IoT web-based platform for Athlete's development

Raouf Brahim
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
erb4539@umoncton.ca

Nidhal Hadj Abdallah
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
enh4523@umoncton.ca

Haifa Souifi
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
ehs4973@umoncton.ca

Yassine Bouslimani
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
yassine.bouslimani@umoncton.ca

Mohsen Ghribi
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
mohsen.ghribi@umoncton

Azeddine Kaddouri
Electrical Engineering Department
Universite de Moncton
Moncton, NB, Canada
kaddoua@umoncton

***Abstract*—This paper aims to present a web-based platform development based on the integration of IoT functionalities. This developed platform is built for athlete's workout routines to facilitate their basic training by collecting data during exercises and to improve their abilities during training sessions. The proposed approach allows coaches to monitor training progress in real-time as well as adjust routines and instructions during the sessions. This present work includes methods of sending, processing, storing and then accessing to the data collected by IoT devices during training, in order to visualize statistics and information on player's performance. A customized cloudbased platform capable of analyzing, as an example, ball shots during a handball training was also presented. In addition, this platform will provide coaches with real-time access to player's data and allow monitoring athlete's exercises and managing specific trainings to improve performance results. A comparison of processing time is made for different commercial Cloud platforms widely used such as Google Cloud, Microsoft Azure and Amazon web Services. Furthermore, this study presents an easyto-build cloud platform able to analyze athlete's data collected using IoT devices.**

***Index Terms*—Internet of Things (IoT), Azure, Amazon, AWS, GCP, Cloud, Message Queuing Telemetry Transport (MQTT), REST, SOAP, XML, NoSQL, MongoDB, JSON, WEBSOKET (WS).**

I. Introduction

Sport training is a human-based activity, while the development of a professional player is a very complicated process which requires coaches and qualified human resources [1]. In most team sports, coaches find it difficult to monitor the performance and progress of their players. In a specific training, and in order to increase the performance of the athletes, the coach needs the intervention of connected circuits and the application of appropriate exercises and techniques using new technologies.In [20] and [21] , authors made a comparison between the best Handball and Basketball teams and the worst ones. This comparison was based on their results during the seasons of playing. In fact, for collective games there is always a win and loss percentage. Sometimes, the team which is expected to win lose. Indeed, statistics of matches are important to improve the teams' performance but the most important thing that is related to teams' results is how they prepared their matches and the way they trained. The calculation and the treatment of Handball players' statistics in real time is a process that exists a powerful technological architecture. Cloud computing and Internet of Things (IoT) development allow to process and simplify complex calculations in order to transform training into an interactive process, and track athlete's progression over time based on parameters fixed by coaches. The IoT is becoming the source of information not only for researchers, but also for athletes and Sports Experts. By using IoT devices and cloud platforms in sports, the result or the success rate in training has become a measurable parameter through the collected data. Consequently, the percentage of success of an athlete during his exercise can be calculated. Cloud platforms are likewise designed so as to store and process real-time big data, quickly and efficiently [2].Hence, using these platforms will help to compare the results of a training and performances over the time and to compare the results between different players. Nowadays, there are many Cloud platforms available to the public such as Amazon Web Services (AWS), Azure Cloud, Google Cloud Platform (GCP), among others [3].There were around 5 to 9 billion connected devices in 2015 [4] and this number has increased exponentially to reach 50 billion [5] in 2018. This explains the investments made by companies in IoT platforms to reach more than half a trillion dollars in 2021, compared to 235 billion dollars spent in 2017 [6]. Cloud-IoT platforms provide middleware to connect and manage hardware devices and collected data. The role of this middleware is to manage, analyze, save and respond to network requests. In [7] , the authors proposed a method that consists in sending a number of messages to Clouds Amazon Aws, Google Cloud and Azure Cloud, and then comparing the response time of each according to the number of messages per second (MPS). Message processing in both Google and Amazon platforms is always in the 20 to 35 millisecond range. MPS processing time in Azure platform is more than 120 milliseconds, when the number of messages sent per second is less than 6000. In addition, Google Cloud is faster compared

978-1-6654-2854-5/21/$31.00 ©2021 IEEE

to other platforms except for values between 150 and 750 MPS or Amazon offers the best performance [7].Google's Cloud will be well suited to big data and high-performance computing. But its main limitation is the fact that it does not integrate administrative features and therefore its users have to download additional packages [8].

As reported in the literature [8] , [10] , connecting IoT objects with a Cloud platform is also investigated. Augusto et al. [8] suggested a system based on VR (Virtual Reality) using an innovative mechanism called VR-IoT Environment Synchronization Scheme (VRITESS). This technology allows users to seamlessly use IoT devices to view data provided by sensors in a virtual environment. Message Queuing Telemetry Transport (MQTT) protocol and RESTful API response time performance tests show that MQTT communications are more than 20 times faster than the RESTful API [8]. Furthermore, the authors in [9] , [10] have shown the usefulness of the MQTT protocol. They disclosed that this protocol is one of the most important pillars to connect a machine or sensor by a server and to guarantee end-to-end encryption [9]. In addition, MQTT is the most recommended protocol thanks to the 4 security requirements that are based on mutual authentication, control access, control message security and end-to-end security [10].

More recently, many research studies put emphasize on the topics regarding the problem of Data processing in a Cloud platform [11] , [12].In his work [11],Dagenais presented a comparative study of three programming languages: Java, Go and Node.js. The research findings show that a program written in Node.js occupies 29 % of other code written in Java language. Whereas the processing time of an application written using Go and Node.js, is weaker compared to another created in Java language. Finally, both the two languages Node.js and Go, required low RAM during their executions [11].Another similar investigation was done by Chang et al. [12].The obtained results disclosed that Node.js adopts an event-driven architecture and provides a multi-process programing model i.e., Node.js allows the execution of several events in parallel.

The client can interact in real time with the machine and with the database through web services. Access to web services is generally done by two methods: Representational state transfer (REST) or Simple Object Access Protocol (SOAP). Malik et al. [13] shed light on a comparison of two REST and SOAP web services. The results show that the use of REST web service is the best solution in the context of an IoT system [13]. In addition, this web service simplifies the development and integration of Web components by improving the performance of the application. SOAP does not represent an efficient solution for communicating web components, since it only accepts the Extensible Markup Language (XML) format of the data a [13]. SOAP requires a high processing time because the XML format needs a parser at the server and client level.

In this respect, so as to develop a web server for real-time monitoring of the handball player's progress, it is crucial to study the architectures of IoT systems and the Cloud platforms used to tackle the similar problems. Knowing which Cloud platforms guarantee the fastest and the most secured data transfer methods is an essential task of this present study. Moreover, it is necessary to make the right choice of protocols which communicate the server with the data measured by the sensors. as well as studying both types of SQL and Nosql databases and choosing the appropriate database among them, depending on the response time of each and the amount of athlete data to be stored. Many of these choices come with very high costs for real-time data collection especially if it requires big storage and large size of messages.

To achieve our goals, this intended study aims to develop a customized server with a web architecture which can communicate, process than store data, connect devices used during athlete's training, and provide real time access to the data.

II. PROPOSED METHOD

During a handball player's training session, there are repetitive actions used to evaluate player performances by determining the player locations, ball placement, and when a ball hits or miss. Indeed, the proposed platform depicted in Fig. 1 is able to analyze all these actions (shots, locations, and results) and generate statistics on each handball athlete to help coaches to follow player progress in real-time.

Player's data are stored in a database to compare and display their ratings over time. As previously mentioned, all the information that comes from connected objects in a NoSQL-type database, which is MongoDB, were stored. A MongoDb database has the following advantages:

- The basic NoSQL response time such as MongoDb is much smaller than that of SQL [9].
- MongoDB is a distributed database that can use the technique called "Sharding" which increases request processing time [14].
- It allows data to be stored in JavaScript Object Notation (JSON) format, which makes it easier to process this data in the backend side of the application [15].Because JSON is a data format that is easy to parse by languages [16].

Each time the ball is shot or the locations of the player and ball are changed, the connected sensors send action to the server using the MQTT communication protocol. Indeed, the devices which host the sensors will be able to publish messages on an intermediary server (broker) via subjects (topics). Multiple connected cards can subscribe and publish to the same Topic. Therefore, the MQTT protocol enables small, low power, low memory devices to be routed in vulnerable areas and low bandwidth networks.

This protocol has a variety of advantages. It is fast, light, energy efficient and delivers varying levels of service quality as well. MQTT implements a classic publish / subscribe (pub/sub) model with a central broker [17].It has three levels of service quality (QOS) [8]. and has several security mechanisms including data encryption and entity authentication [18].

Unlike the traditional client-server model in which the client communicates directly with an endpoint, which is called a web service, MQTT clients are divided into two main groups: a sender (called Publisher in MQTT) and a consumer who receives the data (an MQTT subscriber). The Topic is an MQTT broker. It also acts as a ”traffic cop” who directs messages from Publisher to all endpoints (all subscribers).

The JSON format was chosen for sending sensor data to the MQTT Topic. It is a data exchange format that allows to store data of different types: character strings (including base64 images), numbers, arrays, objects, Boolean, etc. It is a data format which is easy to read and write for many uses. It has an interchangeable data format with a data structure

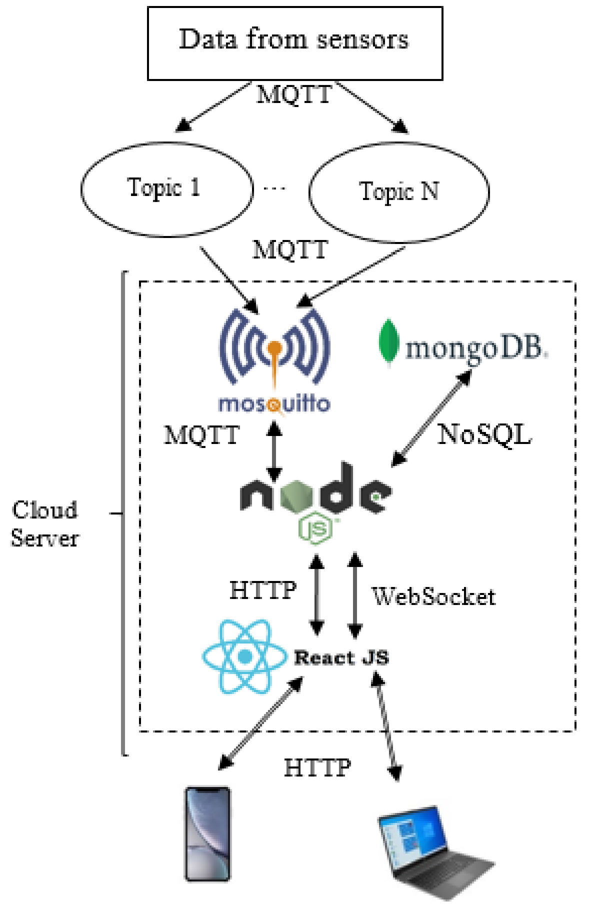

Fig. 1. Proposed architecture for the IoT web-based server.

built into programming languages that eliminates translation time and reduce complexity and processing time [16].Fig.2 shows a player position results relative to the middle of the field in JSON format sent by an electronic device using an ESP32 microcontroller. The proposed system is based on the three-thirds MVC in which the programming language Node.js was used to process data in the server side, and the MQTT protocol to communicate the sensors with the AWS server. MVC is defined as an architectural model commonly used to develop user interfaces that divide an application into three interconnected parts [19].

The three main layers used in our architecture are:

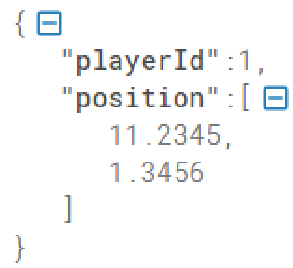

Fig. 2. Player position in a handball court using JSON Format.

- The application layer: that represents the data visible to users in web interfaces or mobile applications. In this present study, these interfaces are based on the ReactJS programming language.
- The network Layer: that includes the methods and protocols used to transfer sensor data to web / mobile of ReactJS interfaces as well as to control electrical systems and actuators from an order sent by the application layer. Two Hypertext Transfer Protocol (HTTP) communication protocols to link Node.js code with ReactJS user interfaces and MQTT to link sensor data with the developed web platform are used in the current work.
- The perception layer: that represents all the physical entities (sensors and actuators) responsible for processing orders from users and detecting a physical quantity such as temperature, the position of an object or another physical parameter.

In the proposed approach, the web interfaces can interact in real time with the sensors and the database through web services (API REST). Access to web services is generally done through the REST method which simplifies the development and integration of web components while improving the performance of the application.

To ensure that the data will be presented in real time, a WebSocket was used. Relying on this technique, the web client does not need to initiate an http request every time to establish a synchronous connection with the server. The customer establishes the connection only the first time.

Fig. 3. highlighted an example of a Handshake by WebSocket when sending a successful ball shot through a circuit in a handball court.

In the proposed architecture, two programming languages were used: Node.js on the server side and ReactJS on the web side. Owing to these two languages, it becomes easier to switch from the server side to interface code because both are developed with JavaScript language. The processing time of these languages is very fast because these languages use the least hardware resources (RAM and processor) to meet the user requests.

III. EXPERIMENTS AND RESULTS

To prove the performance of the Cloud platform, the server was deployed on different instance (computers on the cloud). Three commercial cloud services have been used to host

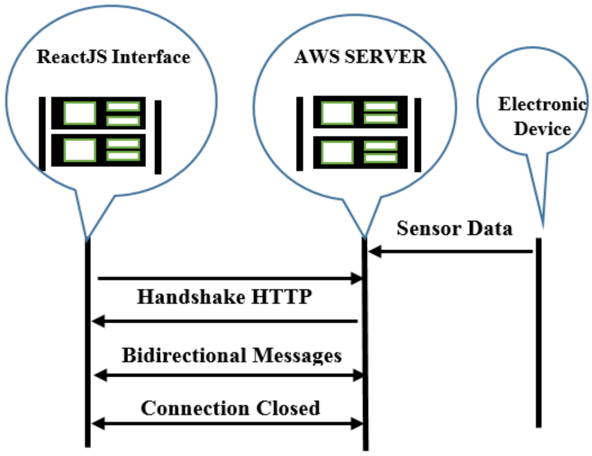

Fig. 3. Handshake by WebSoket when sending a successful ball shot.

the platform: Amazon AWS (EC2), Microsoft Azure Instance and Google Cloud Instance. We used for these testing three different instances but with the same features: 1 GB Ram, 8 GB Drive and Ubuntu 18 operating system. In each Cloud environment, Node.js (v 8.17.0) has installed with NGINX as web server and to manage the access to each instance.

hree Clouds platforms hosted on AWS, Microsoft Azure and GC then the response time was calculated for each according to the number of messages per second. In each instance, a Node.js code was runed based on the express library which allows you to respond with ”OK” to HTTP requests sent by clients. Fig. 4. shows the response time of 985 messages sent using PostMann software.

In Fig. 4. the processing time of these messages is smaller with in the AWS platform compared to MS Azure and GC. AWS requires a time between 20 and 30 seconds to respond to 985 messages, on the other hand, the response time of two platforms M. Azure and GC, is between 50 and 60 seconds.

Fig. 4. Cloud service time as a function of the number of messages sent.

When a successful or unsuccessful ball is detected, the Broker receives a message in JSON format which contains the shot position in meters compared to the handball field center and a Boolean variable used to indicate the shooting result either a success or a failure. After that, quick calculations perform to update all player statistics. Among these calculations, we have the one to trace a modulization of a handball court that contains green balls in the successful shot positions and crosses surrounded by red circles in the unsuccessful shot positions. This calculation consists of using the rule of three in order to transform the initial positions from meters to pixels (x: pixel, y: pixel). Once the processing is complete, the server initiates a handshake to update the web interfaces opened by the trainer.

The web interfaces visualize the statistics of handball athletes in real time. With each success or failure of a balloon shot, all statistics and interfaces are updated thanks to the integration of WebSocket in the developed platform. In this section, two examples of statistics generated by the web interfaces was illustrated.

Fig.5 presents the statistics of the failure (cross surrounded by a red circle) and the successes (green ball) of ball shooting.

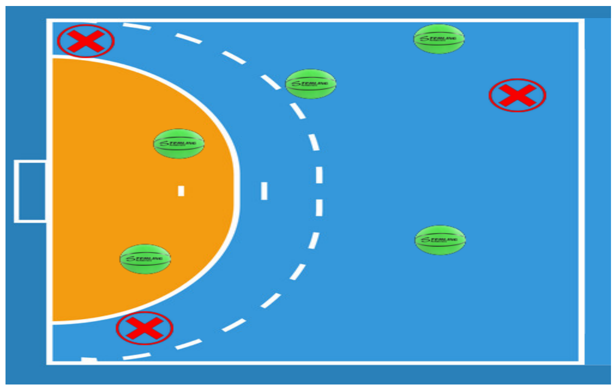

Fig. 5. Shooting Failures and Successes in the handball court.

The statistics saved in the Mongo DB database help coaches to evaluate their players during a predefined period (day, week, month, year and a number of predefined days). Functions that allow calculation of the average of ball shots per distance and angle are prepared to clarify to the coach the performance and the weakness of the player in each position of the field. The input to these functions is the period requested by the trainer (start date and end date) and after interacting with the MongoDB database, these functions respond to web requests sent by ReactJS interfaces. The success of the ball shot is calculated according to the periods fixed by the coach whether it is a specific day, a week, a month, a year or a well-defined period. An example of a player’s progress in shooting success is shown in Fig. 6.

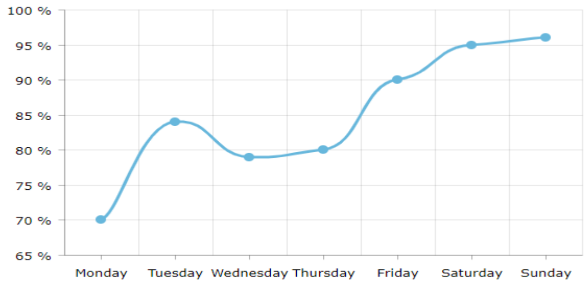

Fig. 6. Progress of a player for a week.

the Fig. 7. presents the percentage of successful shots of Handball players in relation with shooting angle in which we devided the Handball court into 8 areas according to shooting angle.

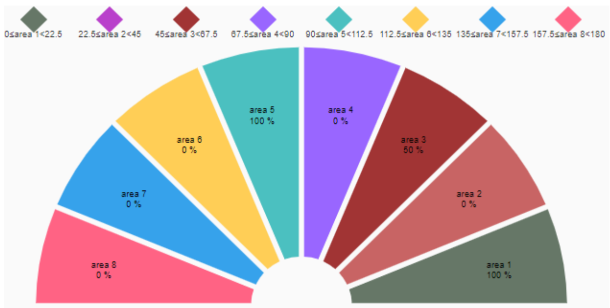

Fig. 7. statistics of a player according to the angle of shoot.

On the other hand, the web interfaces of our platform are responsive. Their display is adaptable automatically with all devices to facilitate access to data by trainers.

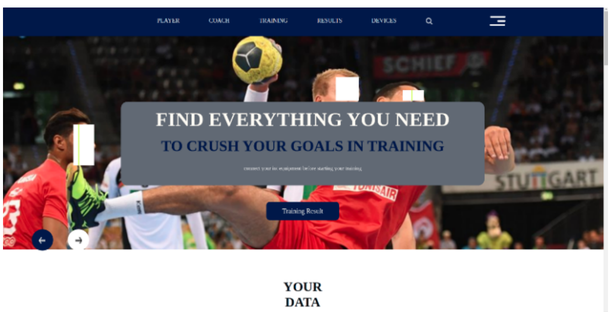

Fig. 8. Display of principal interface on a laptop.

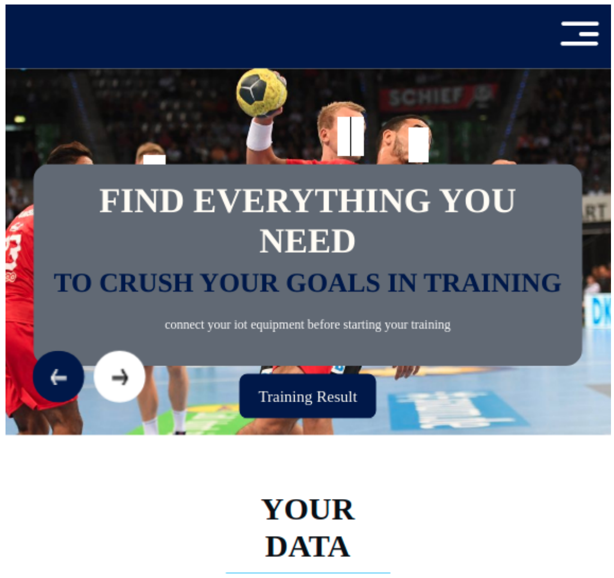

Fig. 9. Display of principal interface on an iPad.

The following Fig.8 display of principal interface seen on a laptop computer. Fig.9 illustrates the display of the same interface on an iPad.

IV. CONCLUSION

This paper presents a new approach to help coaches monitoring the performance of handball players. A web platform architecture has been developed to process player data in real-time. The results of this research show that customized we-based platform is efficient in terms of response time. As demonstrated, the platform applications can be fully deployed on the cloud. AWS instance seems to a good choice for hosting the platform in order to manage the athletes' data and information collected during training sessions. The data is transmitted from the connected devices to the Cloud platform via the MQTT protocol. This protocol widely used in IoT applications, meets the platform needs in terms of response time, memory occupancy and protection of data in traffic. Using MongoDB as database has backed up all the collected data. The proposed platform makes it possible for coaches to evaluate the progress of players over the time or redefined period. The results of this work show the facility of analyzing player skills via user-friendly graphical interfaces. This platform can be applied to other sports playable with a ball such as basketball and football.

REFERENCES

[1] W. Hong-Jiang, Z. Hai-Yan, and Z. Jing, "Application of the cloud computing technology in the sports training," in 2013 3rd International Conference on Consumer Electronics, Communications and Networks, Nov. 2013, pp. 162–165.

[2] L. Zhang and X. Shen, "Research and development of real-time monitoring system based on WebSocket technology," in Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Dec. 2013, pp. 1955–1958.

[3] A. Gupta, P. Goswami, N. Chaudhary, and R. Bansal, “Deploying an Application using Google Cloud Platform,” in 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Mar. 2020, pp. 236–239.

[4] A. Sheth, “Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing,” IEEE Intelligent Systems, vol. 31, no. 2, pp. 108–112, Mar. 2016.

[5] G. Davis, “2020: Life with 50 billion connected devices,” in 2018 IEEE International Conference on Consumer Electronics (ICCE), Jan. 2018, pp. 1–1.

[6] A. Bosche, D. Crawford, D. Jackson, M. Schallehn, and C. Schorling, “Unlocking Opportunities in the Internet of Things,” p. 12.

[7] P. Pierleoni, R. Concetti, A. Belli, and L. Palma, “Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison,” IEEE Access, vol. 8, pp. 5455–5470, 2020.

[8] A. A. Simiscuka, T. M. Markande, and G. Muntean, “Real-Virtual World Device Synchronization in a Cloud-Enabled Social Virtual Reality IoT Network,” IEEE Access, vol. 7, pp. 106588–106599, 2019.

[9] W. Khan, W. Ahmad, B. Luo, and E. Ahmed, “SQL Database with physical database tuning technique and NoSQL graph database comparisons,” in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Mar. 2019, pp. 110–116.

[10] C.-S. Park and H.-M. Nam, “Security Architecture and Protocols for Secure MQTT-SN,” IEEE Access, vol. 8, pp. 226422–226436, 2020.

[11] M. Dagenais, “Comparaison de plateformes logicielles pour programmation de services Web dans un environnement aux ressources limitées,” p. 85.

[12] X. Chang, W. Dou, Y. Gao, J. Wang, J. Wei, and T. Huang, “Detecting Atomicity Violations for Event-Driven Node.js Applications,” in 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), May 2019, pp. 631–642.

[13] S. Malik and D. Kim, “A comparison of RESTful vs. SOAP web services in actuator networks,” in 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Jul. 2017, pp. 753–755.

[14] M. Patil, “A qualitative analysis of the performance of MongoDB vs MySQL Database based on insertion and retriewal operations using a web/android application to explore Load Balancing – Sharding in MongoDB and its advantages,” Feb. 2017.

[15] D. Ramesh, A. Sinha, and S. Singh, “Data modelling for discrete time series data using Cassandra and MongoDB,” in 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), Mar. 2016, pp. 598–601.

[16] A. A. A. El-Aziz and A. Kannan, “JSON encryption,” in 2014 International Conference on Computer Communication and Informatics, Jan. 2014, pp. 1–6.

[17] M. Collina, G. E. Corazza, and A. Vanelli-Coralli, “Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST,” in 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), Sep. 2012, pp. 36–41.

[18] D. Dinculeană, “Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices,” Applied Sciences, vol. 9, p. 848, Feb. 2019.

[19] X.-B. Fu, S.-L. Yue, and D.-Y. Pan, “Camera-based Basketball Scoring Detection Using Convolutional Neural Network,” Int. J. Autom. Comput., vol. 18, no. 2, pp. 266–276, Apr. 2021.

[20] Saavedra, Jose M., et al. “ Handball game-related statistics in men at Olympic Games (2004-2016): Differences and discriminatory power (Estadísticas de juego en balonmano masculino en los Juegos Olímpicos (2004-2016): Diferencias y poder discriminatorio)”. Retos, no 32, mai 2017, p. 260-63. DOI.org (Crossref)

[21] Ibáñez, Sergio J., et al. “ Basketball Game-Related Statistics That Discriminate between Teams’ Season-Long Success ”. European Journal of Sport Science, novembre 2008. world, www.tandfonline.com