Computer Science and Information Systems 19(2):957–978 https://doi.org/10.2298/CSIS2201180100

A Dockerized Big Data Architecture for Sports Analytics

Yavuz Melih Özgüven[1], Utku Gönener[2], and Süleyman Eken[3]

[1] Kocaeli University, Department of Computer Engineering
Izmit 41001, Turkey
yavuzozguven@hotmail.com
[2] Kocaeli University, Faculty of Sports Sciences
Izmit 41001, Turkey
utku.gonener@kocaeli.edu.tr
[3] Kocaeli University, Department of Information Systems Engineering
Izmit 41001, Turkey
suleyman.eken@kocaeli.edu.tr

Abstract. The big data revolution has had an impact on sports analytics as well. Many large corporations have begun to see the financial benefits of integrating sports analytics with big data. When we rely on central processing systems to aggregate and analyze large amounts of sport data from many sources, we compromise the accuracy and timeliness of the data. As a response to these issues, distributed systems come to the rescue, and the MapReduce paradigm holds promise for large-scale data analytics. We describe a big data architecture based on Docker containers with Apache Spark in this paper. We evaluate the architecture on four data-intensive case studies in sport analytics including structured analysis, streaming, machine learning approaches, and graph-based analysis.

Keywords: big data, sports analytics, containers, wearable devices, IoT, reproducible research.

1. Introduction

Decision making in sports based on the information acquired by observation has changed with technological advances. Sports analytics has been more popular in recent days [53]. Sports analytics has the concept of using sports data to create valuable statistics for analysis the need for proper data models is present [50]. There are different approaches on how to treat sports data in combination with data analytics to create statistics and other beneficial information. The big collection of sports data then benefits the analysis and decision making from the sports games [34]. 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.

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 [35]. 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 [47]. When we rely on central processing systems to aggregate and analyze large amounts of sports data from many sources, we compromise the accuracy and timeliness of the data. As a result, we must apply distributed and parallel computing technology to sports analytics study.

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 illustrate how to apply the architecture in four data-intensive applications in sport analytics, including structured analysis, streaming, machine learning approaches, and graph-based analysis.

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

This subsection summarizes structured sport data analysis in the literature. 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 [37] 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. [29] 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. [40] also used standard SQL to analyze large volumes of annotated sport motion data. Their goal was to detect particular types of play, activities, and predefined situations automatically, as well as generate numerous relevant information.

2.2. Sport data streaming

This subsection summarizes stream based sport data analysis in the literature. A number of low-cost wearable devices and gadgets aimed at sport tracking and monitoring have been launched to the market in recent years. Many major technology trends of other Internet of Things (IoT) solutions are shared by sport tracking and monitoring systems. Due to latency and bandwidth limits, cloud and fog computing principles may be a solution to the challenge of real-time analysis and feedback of these IoT devices.

Pustišek et al. [43] touched on the relevance of technology in sports for motor learning, as well as the features and limits of various sensors utilized for activity signal gathering, communication channels, and ways of communication. They created feedback systems that may be used for a variety of augmented motor learning applications using smart sports equipment. Grün et al. [21] created a system that can track a huge number of high-dynamic objects in real-time inside a pre-defined region of interest, such as during a football game. Probst et al. [42] designed a complete team sports analysis infrastructure. This system can identify collaborative events automatically, create statistics based on a continuous stream of raw locations, show the analysis findings in real time, and then save the analysis results in permanent storage for offline use and intuitive sketch-based video retrieval later. Capobianco et al. [7] 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 [22] developed an integrated and extensive learning based Hadoop platform for forecasting game outcomes. Dinesh et al. [44] presented a system for detecting violence in a football stadium in real time. In the Spark environment, the HOG function was utilized to extract features from video frames. Proposed system alerts the security forces. Baerg [2] analyzed the athlete's performance with big data. Stein et al. [55] first discussed how to evaluate team sport data in general before proposing a multi-faceted approach that included pattern recognition, context-aware analysis, and visual explanation. Luo et al. [33] reported a wood-based triboelectric nanogenerator that is flexible and robust for self-powered sensing in sports 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. [41] created a new image dataset of four comparable sports (American football, rugby, soccer, and field hockey) and used CNN transfer learning with Hyper-Parameter Optimization (HPO) to categorize the images. 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. [11] developed probabilistic models based on possession rates and other historical statistics of various teams to predict the outcome of matches. Kapadia et al. [25] used machine learning techniques to solve the same problem but for the cricket world in the Indian Premier League (IPL). Jayalath [24] considered the popular logistic regression model to study the significance of one-day international (ODI) cricket predictors. Kerr [28] presented three experiments in his thesis. In the first experiment, three models were created utilizing various attributes to predict which side would win a particular game without knowing the score. Several classifiers were employed in the second experiment to predict which team created the sequence of

ball-events that happened during a game. And in the last one, he predicted which team attempted a given set of passes. Brooks et al. [5] focused on examining characteristics of passing in soccer and introduced two methods for obtaining insights from that. Ehrlich and Ghimire [13] 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. [17] 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 used a two-stage least square (2-SLS) approach to test the robustness of a series of linear fixed effects regression models to explain variance in Real Plus-Minus (RPM) between player seasons. In order to identify essential features and generate interpretable models for sport data analytics in professional speed skating, Knobbe et al. [29] employed linear modeling and subgroup discovery. Vinué and Epifanio [56] developed a useful mathematical tool based on archetypoid analysis (ADA) to evaluate the worth of players and clubs in a league by analyzing their performance. In three cases, the value of archetypoids in sports was demonstrated using data from basketball and soccer. Janetzko et al. [48] created a method that allows users to interactively examine and evaluate movement characteristics and game events in high-frequency position-based soccer data at various degrees of detail. Sidle and Tran [52] applied multi-class classification methods to the problem of predicting baseball live pitch types. While Chu and Swartz [9] proposed a Bayesian inference system with parametric models to analyze fouling time distributions. Karetnikov [27] 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. [12] and Pena and Touchette [38] examined weighted pass graphs. Players were represented by nodes, passes by edges, and the efficiency of passes by weights. Cintia et al. [10] analyzed a passing network using network centrality metrics from two perspectives: passes between players and passes between pitch zones. Zheng et al. [61] predicted game outcomes from available sports statistics using a graph signal processing (GSP) perspective. Roane et al. [46] 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 [4] presented a graph-based approach to analyze player interaction in team sports. Shi and Tian [51] used a game graph from the perspective of Bayesian correction with game results to build a generalized PageRank model for sports. Wu et al. [58] 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 a web application that allows users to engage with data visualizations of soccer passing networks.

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

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.

Every day, developers find new ways to put containerization to work to solve their challenges. There is no shortage of ways to use containerization, and every application will likely produce unique benefits. Here are some of the most common benefits of our proposed containerized big data architecture: (i) Portability between different platforms and clouds—it's truly write once, run anywhere. (ii) Efficiency through using far fewer resources than VMs and delivering higher utilization of compute resources. (iii) Agility that allows developers to integrate with their existing DevOps environment. (iv) Improved security by isolating applications from the host system and from each other. (v) Faster app start-up and easier scaling.

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, armchairanalysis.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 Base-

[5] https://statsbomb.com/academy/
[6] https://openfootball.github.io/
[7] http://nflsavant.com/

ball Database[8] for baseball, FiveThirtyEight[9] for others, Sport Database [49] for cardiorespiratory data, Heimdallr [45] 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 [31]. In contrast with competitors (e.g., SourceForge[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 [20].

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 [14]. 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 [39][15].

[8] http://www.seanlahman.com/baseball-archive/
[9] https://data.fivethirtyeight.com/
[10] https://github.com/
[11] https://sourceforge.net/
[12] https://www.assembla.com/
[13] https://github.com/feinoujc/gh-search-cli

4. Containerized Architecture

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

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 [57]. 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 [30], Mesos [23], and Hadoop YARN).

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 [60]. An illustration of this architecture can be seen in Fig. 1.

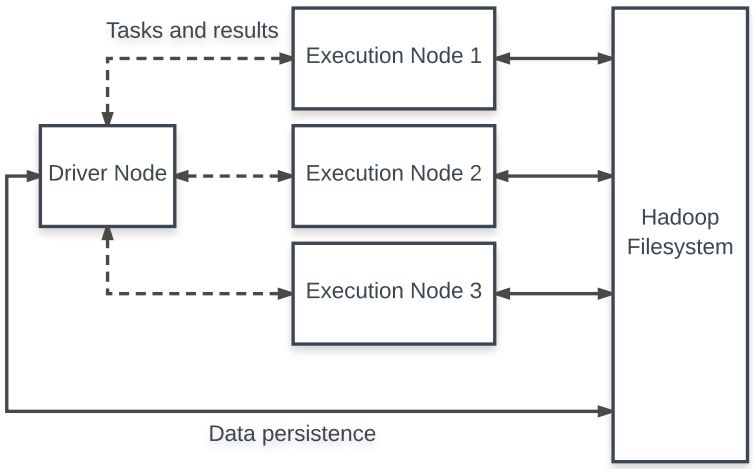


Fig. 1. Simplified diagram of a typical Spark cluster

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 [26]. 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 [60], with over 1600 contributors [18] 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 [16], 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 [3], 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.

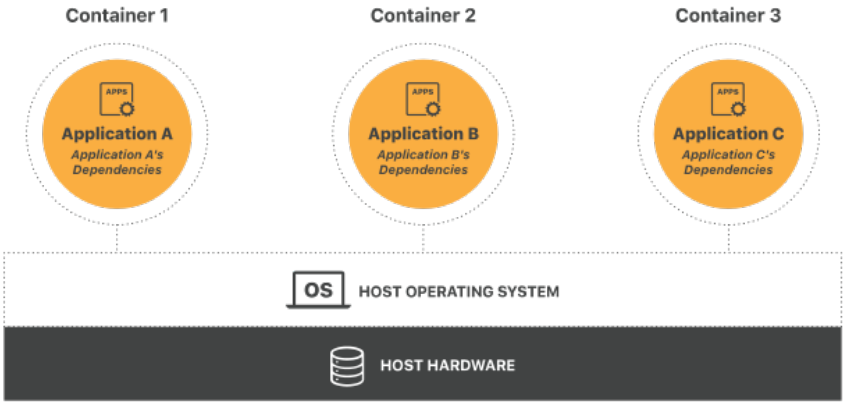


Fig. 4. Container architecture

Docker is one of the most popular software containerization platforms. Developers, operators, and enterprises use it for the previously mentioned merits. Users may substantially shorten the time between developing code and executing it in production by utilizing Docker's techniques for shipping, testing, and deploying code rapidly. Because of the isolation and security, users may operate several containers on a single host [1].

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 [8]. It enables scalable data-intensive processing.

5.2. Case Studies

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.

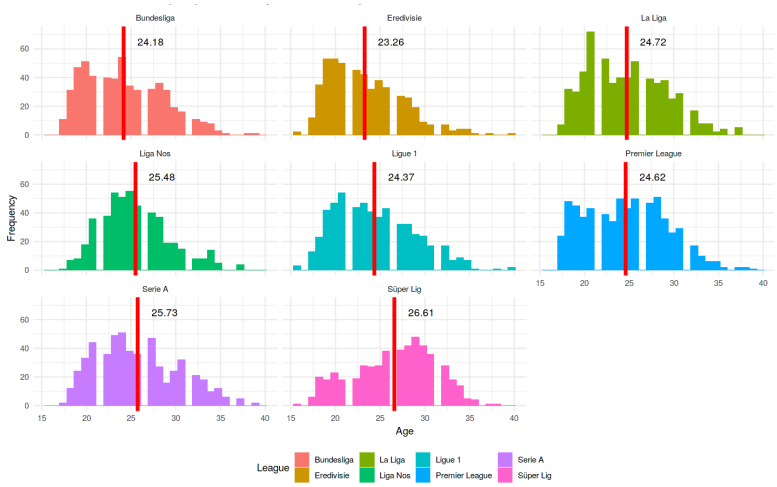


Fig. 5. Distribution and the average age of the players in each league for FIFA 19

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. Players' data for Career Mode from FIFA 15 to FIFA 20 is included in the databases. The information enables for numerous comparisons of the same players over the videogame's past six versions. Fig. 5 shows 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
```

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.

Case Study 2: Machine learning practices on different sport datasets with Spark MLlib A powerful analytics library and Spark MLlib [36], a built-in general-purpose machine learning framework, are the key features contributing to the use of Spark. Because of its simplicity, language compatibility, scalability, performance, and ease of interaction with other tools, it is highly popular among data scientists. It allows data scientists to focus entirely on data-related activities, bypassing the complexity of infrastructure and

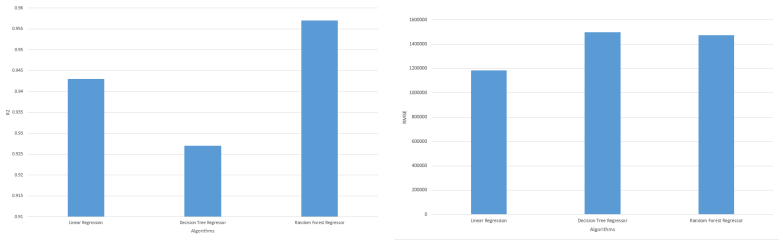
[14] https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset

```
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

setup. Spark MLlib includes a set of efficient machine learning methods (such as regression, classification, clustering, filtering, and collaboration) as well as the ability to adapt the algorithms for specific 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

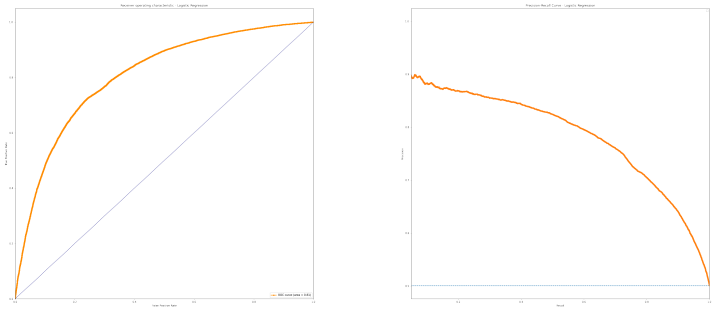
Fig. 7. Performance results for different regression algorithms on FIFA 20 complete player dataset

Association of Tennis Professionals (ATP) Matches dataset[15] is used for classification task. Individual csv files for ATP tournaments from 2000 to 2017 may be found in these databases. 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 sub-steps: (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

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

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.

Fig. 8. Grand slam match wins per year



(a) area under precision-recall curve **(b)** ROC curve

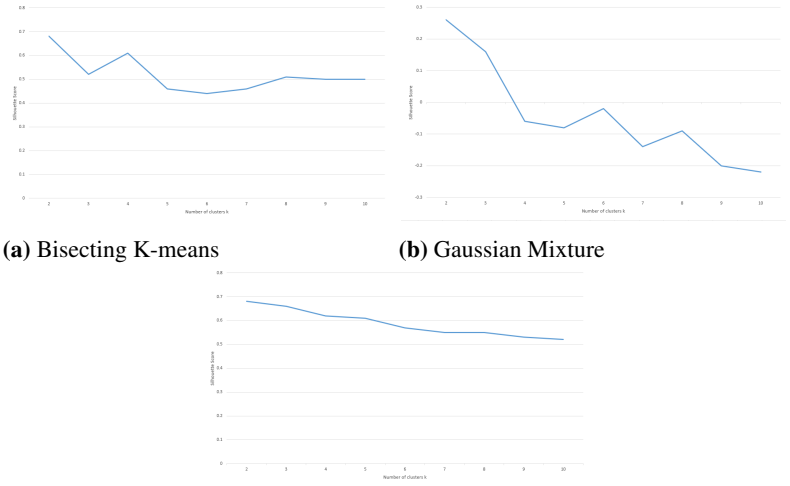
Fig. 9. Performance results for logistic regression

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

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

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.



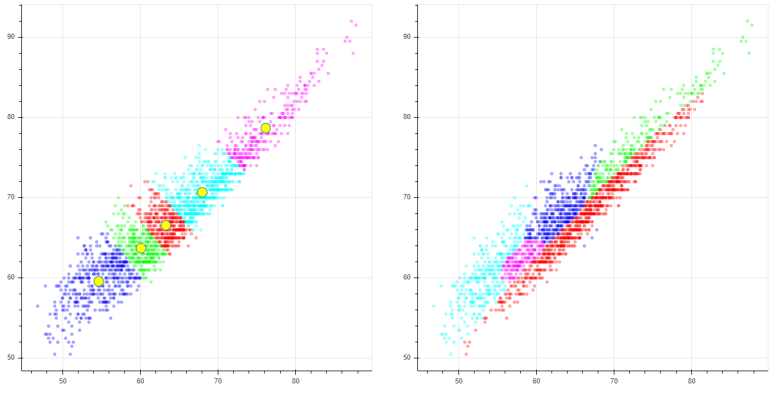
(a) Bisecting K-means **(b)** Gaussian Mixture

(c) K-means

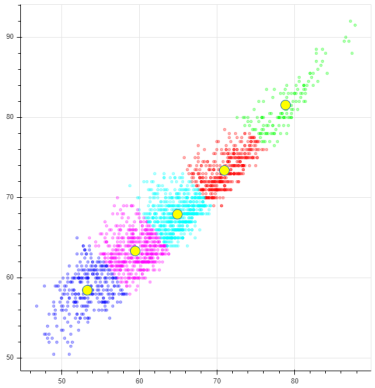
Fig. 10. Determining the optimal number of clusters for different algorithms

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

Case Study 3: Anomaly detection in multimodal eSports data using Spark Streaming and Apache Pulsar For more sophisticated data processing, Apache Spark is utilized in conjunction with Hadoop. Resilient Distributed Datasets is an efficient in-memory (RAM) cluster computing data format included with Spark Engine (RDDs). The system's data aggregation is done by Spark Streaming, which can handle both online and offline data streams.



(a) Bisecting K-means **(b)** Gaussian Mixture



(c) K-means

Fig. 11. Clustering results for different algorithms on FIFA 20 complete player 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 this paper, The high-performance distributed messaging platform Apache Pulsar[16] (version 2.6.2) is utilized for the topic-based pub/sub system. It was initially developed by Yahoo and is now part of the Apache Software Foundation. It's used for reporting,

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

monitoring, marketing and advertising, customization, and fraud detection, and it's utilized for gathering and processing diverse events in near-realtime. Pulsar has been used to enhance the user experience at eBay, for example, by monitoring user interactions and behaviors. In terms of features and use cases, Pulsar is quite similar to Apache Kafka. It has excellent scalability for large-scale message processing, with high throughput and low end-to-end latency. Messages received are continuously saved with Apache BookKeeper, and message transmission between producers and consumers is assured. While Pulsar is not a full-fledged stream processing framework like Apache Storm or Spark Streaming, it does offer some light stream processing capabilities via Pulsar Functions.

Electroencephalography (EEG) data in multimodal eSports dataset[17] is used for streaming task [54]. 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 [32], 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 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

Case Study 4: Football passing networks using Spark GraphX Spark GraphX [59] 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

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

data. Events for each match are stored in events as json documents.In this section, we focus on football passing networks using Spark 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 [19][6].

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.

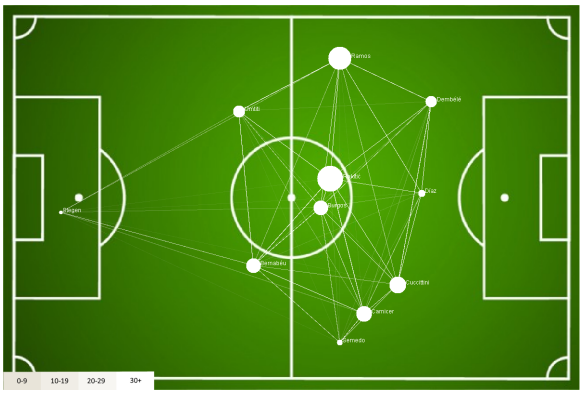


Fig. 14. Barcelona's passing network against Deportivo

```
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. Conclusions

The key contributions and findings of this work are summarized in this section. It also includes some broad "lessons learned" from the perspectives of both sports data analytics and big data, as well as potential future research topics.

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. To that purpose, we identified a number of flaws in the current literature and contributed to two major areas of sports analytics: (1) For sports data analytics pipelines, we offer a big data architecture based on Docker containers in Apache Spark. We presented an open source architecture that adds Docker container functionality to Apache Spark. (2) We presented the architecture in four data-intensive case studies in sports analytics, including structured analysis, streaming, machine learning techniques, and graph-based analysis.

6.2. Lessons learned

We looked at how the study disciplines of sports data analytics, distributed systems, and big data may be integrated, as well as what these research topics have to offer each other, in this part. We compiled a list of general findings and recommendations for 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. Researchers must make both their data and computer code open to their colleagues in order to do this. 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:** Users may substantially shorten the time between developing code and executing it in production by utilizing Docker's techniques for shipping, testing, and deploying code rapidly. **Different types of data analytics:** We presented the architecture in four data-intensive case studies in sport analytics, including structured analysis, streaming, machine learning approaches, and graph-based analysis. **Big data:** The fields of big data and artificial intelligence provide a variety of approaches for extracting information, knowledge, wisdom, and judgment from raw data, which may be utilized to answer critical issues in sports analytics.

From the view of big data and distributed systems, we explore five lessons learned. **Domain knowledge:** Domain expertise might help big data and artificial intelligence models perform better. **Interpretability:** Experts are interested in putting the findings of analytics into effect in the end. It is to communicate these findings aesthetically, such as (interactive) drawings, graphs, and maps, to make this easier. **Ground truth data acquisition:** Real-world data in various sports sometimes lacks ground truth labeling. These may be difficult to get by or just do not exist.

6.3. Future work

Containerized sports data analytics is the paper's contribution. However, there are several unanswered issues and problems in the subject. This section outlines a number of potential future research directions. (i) Data privacy is a chief concern (buying fan data and types of data and how data is analysed). They apply to a wide range of sectors and are not limited to sports. As a result, it's a huge work to refine this data so that it's fit for fan consumption. (ii)In sports, data analytics is critical. As previously said, accurate data is critical in aiding in the improvement of 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.

Software. Code underlying this article are available in Github, at https://github.com/yavuzozguven/Dockerized-Sport-Data-Analytics.

References

1. Anderson, C.: Docker. IEEE Software 32(3), 102–105 (2015)
2. Baerg, A.: Big data, sport, and the digital divide: Theorizing how athletes might respond to big data monitoring. Journal of Sport and Social Issues 41(1), 3–20 (2017)
3. Boettiger, C.: An introduction to docker for reproducible research. ACM SIGOPS Operating Systems Review 49(1), 71–79 (2015)
4. Brandt, M., Brefeld, U.: Graph-based approaches for analyzing team interaction on the example of soccer. In: MLSA@ PKDD/ECML. pp. 10–17 (2015)
5. Brooks, J., Kerr, M., Guttag, J.: 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)
6. Buldú, J.M., Busquets, J., Martínez, J.H., Herrera-Diestra, J.L., Echegoyen, I., Galeano, J., Luque, J.: Using network science to analyse football passing networks: Dynamics, space, time, and the multilayer nature of the game. Frontiers in psychology 9, 1900 (2018)
7. Capobianco, G., Di Giacomo, U., Mercaldo, F., Santone, A.: A formal methodology for notational analysis and real-time decision support in sport environment. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 5305–5307. IEEE (2018)
8. Capuccini, M., Dahlö, M., Toor, S., Spjuth, O.: Mare: Processing big data with application containers on apache spark. GigaScience 9(5), giaa042 (2020)
9. Chu, D., Swartz, T.B.: Foul accumulation in the nba. Journal of Quantitative Analysis in Sports 1(ahead-of-print) (2020)
10. Cintia, P., Rinzivillo, S., Pappalardo, L.: A network-based approach to evaluate the performance of football teams. In: Machine learning and data mining for sports analytics workshop, Porto, Portugal (2015)
11. Constantinou, A.C., Fenton, N.E., Neil, M.: pi-football: A bayesian network model for forecasting association football match outcomes. Knowledge-Based Systems 36, 322–339 (2012)
12. Duch, J., Waitzman, J.S., Amaral, L.A.N.: Quantifying the performance of individual players in a team activity. PloS one 5(6), e10937 (2010)
13. Ehrlich, J., Ghimire, S.: Covid-19 countermeasures, major league baseball, and the home field advantage: Simulating the 2020 season using logit regression and a neural network. F1000Research 9(414), 414 (2020)
14. Eken, S.: An exploratory teaching program in big data analysis for undergraduate students. Journal of Ambient Intelligence and Humanized Computing 11(10), 4285–4304 (2020)

15. Eken, S., Şara, M., Satılmış, Y., Karslı, M., Tufan, M.F., Menhour, H., Sayar, A.: A reproducible educational plan to teach mini autonomous race car programming. The International Journal of Electrical Engineering & Education 57(4), 340–360 (2020)
16. Foundation, A.: Spark Overview. https://spark.apache.org/docs/latest/index.html (2021), accessed 21-February-2021
17. Ghimire, S., Ehrlich, J.A., Sanders, S.D.: Measuring individual worker output in a complementary team setting: Does regularized adjusted plus minus isolate individual nba player contributions? PloS one 15(8), e0237920 (2020)
18. GitHub: Apache Spark Contributors. https://github.com/apache/spark (2021), accessed 11-February-2021
19. Gonçalves, B., Coutinho, D., Santos, S., Lago-Penas, C., Jiménez, S., Sampaio, J.: Exploring team passing networks and player movement dynamics in youth association football. PloS one 12(1), e0171156 (2017)
20. Gousios, G.: The ghtorent dataset and tool suite. In: 2013 10th Working Conference on Mining Software Repositories (MSR). pp. 233–236. IEEE (2013)
21. von der Grün, T., Franke, N., Wolf, D., Witt, N., Eidloth, A.: A real-time tracking system for football match and training analysis. In: Microelectronic systems, pp. 199–212. Springer (2011)
22. Haiyun, Z., Yizhe, X.: Sports performance prediction model based on integrated learning algorithm and cloud computing hadoop platform. Microprocessors and Microsystems 79, 103322 (2020)
23. Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A.D., Katz, R.H., Shenker, S., Stoica, I.: Mesos: A platform for fine-grained resource sharing in the data center. In: NSDI. vol. 11, pp. 22–22 (2011)
24. Jayalath, K.P.: A machine learning approach to analyze odi cricket predictors. Journal of Sports Analytics 4(1), 73–84 (2018)
25. Kapadia, K., Abdel-Jaber, H., Thabtah, F., Hadi, W.: Sport analytics for cricket game results using machine learning: An experimental study. Applied Computing and Informatics (2020)
26. Karau, H., Warren, R.: High performance Spark: best practices for scaling and optimizing Apache Spark. " O'Reilly Media, Inc." (2017)
27. Karetnikov, A.: Application of data-driven analytics on sport data from a professional bicycle racing team (2019)
28. Kerr, M.G.S.: Applying machine learning to event data in soccer. Ph.D. thesis, Massachusetts Institute of Technology (2015)
29. Knobbe, A., Orie, J., Hofman, N., van der Burgh, B., Cachucho, R.: Sports analytics for professional speed skating. Data Mining and Knowledge Discovery 31(6), 1872–1902 (2017)
30. Kubernetes:
31. Lima, A., Rossi, L., Musolesi, M.: Coding together at scale: Github as a collaborative social network. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 8 (2014)
32. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
33. Luo, J., Wang, Z., Xu, L., Wang, A.C., Han, K., Jiang, T., Lai, Q., Bai, Y., Tang, W., Fan, F.R., et al.: Flexible and durable wood-based triboelectric nanogenerators for self-powered sensing in athletic big data analytics. Nature communications 10(1), 1–9 (2019)
34. Marr, B.: Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance. John Wiley & Sons (2015)
35. Mayer-Schönberger, V., Cukier, K.: Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt (2013)
36. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al.: Mllib: Machine learning in apache spark. The Journal of Machine Learning Research 17(1), 1235–1241 (2016)

37. Metulini, R.: Filtering procedures for sensor data in basketball. arXiv preprint arXiv:1806.10412 (2018)
38. Pena, J.L., Touchette, H.: A network theory analysis of football strategies. arXiv preprint arXiv:1206.6904 (2012)
39. Peng, R.D.: Reproducible research in computational science. Science 334(6060), 1226–1227 (2011)
40. Pers, J., Kovacic, S., Vuckovic, G.: 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. pp. 339–344. IEEE (2005)
41. Podgorelec, V., Pečnik, Š., Vrbančič, G.: Classification of similar sports images using convolutional neural network with hyper-parameter optimization. Applied Sciences 10(23), 8494 (2020)
42. Probst, L., Rauschenbach, F., Schuldt, H., Seidenschwarz, P., Rumo, M.: Integrated real-time data stream analysis and sketch-based video retrieval in team sports. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 548–555. IEEE (2018)
43. Pustišek, M., Wei, Y., Sun, Y., Umek, A., Kos, A.: The role of technology for accelerated motor learning in sport. Personal and Ubiquitous Computing pp. 1–10 (2019)
44. R, D.J.S., Fenil, E., Manogaran, G., Vivekananda, G., Thanjaivadivel, T., Jeeva, S., Ahilan, A.: Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional lstm. Computer Networks 151, 191–200 (2019)
45. Riegler, M., Dang-Nguyen, D.T., Winther, B., Griwodz, C., Pogorelov, K., Halvorsen, P.: Heimdallr: a dataset for sport analysis. In: Proceedings of the 7th International Conference on Multimedia Systems. pp. 1–6 (2016)
46. Roane, A.R., Ekkaewnumchai, C., McNamara, C.W., Richards, K.: Graph-based sports rankings. Tech. rep., Worcester Polytechnic Institute (2019)
47. Runkler, T.A.: Data Analytics. Springer (2020)
48. Sacha, D., Stein, M., Schreck, T., Keim, D.A., Deussen, O., et al.: Feature-driven visual analytics of soccer data. In: 2014 IEEE conference on visual analytics science and technology (VAST). pp. 13–22. IEEE (2014)
49. Sbrollini, A., Morettini, M., Maranesi, E., Marcantoni, I., Nasim, A., Bevilacqua, R., Riccardi, G.R., Burattini, L.: Sport database: Cardiorespiratory data acquired through wearable sensors while practicing sports. Data in brief 27, 104793 (2019)
50. Severini, T.A.: Analytic methods in sports: Using mathematics and statistics to understand data from baseball, football, basketball, and other sports. Crc Press (2020)
51. Shi, J., Tian, X.Y.: Learning to rank sports teams on a graph. Applied Sciences 10(17), 5833 (2020)
52. Sidle, G., Tran, H.: Using multi-class classification methods to predict baseball pitch types. Journal of Sports Analytics 4(1), 85–93 (2018)
53. Silva, R.M.: Sports analytics. Ph.D. thesis, Science: Statistics and Actuarial Science (2016)
54. Smerdov, A., Zhou, B., Lukowicz, P., Somov, A.: Collection and validation of psycophysiological data from professional and amateur players: a multimodal esports dataset. arXiv preprint arXiv:2011.00958 (2020)
55. Stein, M., Janetzko, H., Seebacher, D., Jäger, A., Nagel, M., Hölsch, J., Kosub, S., Schreck, T., Keim, D.A., Grossniklaus, M.: How to make sense of team sport data: From acquisition to data modeling and research aspects. Data 2(1), 2 (2017)
56. Vinué, G., Epifanio, I.: Archetypoid analysis for sports analytics. Data Mining and Knowledge Discovery 31(6), 1643–1677 (2017)
57. Wolke, A., Meixner, G.: TwoSpot: A Cloud Platform for Scaling Out Web Applications Dynamically, pp. 13–24. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
58. Wu, Y., Xia, Z., Wu, T., Yi, Q., Yu, R., Wang, J.: Characteristics and optimization of core local network: Big data analysis of football matches. Chaos, Solitons & Fractals 138, 110136 (2020)

59. Xin, R.S., Gonzalez, J.E., Franklin, M.J., Stoica, I.: Graphx: A resilient distributed graph system on spark. In: First international workshop on graph data management experiences and systems. pp. 1–6 (2013)
60. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., et al.: Apache spark: a unified engine for big data processing. Communications of the ACM 59(11), 56–65 (2016)
61. Zheng, H., Cheung, G., Fang, L.: Analysis of sports statistics via graph-signal smoothness prior. In: 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). pp. 1071–1076. IEEE (2015)

Yavuz Melih Güven received the BSc degree in Computer Engineering from Kocaeli University in 2021. He is now MSc candidate in Information Systems Engineering at Kocaeli University. He is interested in big data, distributed systems and machine learning.

Utku Gönener received the BSc degree in Electrical and Electronics Engineering from Okan University in 2012. He is now PhD candidate in Sports Sciences at Kocaeli University. Currently, he works as a research assistant at the same university. His interests are exercise sciences and periodization, biomechanics, physiological and performance tests.

Süleyman Eken received his MS degree and PhD degree in Computer Engineering at the Kocaeli University. He was a research assistant at Kocaeli University, Turkey, from 2010 to 2019. Currently, he works as an Associate Professor of Information Systems Engineering, Kocaeli University, Izmit, Turkey. His main research work focuses on distributed systems and big data analysis.

Received: January 18, 2022; Accepted: March 20, 2022.