate explode, and with the increasing sensors and interactions in society some smart solutions need to be carried [6]. Not only the increasing devices has made an impact, but also the behavior of the users. A technology such as positioning generates a boosted amount of data, and there are many areas such as business data, image data and industrial process data [7].

Aggregating and processing big data from all of these sources becomes challenging if we rely on central processing techniques, which hurts the accuracy and the timeliness of the information. Therefore, we need to adapt distributed and parallel computing technologies in the research of sports analytics. A distributed system is a group of separate and self-sufficient computing elements (nodes) combined and presented to its users as a single coherent system. Each

[1] https://www.degruyter.com/journal/key/JQAS/html

[2] https://journalofsportsanalytics.com/

[3] www.sfu.ca/sportsanalytics.html

node is autonomous and has its own notion of time. This lack of a global clock leads to major synchronization and coordination problems. The scalability of distributed systems can be achieved in various ways: size scalability, geographical scalability, administrative scalability. They denote the number of users and/or processes, the maximum distance between nodes, and the number of administrative domains, respectively. Size scalability is often the one problem most addressed by such systems. Parallel computing, where multiple powerful servers operating independently in parallel, is an alternative solution. In this model, however, a global clock is a requirement to synchronize the processing done independently by simultaneously on multiple sub-tasks during each clock-cycle, and combine their results to solve the original task. Parallel computing, cluster computing, grid computing, and cloud computing are kind of high performance distributed computing mechanisms [8]. Considering big data processing and analytics, emergent hardware technologies and new computing paradigms such as co-processors, fog computing, and dew computing are possible [9].

1.1 Contributions

Contributions to the literature with the paper can be listed as follows:

- Current MapReduce based frameworks offer poor support for reusing existing processing tools in sports data analytics pipelines. We give an open source architecture that introduces support for Docker containers in Apache Spark.
- We demonstrate the architecture on four data-intensive applications including structured analysis, streaming, machine learning methods, and graph-based analysis in sport analytics, showing ease of use.

1.2 Paper organization

The remainder of this article is organized as follows. Section 2 gives a literature review on structured sports data analysis, sport data streaming, machine learning approaches in sports, and graph-based sport data analysis. Section 3 gives a sports data search mechanism and repository analysis from a reproducible research perspective. Section 4 gives details of the containerized big data architecture. Section 5 presents the performance of the system. Section 6 summarizes and and also gives lessons learned and future work.

2 Related Works

This section will demystify the analytical thinking behind the data revolution in sports through a wide range of topics related to sport data analytics in the literature. We organize this section as four sub-sections.

2.1 Structured sport data analysis

We can classify data analysis utilisation depending on the velocity and variation of data i.e. real-time, batch processing, and structured, semi-structured, unstructured. Analytics can acquire both insights and foresight from the data. An ELT process, extract-load-transform, where data is extracted and loaded in a raw format and transformation steps are diverted towards the database engine to be performed as small atomic tasks through SQL statements. The transformed data is then moved into a data model that is accessible by users. Following paragraph consists of structured sport data analysis related works.

Metulini [10] concerned with basketball data processing, and aimed to suggest an ad-hoc procedure to automatically filter a data matrix containing players' movement information to the moments in which the game is active, and by dividing the game into sorted and labelled actions as offensive or defensive. Knobbe et al. [11] worked on professional speed skating and devised a number of features that capture various aspects of sports events by aggregating discrete sequences of such events. The aggregation can be done in two ways: one that is easy to compute and interpret (uniform window), one that is more physiologically plausible but harder to compute (the Fitness-Fatigue model). SQL statements were used to perform these aggregations. Pers et al. [12] also used standard SQL to analyze large volumes of annotated sport motion data. Their goal was to automatically detect certain kinds of play, activities, predefined scenarios and to generate various related statistics.

2.2 Sport data streaming

In recent years, a number of inexpensive wearable devices and gadgets aiming for the sport tracking and monitoring have been introduced to the sport market. Sport tracking and monitoring systems share many key technological trends of other Internet of Things (IoT) solutions. Cloud and fog computing principles can be solution to problem, of the real-time analysis and feedback of these IoT devices, caused by latency and bandwidth limitations.

Pustišek et al. [13] discussed the importance of technology for motor learning in sports, and studied the properties and limitations of various sensors used for activity signal acquisition, means of communication, and communication channels. They designed feedback systems that satisfy a wide range of possible uses for augmented motor learning with the help of smart sports equipment. Grün et al. [14] designed a system capable of tracking a large number of high dynamic objects within a pre-defined area of interest in real-time, like during a football game. Probst et al. [15] designed a complete team sports analysis infrastructure by combining their real-time analysis system STREAMTEAM and their video retrieval system SPORTSENSE. This system can automatically detect collaborative events, generate statistics based on a continuous stream of raw positions, visualize the analysis results, all in real-time, and then put the analysis results in persistent storage for offline activities and intuitive sketch-based video retrieval later. Capobianco et al. [16] proposed a formal methodology for designing an expert system based on big data acquired from various sources, the purpose of the system is to support real-time decisions for notational analysis in a sports environment. Haiyun and Yizhe [17] developed a Hadoop platform for predicting game results which is an integrated learning and a comprehensive learning algorithms. Dinesh et al. [18] proposed a real time violence detection framework for football stadium. HOG function was used to extract the features from the video frames in Spark environment. Proposed system alerts the security forces. Baerg [19] considered the relationship between Big Data and the athlete. Stein et al. [20] explained how to analyze team sport data in general then proposed a multi-facet view and analysis including pattern detection, context-aware analysis, and visual explanation. Luo et al. [21] reported a flexible and durable wood-based triboelectric nanogenerator for self-powered sensing in athletic big data analytics.

2.3 Machine learning based sport data analysis

This subsection summarizes machine learning based sport data analysis in the literature. Podgorelec et al. [22] built a new image dataset of four similar sports (American football, rugby, soccer, and field hockey) and developed a method to classify those images using transfer learning of CNN with Hyper-Parameter Optimization (HPO). Their proposed method was then compared to a conventional CNN and a CNN with transfer learning but handpicked hyper-parameters for fine-tuning. Constantinou et al. [23] developed probabilistic models based on possession rates and other historical statistics of various teams to predict the outcome of matches. Kapadia et al. [24] used machine learning techniques to solve the same problem but for the cricket world in the Indian Premier League (IPL). Jayalath [25] considered the popular logistic regression model to study the significance of one-day international (ODI) cricket predictors. Kerr [26] presented three experiments in his thesis. In the first experiment, three models were constructed using different features to predict which team won a given game, without any knowledge of goals. In the second experiment, several classifiers were used to predict which team produced the sequence of ball-events that occurred during a game. And in the last one, he predicted which team attempted a given set of passes. Brooks et al. [27] focused on examining characteristics of passing in soccer and introduced two methods for obtaining insights from that. Ehrlich and Ghimire [28] took note of the effect the presence or absence of fans can have on a team's performance in Major League Baseball. He analyzed various scenarios in the context of physical distancing due to COVID-19 and used logit regression and a neural network to simulate the 2020 season.

Ghimire et al. [29] used Adjusted Plus-Minus (APM) measures to evaluate player contribution in basketball and hockey. APM measures estimate the impact of an individual player on his team's scores using seasonal play-by-play data. They run sets of linear fixed effects regression models to explain variation in Real Plus-Minus (RPM) across player-seasons and checked robustness using a two-stage least square (2-SLS) method. Knobbe et al. [11] used linear modelling and subgroup discovery in order to select key features and produce interpretable models for sport data analytics in professional speed skating. Vinué and Epifanio [30] developed a useful mathematical tool based on archetypoid analysis (ADA) to analyze sporting performance and to assess the value of players and teams in a league. The utility of archetypoids in sports was illustrated with basketball and soccer data in three scenarios. Janetzko et al. [31] introduced a system to interactively explore and analyze movement features and game events using various levels of details in high-frequency position-based soccer data. Sidle and Tran [32] applied multi-class classification methods to the problem of predicting baseball live pitch types. While Chu and Swartz [33] proposed a Bayesian inference system with parametric models to analyze fouling time distributions. Karetnikov [34] proposed a principally new complex performance prediction framework for cycling with are the Maximum Mean Power (MMPs) and the race position performance metrics.

2.4 Graph-based sport data analysis

This subsection summarizes graph based sport data analysis in the literature. Duch et al. [35] and Pena and Touchette [36] examined weighted pass graphs. Players were represented by nodes, passes by edges, and the efficiency of passes by weights. Cintia et al. [37] used network centrality measures for analyzing a passing network. Taking two perspectives into account: passes between players and passes between pitch zones. Zheng et al. [38] predicted game outcomes from available sports statistics using a graph signal processing (GSP) perspective. Roane et al. [39] developed an approach to sports rankings that reflects the strength of each team while accounting for game results. They represented teams and the games between them as a digraph and considered minimizing the number of backedges in a ranking. Brandt and Brefeld [40] presented a graph-based approach to analyze player interaction in team sports. Shi and Tian [41] used a game graph from the perspective of Bayesian correction with game results to build a generalized PageRank model for sports. Wu et al. [42] created a social network from player positions and passings to comprehensively measure the importance of playing positions. Features such as degree, closeness, betweenness, eigenvector, and load centralities, as well as reciprocity, and clustering were used. Football Passing Networks[4] is an interactive web application to explore data visualizations on soccer passing networks.

Although there are already approaches focused on different aspects of sports, to the best of our knowledge, there is no open source containerized big data architecture yet that jointly supports the structured-based, stream-based, machine learning-based, and graph-based sports data analytics. All of these topics are the most used types of data analysis in other big data fields.

[4] https://grafos-da-bola.netlify.app/

3 Sport Dataset Search and Repository Analysis

This section covers searching a problem-specific dataset and repository analysis related to sport data analytics.

3.1 Sports Dataset Search

Many types of datasets exist in sports such as (i) raw dataset: game box scores; play-by-play; player tracking, (ii) extracted events: hits, runs, points, rebounds, assists, etc, and (iii) stats: batting avg, total bases, RBI, shooting %, etc. In general, it is very difficult to find a public dataset for a problem, and it is a problem that anyone cannot predict how much the dataset she/he find will work. Briefly, dataset sources can be summarized as follows: (i) websites: -leagues: MLB.com, NBA.com, -general: ESPN, baseball/basketball/football reference, FanGraphs, (ii) API/published: PitchF/X, Statcast, NBA Stats, (iii) curated (not necessarily free): Lahman Database, Retrosheet, armchair-analysis.com (cheap with .edu email), and (iv) other: -API tools and scrapers published on GitHub (lots of repos out there), -data collectives: Kaggle, data.world.

When seeking high-quality datasets, there are a few things to consider:

- The dataset should not be messy, otherwise significant time will be wasted on cleaning it. The cleaner, the better.
- The dataset should not have too many rows or columns, so it is easy to work with.
- There should be a question/decision to answer using the data.

Anyone can find a public dataset related to different sport branchs using well-known repositories such as Google Dataset Search, Kaggle, UCI Machine Learning Repository, and Data.gov. Also, there are different specific data sources such as StatsBomb Open Data[5], open football[6] for soccer, NFLsavant.com[7] for Amercan football, Lahman's Baseball Database[8] for baseball, FiveThirtyEight[9] for others, Sport Database [43] for cardiorespiratory data, Heimdallr [44] for action recognition and pose estimation and etc.

3.2 Repository Analysis

GitHub[10], a hosting platform for open-source software projects, has gained much popularity in recent years [45]. In contrast with competitors (e.g., Source-

[5] https://statsbomb.com/academy/

[6] https://openfootball.github.io/

[7] http://nflsavant.com/

[8] http://www.seanlahman.com/baseball-archive/

[9] https://data.fivethirtyeight.com/

[10] https://github.com/

Forge[11], Assembla[12]), Github offers more than just version control hosting, but also an easy-to-use and cheap or free (depending on the version) online tool for collaborative software development and other attractive features[46].

We consider sports analytics repositories and their data on GitHub to follow their growth and development processes. We use git command-line search CLI[13] to retrieve git repository "statistics". It provides a cli for searching github.com and supports repositories, code, issues and commits. These "statistics" include repos, code, commits, issues, users, wikis, and topics. Table 1 shows statistics for "sport analytics" keyword. According to these statistics, there are 297 repos which titles include "sport analytics". Mostly used three languages in repos are Jupyter Notebook, Python, and R. These are also mostly used languages in other data analysis/analytics works [47]. Similarly, other keywords related to sport such as "sport", "sport data", "sport materials", and "sport activity" can be searched.

Table 1: GitHub statistics for "sport analytics" keyword

Repositories	297 repos
Language	Jupyter Notebook (67), Python (45), R (44), HTML (30), Java (9), JavaScript (7), PHP (3), MATLAB (2), C (1), C++ (1)
Commints	58 commits
Issues	16 issues States (8 Closed and 8 Open) Languages (Python (5), Java (4), JavaScript (2), HTML (1), Jupyter Notebook (1), R (1))
Topics	# sports-analytics (121 repositories) # sport-analytics (7 repositories)
Wikis	7 wiki results
Users	8 users

Repositories also serve reproducibility. Reproducibility is the minimum attainable standard for assessing scientific claims. To fulfill this, researchers are required to make both their data and computer code available to their peers. This, however, still falls short of full replication since independently collected data is not used. Nevertheless, this standard allows an assessment to some degree by verifying the original data and codes [48, 49].

4 Containerized Architecture

This section gives the players of the our containerized big data architecture: Apache Spark and Docker.

[11] https://sourceforge.net/

[12] https://www.assembla.com/

[13] https://github.com/feinoujc/gh-search-cli

4.1 Apache Spark

The amount of data being processed when streaming sports data, especially with multiple users and when streaming a broad set of activities, commands large amounts of computing power that cannot be provided by solely scaling up, meaning increasing the performance of a single machine. Instead, the performance required is achieved by scaling out, meaning distributing the computation across multiple machines [50]. Spark manages this scaling out by abstracting these machines as so-called execution nodes (worker nodes, slave nodes), on which programs (tasks), called sparkjobs, are run. These abstract execution nodes can also be separate processes on a single machine, efficiently utilizing multiple cores. Apache Spark can run in stand-alone settings, as well as on some popular platforms (e.g., Kubernetes [51], Mesos [52], and Hadoop YARN [53]).

The distribution of tasks to these nodes, and the collection of results from them, is managed by the master node (driver node). It utilizes a HDFS (Apache Hadoop Distributed File System) to persist data across these nodes [54]. An illustration of this architecture can be seen in Fig. 1.

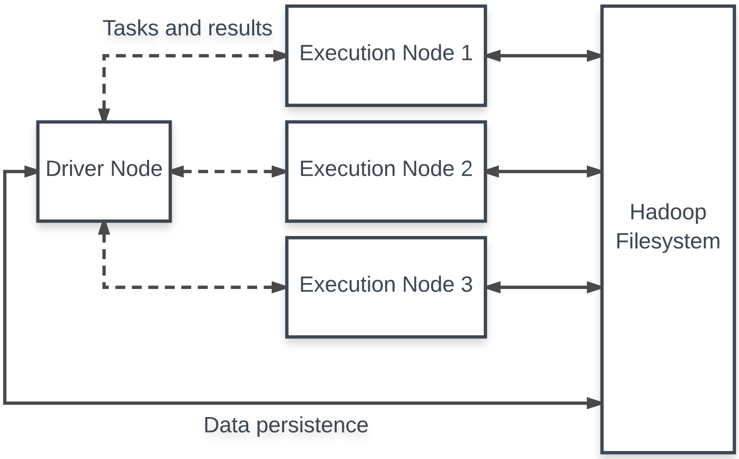


Fig. 1: Simplified diagram of a typical Spark cluster

Spark also offers functionality to perform machine learning, graph processing, structured data analysis and more on data from streams, files or databases, in a distributed setting, with just a few lines of code [55]. This means that Spark unifies and simplifies a lot of tasks in one framework that previously required multiple technologies. This has lead to a widespread adoption of Spark since its release in 2010, making it the biggest open source big data project [54], with over 1600 contributors [56] and over 1000 adopting organizations. To enable efficient implementation of big data tasks, Spark introduces a concept called RDDs (Resilient Distributed Datasets), through which the parallelization, distribution and persistence of data is abstracted for the developer [57], see Fig. 2 for a simple example.

```
data = [1, 2, 3, 4, 5]
# Wrap the data into a RDD, which is distributed across
# execution nodes by Spark, ready for parallel processing
# sc is the so-called streaming context,
# which provides an interface with the cluster
distributedData = sc.parallelize(data)
# Add a map-action that increments each value,
# distributed across execution nodes
incrementedData = distributedData.map(lambda a: a + 1)
# Run a reduce-transformation, which is run on the driver node
# The RDD is automatically collected (persisted) to the driver node
incrementedData.reduce(lambda a, b: a + b)
# returns 20
```

Fig. 2: A simple code example showing how spark distributes data and collects the result back to the driver node

Furthermore, Spark is developed by the Apache Software Foundation, as is Kafka, which means they are designed to work well with each other. For example, the Python API of Spark offers a range of utility functions to build sparkjobs to consume a Kafka-stream very easily, which means constructing a sparkjob to act as a consumer for a Kafka-stream in a distributed setting can be achieved with very few lines of code, as can be seen in the wordcount-example in Fig. 3.

```
# Stream the data in 1-second windows
streaming_context = StreamingContext(sc, 1)  # 1 second window
# Connect to a kafka stream, specifying which kafka-topics to consume.
# See section "Kafka" for an explanation of topics.
stream = KafkaUtils.createStream(streaming_context,
                                'docker:2181',
                                "stream-1",
                                {"topic-1": 1})
# Each window, count each word in each line
counts = stream.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b)
```

Fig. 3: Consuming data from a stream and processing them in batches

Spark also allows for more complex functions to be applied to the data. The previous examples shown in Figs. 2 and 3 exclusively used lambda expressions. However, more complex functions cannot be sensibly realized as lambda expressions, as they are by definition limited to a single expression. Also, developers might need to use variables defined outside of the function because their initialization is computationally intensive, or uses data that is only available on the driver node. One real strength and important characteristic of Spark is that it can transmit the whole closure of a sparkjob to the execution nodes, as long as they are serializable. This means that computationally expensive initialization of variables only need to be done once instead of on every execution node. Furthermore, imported packages such as libraries and frameworks are transmitted as well, which simplifies their usage in sparkjobs.

4.2 Docker

A container image is a packaged light-weight piece of software including, within itself, everything required to run correctly: code, run-time, system tools, system libraries, and settings. A container isolates software from its surroundings and will always run the same way regardless of the operating system or environment (e.g. development and staging). Make it possible to densely pack multiple apps on the same infrastructure. And help reduce conflicts between teams running different software on the same infrastructure [58], or running the same software on different machines. From Fig. 4 it is seen that in one single host there are three containers running. Each container contains the necessary environment variable inside. So, it is not necessary to have the all environment variable before in a host to run the application. Container itself will create the environment to run the application.

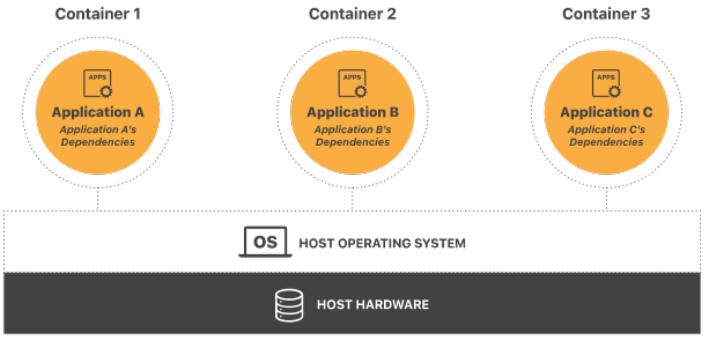


Fig. 4: Container architecture

Docker is one of the most popular software containerization platforms. Developers, operators, and enterprises use it for the previously mentioned merits.

By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, user can significantly reduce the delay between writing code and running it in production. The isolation and security allow user to run many containers simultaneously on a given host [59].

5 Experimental Results and Discussion

5.1 Experimental Setup

All performance tests are done on Microsoft Azure. Specifications of used server are as following: Zone: East-US, Cpu: 2 core, Memory: 8 GB, OS: Ubuntu 16.04-LTS, Disk: 30 GB. Package dependencies are as following: spark-core_2.12, spark-sql_2.12, spark-mllib_2.12, isolation-forest_3.0.0_2.12, spark-graphx_2.10, pulsar-client 2.6.2, and pulsar-spark 2.6.2. Also, we use an open source programming library, MaRe [60]. It enables scalable data-intensive processing.

5.2 Case Studies

5.2.1 Case study 1: Extracting interesting information about footballers with SQL statements

Spark SQL is a module for managing structured data. With Spark SQL, it is feasible to query structured data by utilizing either structured query language or a similar API. It can be used with Python, R, and similar languages. It ensures uniform data access. SQL and DataFrames supply a common way to connect to various data sources, including JDBC, Hive, JSON, Parquet, etc. Spark SQL can scale up to hundreds of nodes simultaneously by utilizing the Spark framework.

In Apache Spark, there are various abstractions for data: Resilient Distributed Datasets (RDDs), DataFrames, Datasets, and SQL Tables. All of these various abstractions show distributed collections of data. RDD was the main API in Spark since its beginning. RDD is an unchangeable distributed compilation of data. RDD is split over nodes in the cluster and might be used simultaneously. From the Apache Spark version 2.0 onwards, DataFrames have been the principal API in Spark. DataFrame's syntax is more instinctive than of RDD's, but their functionality doesn't differ. RDDs are part of the low-level API and the DataFrames are part of the Structured APIs. Similar to an RDD, a DataFrame is an unchangeable distributed compilation of data. Different from an RDD, in a DataFrame, data is formed into named columns. The RDD functionally and visually looks like to the Pandas in Python and R DataFrames. It is also comparable to an Excel Spreadsheet. It is possible to use them to manipulate, explore, and import the data. Additionally, SQL queries might be used within Spark syntax.

FIFA 20 complete player dataset[14] is used for this case study. The datasets (players_20.csv) provided include the players data for the Career Mode from FIFA 15 to FIFA 20. The data allows multiple comparison of the same players across the last 6 version of the videogame. Fig. 5 shows distribution and the average age of the players in each league for FIFA 19.

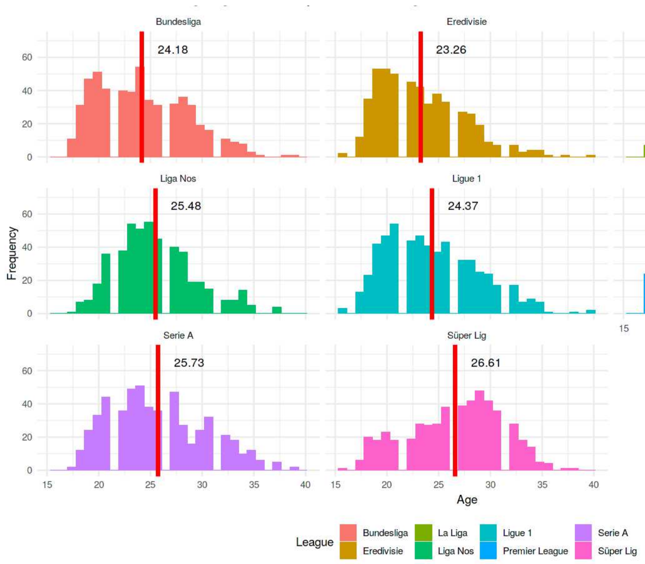


Fig. 5: Distribution and the average age of the players in each league for FIFA 19

Some questions and their SQL statements on "FIFA 20 complete player dataset" is as following:

- Top 10 country with highest mean wage

```
SELECT nationality, AVG(wage_eur), AVG(overall) FROM fifa
GROUP BY nationality ORDER BY AVG(wage_eur) DESC limit 10
```

- Age vs overall rating vs wage

```
SELECT age, AVG(wage_eur), AVG(overall) FROM fifa
GROUP BY age ORDER BY AVG(overall) DESC
```

- Club vs potential top 10

```
SELECT club, AVG(potential) FROM fifa
GROUP BY club ORDER BY AVG(potential) DESC limit 10
```

- Weak foot count

```
SELECT weak_foot, Count(weak_foot) FROM fifa
GROUP BY weak_foot ORDER BY weak_foot ASC
```

[14] https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset

More questions and their statements are available at GitHub repo. Putting these queries into jar (sql.jar) and then copying to an image containing java on the docker named 'sql', the following code snippet in Fig. 6 is run. We initialize MaRe by passing it a player dataset that was previously loaded as an RDD (rddPlayer). We implement the SQL statements' run using the map primitive. We set input and output mount points as text files then we specify a Docker image as sql. Finally, we specify the sql command. As seen, existing other serial tools can be run in MapReduce fashion.

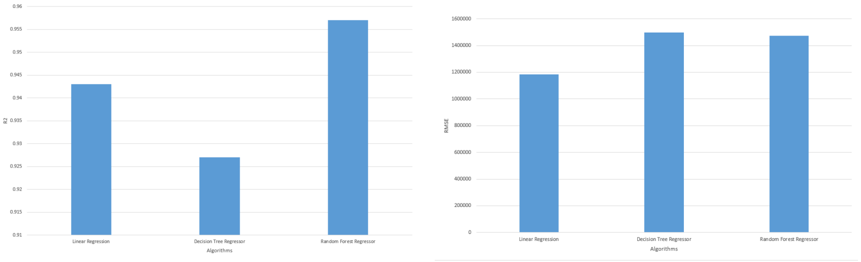
```
val rddPlayer = sc.textFile(path="players_20.csv")
val res = new MaRe(rddPlayer)
    .map(
        inputMountPoint = TextFile("\input"),
        outputMountPoint = TextFile("\output"),
        imageName = "sql",
        command = "java -jar sql.jar > out")
    .rddPlayer.collect()
res.foreach(println(_))
```

Fig. 6: Virtual screening of structured analysis in MaRe

5.2.2 Case Study 2: Machine learning practices on different sport datasets with Spark MLlib

A powerful analytics library and Spark MLlib [61], a built-in general-purpose machine learning framework, are the key features contributing to the use of Spark. It is very popular among data scientists due to its simplicity, language compatibility, scalability, performance, and easy integration with other tools. Thanks to it, data scientists can skip the infrastructure and configuration complexities and solely focus on the data-related tasks. Spark MLlib comes with several optimized machine learning algorithms (e.g., regression, classification, clustering, filtering, collaborative) and provides the flexibility to customize the algorithms for special use cases.

We concern regression, clustering, and classification on different sport datasets. Same FIFA 20 complete player dataset is used for regression purpose. Regression process includes following sub-steps: (i) separating features into categorical and numeric ones, (ii) converting categorical features to numeric values with StringIndexer, (iii) merging dataframes, (iv) vectorizing these merged features, (v) setting training and testing data, (vi) testimonial estimation with different regression models such as linear regressor, decision tree regressor, and random forest regressor, and (vi) measuring their performances with R^2 and Root Mean Square Error (RMSE) metrics. Fig. 7 shows performance results.



(a) R^2

(b) RMSE

Fig. 7: Performance results for different regression algorithms on FIFA 20 complete player dataset

Association of Tennis Professionals (ATP) Matches dataset[15] is used for classification task. In these datasets there are individual csv files for ATP tournament from 2000 to 2017. Fig. 8 shows player's performance of their careers. We concern binary classification (prediction whether a player will beat the match or not) problem here. Classification process includes following substeps: (i) converting categorical features to numeric values with StringIndexer, (ii) target label assigning as 0 or 1, (iii) vectorizing features, (iv) setting training and testing data, (v) fitting different classification models such as logistic regression, decision tree classifier, and random forest classifier, and (vi) measuring their performances with precision, recall, F1-score metrics. Fig. 9 shows area under precision-recall and ROC curves for Logistic regression model. Table 2 shows classification performance results on ATP Matches dataset.

Table 2: Performance results for different classification algorithms on ATP Matches dataset

Classification approach	Precision	Recall	F1-score
Logistic regression	0.63	0.91	0.73
Decision tree classifier	0.66	0.84	0.71
Random forest classifier	0.62	0.84	0.69

In clustering task, it is aimed to group goalkeepers with similar characteristics by using FIFA 20 complete player dataset. The players in the goalkeeper position are grouped by using the average of the goalkeeper characteristics and the average of overall and potential properties. Silhouette method is used to choose the best k-value. Fig. 10 shows the best k-values for different methods.

After determining the best k-value, following steps are done: (i) fitting different clustering models such as Bisecting K-means, Gaussian Mixture, and

[15] https://www.kaggle.com/gmadevs/atp-matches-dataset

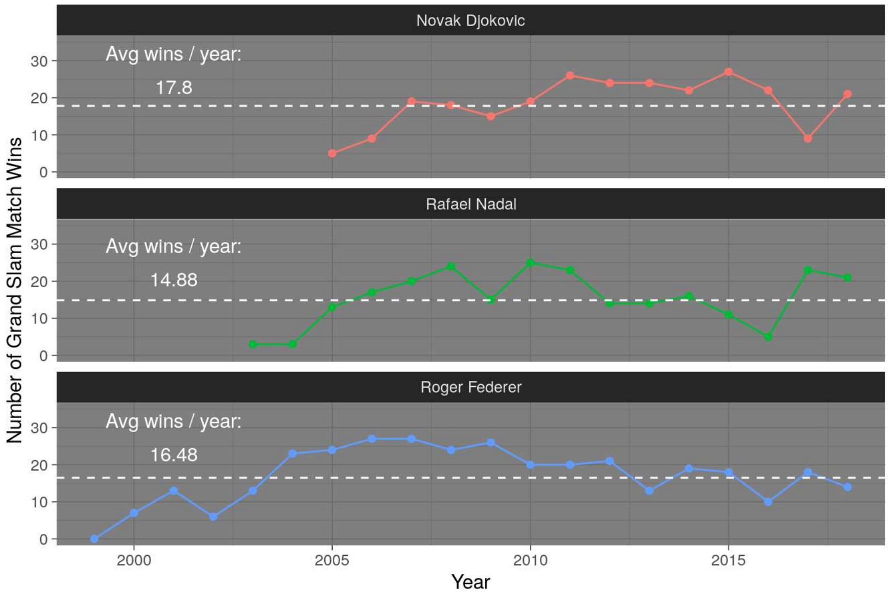
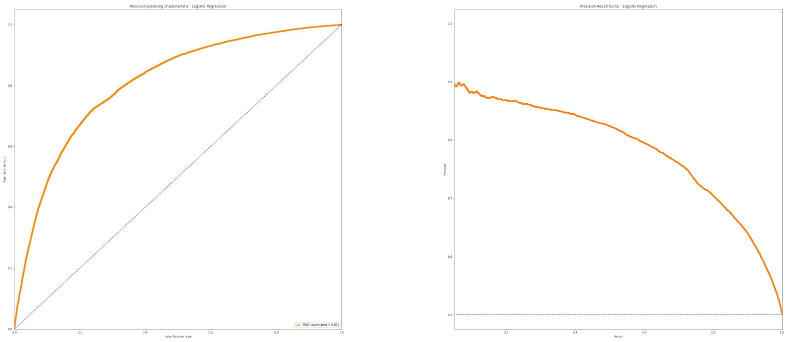


Fig. 8: Grand slam match wins per year

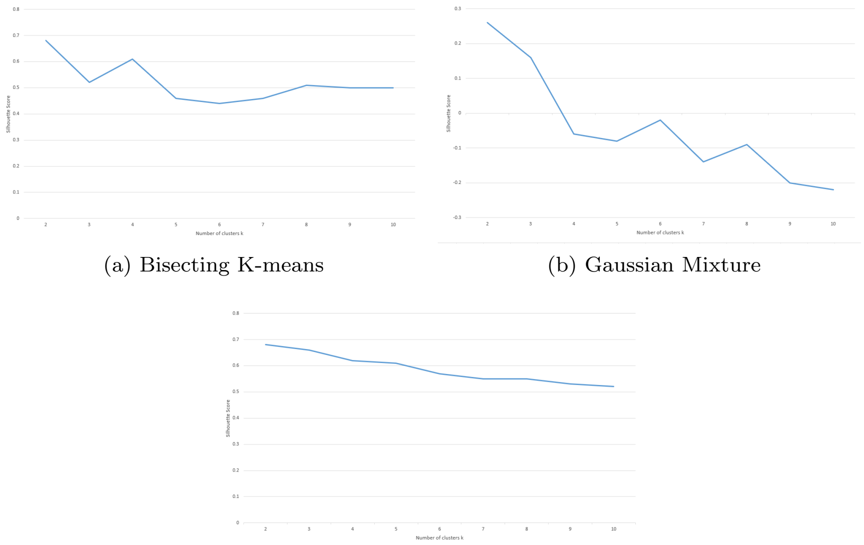


(a) area under precision-recall curve

(b) ROC curve

Fig. 9: Performance results for logistic regression

K-means, and (ii) visual results for these algorithms. Fig. 11 shows clustering results on FIFA 20 complete player dataset. Fig. 12 virtual screening of classification task in MaRe. Other tasks such as regression and clustering are realized in same way.



(a) Bisecting K-means

(b) Gaussian Mixture

(c) K-means

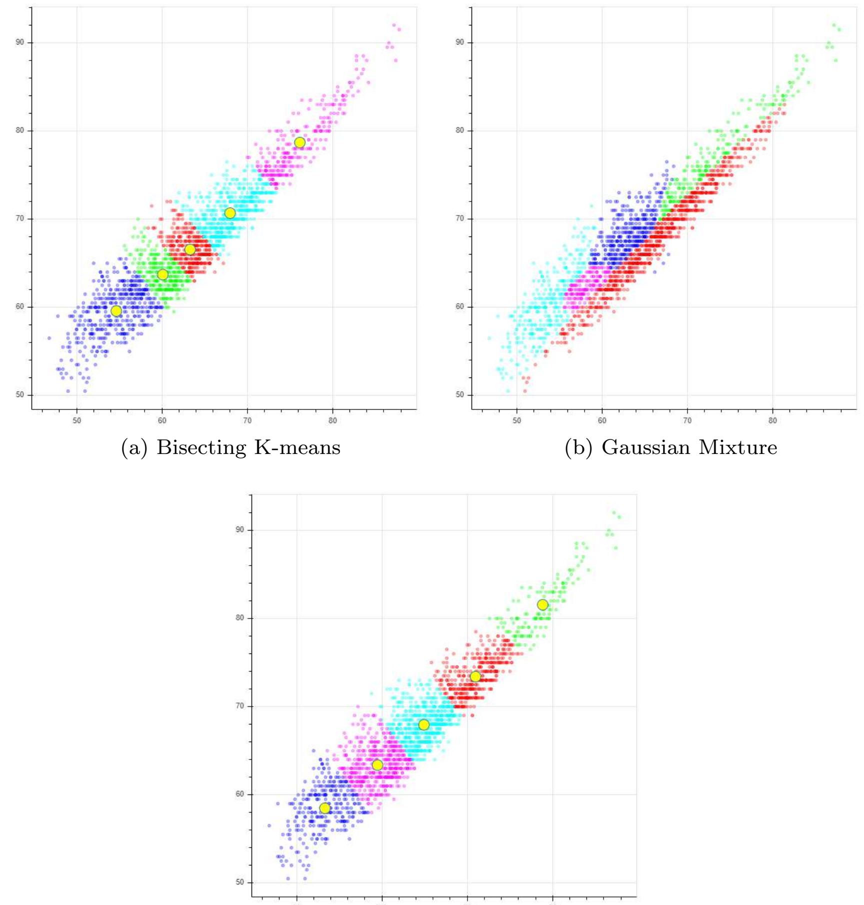
Fig. 10: Determining the optimal number of clusters for different algorithms

5.2.3 Case Study 3: Anomaly detection in multimodal eSports data using Spark Streaming and Apache Pulsar

Apache Spark is used alongside Hadoop for more powerful operations on data. Spark Engine comes with an efficient in-memory (RAM) cluster computing data structure called Resilient Distributed Datasets (RDDs). Data aggregation in the system is handled with Spark Streaming which supports both online and offline data streams.

In this paper, Apache Pulsar[16] (version 2.6.2) high performance distributed messaging platform is used for topic-based pub/sub system. While originally created by Yahoo, it has since become apart of the Apache Software Foundation. It is used for gathering and processing different events in near-realtime, for use cases such as reporting, monitoring, marketing and advertising, personalization and fraud detection. For example, at eBay, Pulsar has been used to improve the user experience by analyzing user interactions and behaviors. Pulsar is closely related to Apache Kafka in terms of features and use cases. It offers great scalability for message processing on a large scale, with high throughput and low end-to-end latency. Messages received are stored persistently with the help of Apache BookKeeper, and message delivery is guaranteed between producers and consumers. While Pulsar is not a stream processing framework as the likes of Apache Storm or Spark Streaming, it does provide some light stream processing features with the use of Pulsar Functions.

[16] https://pulsar.apache.org/



(a) Bisecting K-means

(b) Gaussian Mixture

(c) K-means

Fig. 11: Clustering results for different algorithms on FIFA 20 complete player dataset

Electroencephalography (EEG) data in multimodal eSports dataset[17] is used for streaming task [62]. Sensor data is collected from 10 players in 22 matches in League of Legends. In this task, it is aimed to detect anomalies in the sensor data of e-sports players sent via Apache Pulsar during the tournament. The anomaly detection model is created by using all the features in the sensor data with the IsolationForest algorithm [63], and then the anomalies are detected with this model. Model building process includes following substeps: (i) feature selection, (ii) vectorizing features, (iii) setting training and testing data, (iv) fitting the IsolationForest model. Anomaly detection process

[17] https://github.com/asmerdov/eSports_Sensors_Dataset

```
val rddCluster = sc.textFile(path="tennis.csv")
val res = new MaRe(rddCluster)
    .map(
        inputMountPoint = TextFile("\input"),
        outputMountPoint = TextFile("\output"),
        imageName = "mllib",
        command = "java -cp project.jar Classification > output")
    .rddCluster.collect()
res.foreach(println(_))
```

Fig. 12: Virtual screening of classification task in MaRe

in real-time includes following sub-steps: (i) creation Pulsar client, (ii) making Spark Streaming Pulsar Reciever configurations and loading IsolationForest model, (iii) converting data received in batch form into a string array, and (iv) converting batch into a vector and combining it in a dataframe and anomaly detection over the model. Fig. 13 virtual screening of streaming task in MaRe.

```
val rddStream = sc.textFile(path="esports.csv")
val res = new MaRe(rddStream)
    .map(
        inputMountPoint = TextFile("\input"),
        outputMountPoint = TextFile("\output"),
        imageName = "anomaly",
        command = "java -jar project.jar > output")
    .rddStream.collect()
res.foreach(println(_))
```

Fig. 13: Virtual screening of streaming task in MaRe

5.2.4 Case Study 4: Football passing networks using Spark GraphX

Spark GraphX [64] extends RDD by introducing graphs and graph-parallel computation capabilities. It provides various graph manipulation operations and graph-based algorithms (i.e. triangle counting, counted components, PageRank). Once the analysis process is done and results are obtained, they can be visualized for better understanding.

StatsBomb Open Data[18] is used for graph-based sports data analysis task. The data is provided as JSON files exported from the StatsBomb Data API. Here, we use events data. Events for each match are stored in events as json documents.In this section, we focus on football passing networks using Spark

[18] https://github.com/statsbomb/open-data

GraphX. The passing networks are based on a (generally basic) approach to the graphs theory or analysis, where it is considered the existence of: 1) individual entities (nodes or vertices) which belong to a population or specific group, and 2) the connections between them (edges) in terms an interaction to measure. So, if we translate this to football, the nodes are the players of a same team and the edges are the passes between them [65, 66].

Passing networks are created as following: (i) creating nodes from player who do the pass and player who receive the pass, (ii) creating edges from nodes in a pass relationship, and (iii) graph construction using nodes and edges. When we define the data visualization mapping these are the most frequent considerations: (i) Nodes position- Mean player location when they do and/or receive a pass, (ii) nodes size- variable size depending on amount of passes, (iii) edges color- colored by amount of passes between specific two nodes (0-9, 10-19, 20-29, 30+), (iv) edges direction- this detail is ommited, (v) player ID- text (surname) close to them. Fig. 14 shows the Barcelona's passing network against Deportivo. Fig. 15 virtual screening of graph-based analysis task in MaRe.

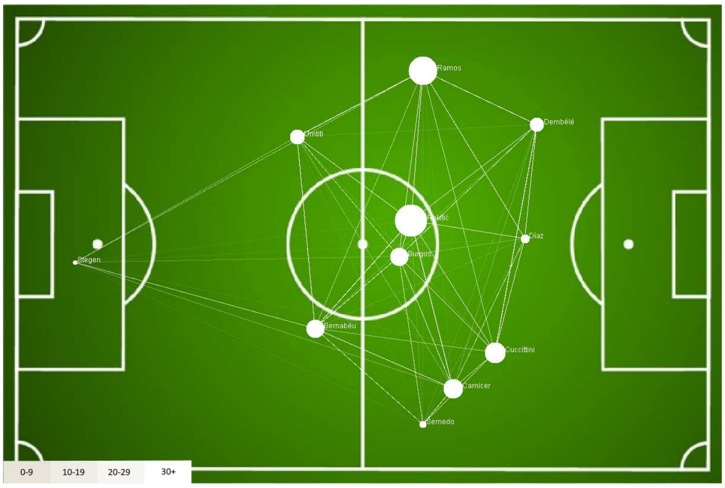


Fig. 14: Barcelona's passing network against Deportivo

6 Conclusions

This section summarizes the main contributions and conclusions of this paper. It also offers some general "lessons learned" from the perspective of both sports data analytics and big data and provides possible directions for future work.

```
val rddGraph = sc.textFile(path="statsbomb_event.csv")
val avg_pos = new MaRe(rddGraph).map(...).rdd
val netw = new MaRe(rddGraph).map(...).rdd
val avg_list = avg_pos.map {...}.collect().toList
val netw = netw.map {...}.collect().toList

Draw.network = netw_list
Draw.data = avg_list
```

Fig. 15: Virtual screening of graph-based analysis task in MaRe

6.1 Summary

The aim of this paper was to show how to analyze different sports data using many approaches from the research field of big data and distributed systems. Towards this end, we grasped a number of shortcomings in the existing literature and made contributions in two main areas of sports analytics: (1) We present a big data architecture based on Docker containers in Apache Spark for sports data analytics pipelines. We gave an open source architecture that introduces support for Docker containers in Apache Spark. (2) We demonstrated the architecture on four data-intensive case studies including structured analysis, streaming, machine learning methods, and graph-based analysis in sport analytics, showing ease of use.

6.2 Lessons learned

In this sub-section, we studied how the research fields of sports data analytics, distrubuted systems and big data can be combined and what these research fields have to offer to each other. We summarized some general lessons learned and advice for respectively sports data analytics practitioners and big data researchers.

Academic awareness: It has been suggested that there is lot of doubt in the world of sports about the real value of business intelligence and analytics tools. The sports analytics utilisation level and practice is relatively neglected in the academic literature. This paper tested and addressed these ideas and contributed to the existing academic literature.

Reproducible research: Reproducibility is the minimum attainable standard for assessing scientific claims. To fulfill this, researchers are required to make both their data and computer code available to their peers. This, however, still falls short of full replication since independently collected data is not used. Nevertheless, this standard allows an assessment to some degree by verifying the original data and codes. Also, repository analysis and data search are important mechanisms in the context of reproducibility.

Containerized sports data processing: By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, user can significantly reduce the delay between writing code and running it in production.

Different types of data analytics: We demonstrated the architecture on 4 data-intensive case studies including structured analysis, streaming, machine learning methods, and graph-based analysis in sport analytics, showing ease of use.

Big data: The field of big data and artificial intelligence offers numerous techniques that can be used to answer important questions in sports analytics to extract information, knowledge, wisdom, and decision from raw data.

We discuss five lessons learned in this paper from the perspective of big data and distributed systems.

Domain knowledge: Including domain knowledge may improve the performance of big data and artificial intelligence models.

Interpretability: Ultimately, experts are interested in turning the findings arising from analytics into practice. To facilitate this, it is to report these findings visually, such as (interactive) drawings, graphs, and maps.

Ground truth data acquisition: In some sport branches, real-world data often lacks ground truth labels. These maybe hard to obtain or simply not exist.

6.3 Future work

The contributions of this paper provided containerized sports data analytics. Nevertheless, there are still many open questions and challenges in the field. This section presents several possible directions for future work. (i) Data privacy is a chief concern (buying fan data and types of data and how data is analysed). They transcend many industries and is not unique to sport. Therefore, refining this data so that it is ready for fan consumption is a mountainous task. (ii) Data analytics in sport is hugely important. As established, clear data is crucial in helping to improve stadium services. (iii) A sport-specific platform that brings together rights holders, sponsors and other stakeholders can be created. (iv) Personalized sport agility training systems can be created using sports nutrition, exercise drills, player activities, tactics, techniques via big data analytics' capabilities.

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

Human and animal participants

This paper does not contain any studies with human participants or animals performed by any of the authors.

Informed consent

The authors declare that the paper is written by ourselves.

Declarations

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Data and code availability

The data and code underlying this article are available in Github, at `https://github.com/yavuzozguven/Dockerized-Sport-Data-Analytics`.

Author Contributions

S.E. devised the project, the main conceptual ideas and proof outline. Y.M.Ö. worked out almost all of the technical details, and performed the experiments. U.G. contributed to the analysis of the results and to the writing of the manuscript. All authors wrote the article.

References

[1] Rajitha Minusha Silva. “Sports analytics”. PhD thesis. Science: Statistics and Actuarial Science, 2016.

[2] Benjamin Alamar and Vijay Mehrotra. “Beyond ‘Moneyball’: The rapidly evolving world of sports analytics, Part I”. In: *Analytics Magazine* (2011).

[3] Bernard Marr. *Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance*. John Wiley & Sons, 2015.

[4] Thomas A Severini. *Analytic methods in sports: Using mathematics and statistics to understand data from baseball, football, basketball, and other sports*. Crc Press, 2020.

[5] Tim B Swartz. “Where should i publish my sports paper?” In: *The American Statistician* 74.2 (2020), pp. 103–108.

[6] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think.* Houghton Mifflin Harcourt, 2013.

[7] Thomas A Runkler. *Data Analytics.* Springer, 2020.

[8] Andrew S Tanenbaum and Maarten Van Steen. *Distributed systems: principles and paradigms.* Prentice-Hall, 2007.

[9] Sven Groppe. "Emergent models, frameworks, and hardware technologies for Big data analytics". In: *The Journal of Supercomputing* 76.3 (2020), pp. 1800–1827.

[10] Rodolfo Metulini. "Filtering procedures for sensor data in basketball". In: *arXiv preprint arXiv:1806.10412* (2018).

[11] Arno Knobbe et al. "Sports analytics for professional speed skating". In: *Data Mining and Knowledge Discovery* 31.6 (2017), pp. 1872–1902.

[12] Janez Pers, Stanislav Kovacic, and Goran Vuckovic. "Analysis and pattern detection on large amounts of annotated sport motion data using standard SQL". In: *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* IEEE. 2005, pp. 339–344.

[13] Matevž Pustišek et al. "The role of technology for accelerated motor learning in sport". In: *Personal and Ubiquitous Computing* (2019), pp. 1–10.

[14] Thomas von der Grün et al. "A real-time tracking system for football match and training analysis". In: *Microelectronic systems.* Springer, 2011, pp. 199–212.

[15] Lukas Probst et al. "Integrated real-time data stream analysis and sketch-based video retrieval in team sports". In: *2018 IEEE International Conference on Big Data (Big Data).* IEEE. 2018, pp. 548–555.

[16] Giovanni Capobianco et al. "A formal methodology for notational analysis and real-time decision support in sport environment". In: *2018 IEEE International Conference on Big Data (Big Data).* IEEE. 2018, pp. 5305–5307.

[17] Zhu Haiyun and Xu Yizhe. "Sports performance prediction model based on integrated learning algorithm and cloud computing Hadoop platform". In: *Microprocessors and Microsystems* 79 (2020), p. 103322.

[18] Dinesh Jackson Samuel R et al. "Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM". In: *Computer Networks* 151 (2019), pp. 191–200.

[19] Andrew Baerg. "Big data, sport, and the digital divide: Theorizing how athletes might respond to big data monitoring". In: *Journal of Sport and Social Issues* 41.1 (2017), pp. 3–20.

[20] Manuel Stein et al. "How to make sense of team sport data: From acquisition to data modeling and research aspects". In: *Data* 2.1 (2017), p. 2.

[21] Jianjun Luo et al. "Flexible and durable wood-based triboelectric nanogenerators for self-powered sensing in athletic big data analytics". In: *Nature communications* 10.1 (2019), pp. 1–9.

[22] Vili Podgorelec, Špela Pečnik, and Grega Vrbančič. "Classification of Similar Sports Images Using Convolutional Neural Network with Hyper-Parameter Optimization". In: *Applied Sciences* 10.23 (2020), p. 8494.

[23] Anthony C Constantinou, Norman E Fenton, and Martin Neil. "pi-football: A Bayesian network model for forecasting Association Football match outcomes". In: *Knowledge-Based Systems* 36 (2012), pp. 322–339.

[24] Kumash Kapadia et al. "Sport analytics for cricket game results using machine learning: An experimental study". In: *Applied Computing and Informatics* (2020).

[25] Kalanka P Jayalath. "A machine learning approach to analyze ODI cricket predictors". In: *Journal of Sports Analytics* 4.1 (2018), pp. 73–84.

[26] Matthew George Soeryadjaya Kerr. "Applying machine learning to event data in soccer". PhD thesis. Massachusetts Institute of Technology, 2015.

[27] Joel Brooks, Matthew Kerr, and John Guttag. "Using machine learning to draw inferences from pass location data in soccer". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 9.5 (2016), pp. 338–349.

[28] Justin Ehrlich and Shankar Ghimire. "Covid-19 countermeasures, major league baseball, and the home field advantage: Simulating the 2020 season using logit regression and a neural network". In: *F1000Research* 9.414 (2020), p. 414.

[29] Shankar Ghimire, Justin A Ehrlich, and Shane D Sanders. "Measuring individual worker output in a complementary team setting: Does regularized adjusted plus minus isolate individual NBA player contributions?" In: *PloS one* 15.8 (2020), e0237920.

[30] Guillermo Vinué and Irene Epifanio. "Archetypoid analysis for sports analytics". In: *Data Mining and Knowledge Discovery* 31.6 (2017), pp. 1643–1677.

[31] Dominik Sacha et al. "Feature-driven visual analytics of soccer data". In: *2014 IEEE conference on visual analytics science and technology (VAST)*. IEEE. 2014, pp. 13–22.

[32] Glenn Sidle and Hien Tran. "Using multi-class classification methods to predict baseball pitch types". In: *Journal of Sports Analytics* 4.1 (2018), pp. 85–93.

[33] Dani Chu and Tim B Swartz. "Foul accumulation in the NBA". In: *Journal of Quantitative Analysis in Sports* 1.ahead-of-print (2020).

[34] Aleksei Karetnikov. "Application of Data-Driven Analytics on Sport Data from a Professional Bicycle Racing Team". Eindhoven University of Technology, The Netherlands, 2019.

[35] Jordi Duch, Joshua S Waitzman, and Luıs A Nunes Amaral. "Quantifying the performance of individual players in a team activity". In: *PloS one* 5.6 (2010), e10937.

[36] Javier López Pena and Hugo Touchette. "A network theory analysis of football strategies". In: *arXiv preprint arXiv:1206.6904* (2012).

[37] Paolo Cintia, Salvatore Rinzivillo, and Luca Pappalardo. "A network-based approach to evaluate the performance of football teams". In: *Machine learning and data mining for sports analytics workshop, Porto, Portugal.* 2015.

[38] Haitian Zheng, Gene Cheung, and Lu Fang. "Analysis of sports statistics via graph-signal smoothness prior". In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA).* IEEE. 2015, pp. 1071–1076.

[39] Abigail R Roane et al. *Graph-based sports rankings.* Tech. rep. Worcester Polytechnic Institute, 2019.

[40] Markus Brandt and Ulf Brefeld. "Graph-based Approaches for Analyzing Team Interaction on the Example of Soccer." In: *MLSA@ PKDD/ECML.* 2015, pp. 10–17.

[41] Jian Shi and Xin-Yu Tian. "Learning to Rank Sports Teams on a Graph". In: *Applied Sciences* 10.17 (2020), p. 5833.

[42] Yao Wu et al. "Characteristics and optimization of core local network: Big data analysis of football matches". In: *Chaos, Solitons & Fractals* 138 (2020), p. 110136.

[43] Agnese Sbrollini et al. "Sport Database: Cardiorespiratory data acquired through wearable sensors while practicing sports". In: *Data in brief* 27 (2019), p. 104793.

[44] Michael Riegler et al. "Heimdallr: a dataset for sport analysis". In: *Proceedings of the 7th International Conference on Multimedia Systems.* 2016, pp. 1–6.

[45] Antonio Lima, Luca Rossi, and Mirco Musolesi. "Coding together at scale: GitHub as a collaborative social network". In: *Proceedings of the International AAAI Conference on Web and Social Media.* Vol. 8. 1. 2014.

[46] Georgios Gousios. "The GHTorent dataset and tool suite". In: *2013 10th Working Conference on Mining Software Repositories (MSR).* IEEE. 2013, pp. 233–236.

[47] Süleyman Eken. "An exploratory teaching program in big data analysis for undergraduate students". In: *Journal of Ambient Intelligence and Humanized Computing* 11.10 (2020), pp. 4285–4304.

[48] Roger D Peng. "Reproducible research in computational science". In: *Science* 334.6060 (2011), pp. 1226–1227.

[49] Süleyman Eken et al. "A reproducible educational plan to teach mini autonomous race car programming". In: *The International Journal of Electrical Engineering & Education* 57.4 (2020), pp. 340–360.

[50] Andreas Wolke and Gerhard Meixner. "TwoSpot: A Cloud Platform for Scaling Out Web Applications Dynamically". In: *Towards a Service-Based Internet: Third European Conference, ServiceWave 2010, Ghent, Belgium, December 13-15, 2010. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 13–24. ISBN: 978-3-642-17694-4. DOI: 10.

1007/978-3-642-17694-4_2. URL: https://doi.org/10.1007/978-3-642-17694-4_2.

[51] Kubernetes. https://kubernetes.io. Accessed 11-February-2021. 2021.

[52] Benjamin Hindman et al. "Mesos: A platform for fine-grained resource sharing in the data center." In: *NSDI*. Vol. 11. 2011. 2011, pp. 22–22.

[53] Hadoop YARN. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html. Accessed 11-February-2021. 2021.

[54] Matei Zaharia et al. "Apache spark: a unified engine for big data processing". In: *Communications of the ACM* 59.11 (2016), pp. 56–65.

[55] Holden Karau and Rachel Warren. *High performance Spark: best practices for scaling and optimizing Apache Spark.* " O'Reilly Media, Inc.", 2017.

[56] GitHub. *Apache Spark Contributors*. https://github.com/apache/spark. Accessed 11-February-2021. 2021.

[57] Apache Foundation. *Spark Overview*. https://spark.apache.org/docs/latest/index.html. Accessed 21-February-2021. 2021.

[58] Carl Boettiger. "An introduction to Docker for reproducible research". In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pp. 71–79.

[59] Charles Anderson. "Docker". In: *IEEE Software* 32.3 (2015), pp. 102–105.

[60] Marco Capuccini et al. "MaRe: Processing Big Data with application containers on Apache Spark". In: *GigaScience* 9.5 (2020), giaa042.

[61] Xiangrui Meng et al. "Mllib: Machine learning in apache spark". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1235–1241.

[62] Anton Smerdov et al. "Collection and Validation of Psycophysiological Data from Professional and Amateur Players: a Multimodal eSports Dataset". In: *arXiv preprint arXiv:2011.00958* (2020).

[63] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest". In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[64] Reynold S Xin et al. "Graphx: A resilient distributed graph system on spark". In: *First international workshop on graph data management experiences and systems*. 2013, pp. 1–6.

[65] Bruno Gonçalves et al. "Exploring team passing networks and player movement dynamics in youth association football". In: *PloS one* 12.1 (2017), e0171156.

[66] Javier M Buldú et al. "Using network science to analyse football passing networks: Dynamics, space, time, and the multilayer nature of the game". In: *Frontiers in psychology* 9 (2018), p. 1900.