applied system innovation

MDPI

Article

Data Lake Architecture for Smart Fish Farming Data-Driven Strategy

Sarah Benjelloun *, Mohamed El Mehdi El Aissi, Younes Lakhrissi and Safae El Haj Ben Ali

SIGER Laboratory, University of Sidi Mohamed Ben Abdellah, Fes 30050, Morocco
* Correspondence: sarah.benjelloun@usmba.ac.ma

Abstract: Thanks to continuously evolving data management solutions, data-driven strategies are considered the main success factor in many domains. These strategies consider data as the backbone, allowing advanced data analytics. However, in the agricultural field, and especially in fish farming, data-driven strategies have yet to be widely adopted. This research paper aims to demystify the situation of the fish farming domain in general by shedding light on big data generated in fish farms. The purpose is to propose a dedicated data lake functional architecture and extend it to a technical architecture to initiate a fish farming data-driven strategy. The research opted for an exploratory study to explore the existing big data technologies and to propose an architecture applicable to the fish farming data-driven strategy. The paper provides a review of how big data technologies offer multiple advantages for decision making and enabling prediction use cases. It also highlights different big data technologies and their use. Finally, the paper presents the proposed architecture to initiate a data-driven strategy in the fish farming domain.

Keywords: data-driven strategy; big data; data lake; fish farming; data analytics

Citation: Benjelloun, S.; El Aissi, M.E.M.; Lakhrissi, Y.; El Haj Ben Ali, S. Data Lake Architecture for Smart Fish Farming Data-Driven Strategy. *Appl. Syst. Innov.* **2023**, *6*, 8. https://doi.org/10.3390/asi6010008

Academic Editor: Christos Douligeris

Received: 16 November 2022
Revised: 1 January 2023
Accepted: 3 January 2023
Published: 7 January 2023

1. Introduction

Nowadays, big data and its applications are ubiquitous since businesses and organizations have increasingly adopted data-driven strategies which allow them to tackle many challenges [1]. In parallel, cloud computing and high-performance computing became increasingly accessible as a service, which made adopting big data solutions in many domains possible [2]. However, despite all the advantages of adopting a data-driven strategy, the agricultural domain has not benefitted sufficiently from it yet, especially in the fish farming field [3]. Moreover, compared to other domains, the agricultural field, in general, and fish farming, in particular, has a high level of uncertainty due to various parameters that affect the quality and quantity of production [4]. Consequently, applying a data-driven strategy in the fish farming domain will not only bring benefits to farmers but also enhance the sector as a whole. The proposition to adopt a strategy for the whole domain instead of focusing on the fish farms specifically has not yet been discussed and the proposition of an architecture to initiate this strategy has not been addressed either. This assertion is a result of the literature and the more detailed comprehensive review is in the methodology section.

A data-driven strategy for fish farming will reduce uncertainties and provide continuous decision support, revolutionizing the fish farming sector [3]. Nevertheless, making the data the pillar of this strategy will directly impact the production efficiency and nutrition quality with minimal environmental damages, which implies social, economic, and ecological benefits [5].

To obtain crucial information about massive datasets, advanced data analysis techniques have been used to transform big data into valuable data since it helps organizations and businesses make better decisions. For example, in the field of healthcare

analytics, large datasets (provided by applications such as electronic health records and clinical decision systems) enable healthcare practitioners to provide effective and accurate solutions for patients based on their overall history rather than relying only on the current collected data [6]. It has to be noted that traditional data analytics are deprecated to use in a big data context as long as the V's that characterize big data, namely, veracity, value, velocity, volume, and variety will immediately affect the accuracy of results. Additionally, other characteristics may be added to big data such as viscosity, venue, validity, and viability [7]. Furthermore, many artificial intelligence (AI) methods were developed to extract valuable information and discover hidden patterns from enormous amounts of data more accurately and faster than traditional techniques [8], such as machine learning (ML) or data mining (DM). For instance, a detailed analysis of historical fish farming data, real-time collected data, and other parameters will provide a detailed and extensive vision of the fish farming market status, thereby taking the appropriate decisions regarding the fish farms management [9]. Fish farming data have three main sources: sensors, flat files, and APIs. The parameters collected from sensors mainly are the following: temperature, light, chemical composition, and average weight of fish. Flat files contain manual data that cannot be automated or has yet to be automated like food quantities or fish count. Furthermore, the APIs provide another type of data like weather data, fish prices, fish food prices, and others.

This article is organized into seven sections: First, an introduction to present big data applications and how they can be used to enhance domains. Second, we present the methodology used in this study. After that, in Section three we highlight the advantages of adopting data-driven strategies in several fields and how it enables the adoption of fish farming data-driven strategies. Section four presents the different big data technologies used for proposing a dedicated data lake architecture. Section five proposes a technical data lake architecture for handling fish farming data based on the Hadoop ecosystem. Section six is a discussion to shed light on the positive impact of adopting a data-driven strategy based on data lake architecture. Finally, Section seven resumes the different axes tackled in this paper, and then it describes future work that includes developing a proof of concept based on the proposed fish farming data lake architecture.

2. Materials and Methods

The primary research method for this study is a review of the literature. The objective is to identify sources that address the same research question: How can big data technologies help enhance the fish farming domain? All the literature available and published after 2018 has been assessed. Other than the period criteria, we used a few criteria to narrow down the sources: scholarly (peer reviewed) sources, full article publication, language publication in English, and articles focusing on technical design. To collect articles from renewed research databases such as web of science, Springer and IEEE Xplore, we used the following query: ["Big Data" or "Data-driven" ok "Big Data Technologies" ok "Data Lake Architecture"] and ["Aquaculture" ok "Fish Farming"]. It is important to note that this research is limited by manpower and time, and thus the sources found may not be exhaustive.

Following that, we examined the academic sources based on a comprehensive review in order to identify the current industry data-based solutions. The remaining sources address the following subjects:

- Review of big data in aquaculture;
- A data science method proposal to address water management, disease detection, feeding strategies, and fish behavior monitoring.

These sources focus on problem resolution at the farm level. The proposition of an architecture to initiate a data-driven strategy for the fish farming domain is yet to be addressed.

In the next step of this study, we follow an exploratory study to explore the existing big data technologies and propose an architecture applicable to the fish farming data-driven strategy.

3. Big Data Enabling the Adoption of Fish Farming Data-Driven Strategies

The amount of available data does not solely determine the big data value but is determined by how it has been used. Analyzing data from different sources increases operational efficiencies, upgrading product development, and driving new revenue and growth prospects by enabling smart decision making.

Big data technologies and their applications have been adopted in many fields and have earned one's spurs. Indeed, it offers multiple advantages in terms of detecting the leading causes of failures/anomalies, evaluating the potential risks that may occur, spotting fraudulent situations before it harms the business or increasing the accuracy of artificial intelligence models. Below is a review shedding light on the multiple advantages that big data offers on some domains [10]:

- Public sector: increasing transparency through accessible, connected data, identifying demands, customizing actions for appropriate products and especially services, decision making with fewer risks, and inventing modern services and products [11].
- Industry: precise demand forecasting, higher supply chain planning, sales support, advanced production operations, and integrating web search-based applications [12, 13].
- Marketing: near real-time clients' behavior analysis based on marketplace collected data, geo-targeted advertising, more effective product design, price and variety optimization, distribution, and logistics management [14].
- Healthcare: clinical decision support systems, personal analytics based on each patient's profile history, personalized treatment, illness pattern analysis, and performance-based payment for medical staff [15, 16].

Putting data at the center of a strategy will provide immense opportunities for organizations to improve [17, 18]. Adopting the same concept in the agricultural domain and especially in the fish farming sector will enhance the domain as a whole by handling multiple parameters (on the farm level as temperature, pH or on the domain level as investing in an area or another) that could tackle the development of the fish farming domain [19]. However, designing a fish farming data-driven strategy can only be possible by implementing a dedicated big data architecture, offering the ability to handle heterogeneous data from different sources.

Handling fish farming data is not as easy as it appears, since the objective is to implement a data-driven strategy whose cornerstone is built upon data. Nevertheless, the collected and generated data records have the characteristics of big data (the V's of big data), so handling it is challenging. As a consequence, designing a dedicated data lake architecture is primordial.

The data lake remains an adequate data handling platform for the fish farming use case, as it relies on the extract load transform (ELT) process rather than the extract transform load (ETL) process adopted in data warehouse architectures [20]. The ELT process allows ingesting data from heterogeneous sources without undergoing a specific schema and then processing the data to extract hidden patterns and valuable information before exposing it to the end user [21].

The data lake architecture has evolved from mono-zone architecture to multi-zone architecture [22]. With the mono-zone concept, the data lake has a flat architecture where all the collected data is stored in their raw format, with low costs. Despite that, the mono-zone data lake architecture has limited user access and restricted data processing operations. The multi-zone concept usually relies on three main zones, namely, the RZ (raw zone), CZ (conformed zone) and AZ (analytics zone). The RZ is where the data is stored as-is without any transformation; the ingestion can be in batch mode or real-time mode. The main goal of this zone is to provide data engineers with the original version of the

data and facilitate subsequent processing. The previously stored data is transformed at the CZ level to match specific needs. It can also be noted that both batch and real-time processing can be adopted to make data accessible for analysis purposes. Processing includes operations such as selecting, joining, and aggregating datasets [23]. The AZ, also known as the access zone, exposes the preprocessed data and access to it is granted to data analysts. This zone provides a self-service data exploration for the analytics use cases (business intelligence, machine learning algorithms, statistical analysis, and reporting).

Along with this, our proposed functional architecture for handling fish farming big data contains three main phases, namely, data generation and collection, data ingestion and treatment, and data analysis and exploration, as presented in the figure 1 below:

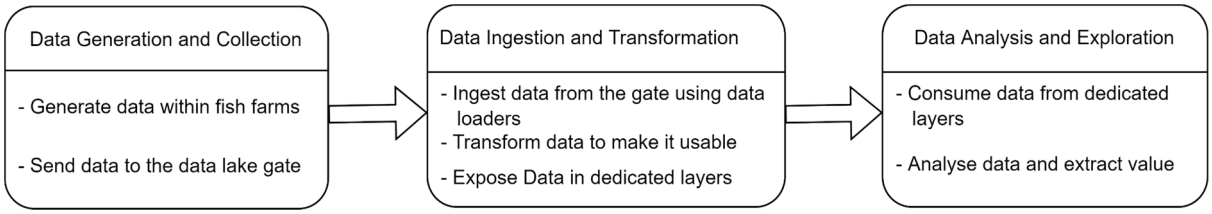

Figure 1. Process of Handling Fish Farming Data.

By concretizing this functional data handling architecture, the fish farming domain will not remain a traditional field but a data-driven domain, thanks to the possibility of analyzing and extracting valuable information from massive data gathered from multiple fish farms [24].

In other terms, the fish farming data lake will make data accessible for multiple users to perform analysis and not restrict access to only users with a solid technical background. Adopting a data-driven strategy will become possible, thus enhancing the whole domain.

4. Big Data Technologies for Fish Farming Use Case

4.1. Hadoop Ecosystem

Apache Hadoop is a popular big data technology with a large community. The main goal of this technology is to tackle the various challenges of performance and complexity while working with massive data using traditional systems. Hadoop is designed to effectively perform advanced processing on colossal data sets through a distributed file system [25]. Indeed, Hadoop runs the data processing tasks where the data are stored in the cluster and not copied in memory. As a result, Hadoop allows querying terabytes of data in a few seconds, with high fault tolerance, as it replicates data on servers to avert data loss [26].

Hadoop relies on two main components to provide a robust big data platform: Hadoop distributed file system (HDFS) and the MapReduce (MapR) model. Moreover, depending on the need, we may install new modules on top of Apache Hadoop to match applications' requirements.

4.2. Hadoop Distributed File System (HDFS)

Hadoop distributed file system is based on master-slave architecture to store massive data files with high-scalability and low-cost. It can store structured, unstructured, and semi-structured data. The main advantage of HDFS is the concept of delegating computation tasks near to data location since it reduces the network congestion in the cluster. Indeed, the cluster is constituted of two types of servers, the name node (master) that is responsible for managing file systems operations and multiple data nodes (slaves) that coordinate data storage and computations [27].

4.3. MapReduce Model (MapR)

MapReduce is a programming model implemented on top of HDFS. It is the cornerstone of big data management as it allows the efficient processing of huge amounts of data. The MapR model relies on parallel processing and contains two main functions: the Map and the Reduce [28].

In the Map function, there are two actions, first splitting the dataset into equal units or chunks constituting key-value pairs. The key-value pairs are given to the mapper that runs parallel mapping tasks across the cluster and resulting intermediate key-value pairs. It has to be noted that the resulting key values are sorted and grouped before transmitting them to the Reduce phase.

In the Reduce function, process the intermediate key-value pairs. For each key, the Reduce function aggregates the values with a coding logic to provide a summary of the entire dataset. Finally, the output is stored in the HDFS.

4.4. Data Exposition: Hive, Hbase, Elasticsearch

Apache Hive is a distributed, fault-tolerant data warehouse system built on top of HDFS. It is designed to efficiently store and analyze huge datasets by centralizing them in a single store. Indeed, Hive is based on structured tables. Each table has a related HDFS directory that is divided into partitions, and each partition is divided into buckets. In addition, for better data analysis, Hive provides a SQL-like language named HiveQL for querying data. Each HiveQL query is converted into MapR jobs processed in parallel and impacting data in HDFS directly [29].

Apache HBase is a distributed, column-oriented, non-relational database with a key/value model built on top of HDFS. It is designed to support high table updates. In addition, HBase allows gathering multiple data elements into column families with a unique row key; however, HBase is most effectively used to store non-relational data in data-driven strategies. The HBase API enables random, strictly consistent, and real-time access to large volumes of data. In fact, HBase tables are known as HStore, and each HStore has one or more associated map files stored in HDFS [30].

Elasticsearch is a NoSQL, document-oriented database. It stores data in an unstructured way and cannot be queried using SQL. Since it is based on indices, the entire object graph needs to be indexed to be searched. Elasticsearch is a distributed search and analytics engine. Kibana is a proprietary data visualization dashboard software for Elasticsearch. Kibana enables interactive exploration of data in Elasticsearch using Kibana Query Language to filter data using free text or field-based search. The elastic stack can be used not only for functional data but also to Store and analyze logs, metrics, and security event data [31].

4.5. Data Ingestion: Apache Sqoop, Apache Flume

Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and relational database management systems such as MySQL and Oracle. It is an open-source command-line interface application ETL tool [32].

Apache Flume is an open-source, powerful, and flexible system to collect, aggregate, and move large amounts of unstructured data from multiple data sources into Hadoop (HDFS or HBase). It is written in Java and has a flexible architecture based on streaming data flows. Flume has its own query processing engine to transform new batches before moving to the intended sink [32].

4.6. Data Processing: Apache Spark

Apache Spark is an open-source distributed data processing engine. Unlike the MapR framework, Spark uses in-memory caching to optimize performance. In addition, Spark enables complex processing on huge datasets through development APIs in Scala, Python, or Java. Indeed, Apache Spark relies on the resilient distributed dataset (RDD) concept,

which is a constructed collection of data from a source system or another RDD stored in memory. In case of failure, Spark can reconstruct the RDD thanks to a direct acyclic graph that describes the sequence of operations, hence the resilience of RDDs. It has to be noted that while working on DataFrames and datasets, Spark implicitly transforms them to RDD before performing any processing. The Spark framework includes many components, such as those listed below [33]:

- Spark Core: It is the cornerstone of the platform as it is responsible for managing memory, distributing tasks, interacting with HDFS, and fault recovery.
- Spark SQL: It is a distributed engine allowing fast, interactive querying compared to the MapR model. Moreover, it can write the output RDD to Hive tables and use HiveQL.
- Spark Streaming: It is a real-time engine allowing streaming data analysis. Mainly, it relies on the micro-batch data ingestion concept and enables data processing with almost the same logic as batch processing.

4.7. Stream Processing: Apache Kafka

Apache Kafka is an open-source distributed event store and stream-processing platform written in Java and Scala. Kafka provides a low-latency, high-throughput platform for handling real-time data feeds. It guarantees zero data loss and is very fast, as it performs over 2 million writes per second. Kafka is used to building real-time streaming pipelines to move data from one system to another. Indeed, Kafka is a publish–subscribe messaging system that receives data from multiple sources and makes it available for listening systems. An application publishes a stream of events to a topic on a Kafka broker, which can then be consumed independently by other applications [34].

4.8. Data Scheduling: Apache Oozie, Apache Airflow

Apache Oozie is a server-based workflow scheduling system to manage Hadoop jobs. The workflows on Oozie are based on XML (extensible markup language) and defined using control flow and action nodes in a directed acyclic graph (DAG) [35].

Airflow is an open-source workflow management platform for data engineering pipelines developed at Airbnb to manage workflows' increasing complexity. It is based on Python, and it authors workflows as DAGs of tasks.

5. Technical Data Lake Architecture Proposal for Fish Farming Data-Driven Strategy

With the emergence of data-driven strategies in many fields and the massive increase in the quantity of generated data by the fish farming domain, it is imperative to design a data lake architecture able to handle data coming from different sources and offer a solid platform to perform advanced analytics. The proposed fish farming data lake architecture, represented in Figure 2, represents a flexible, scalable, and efficient solution that covers the entire process of collecting, transforming, and exposing data.

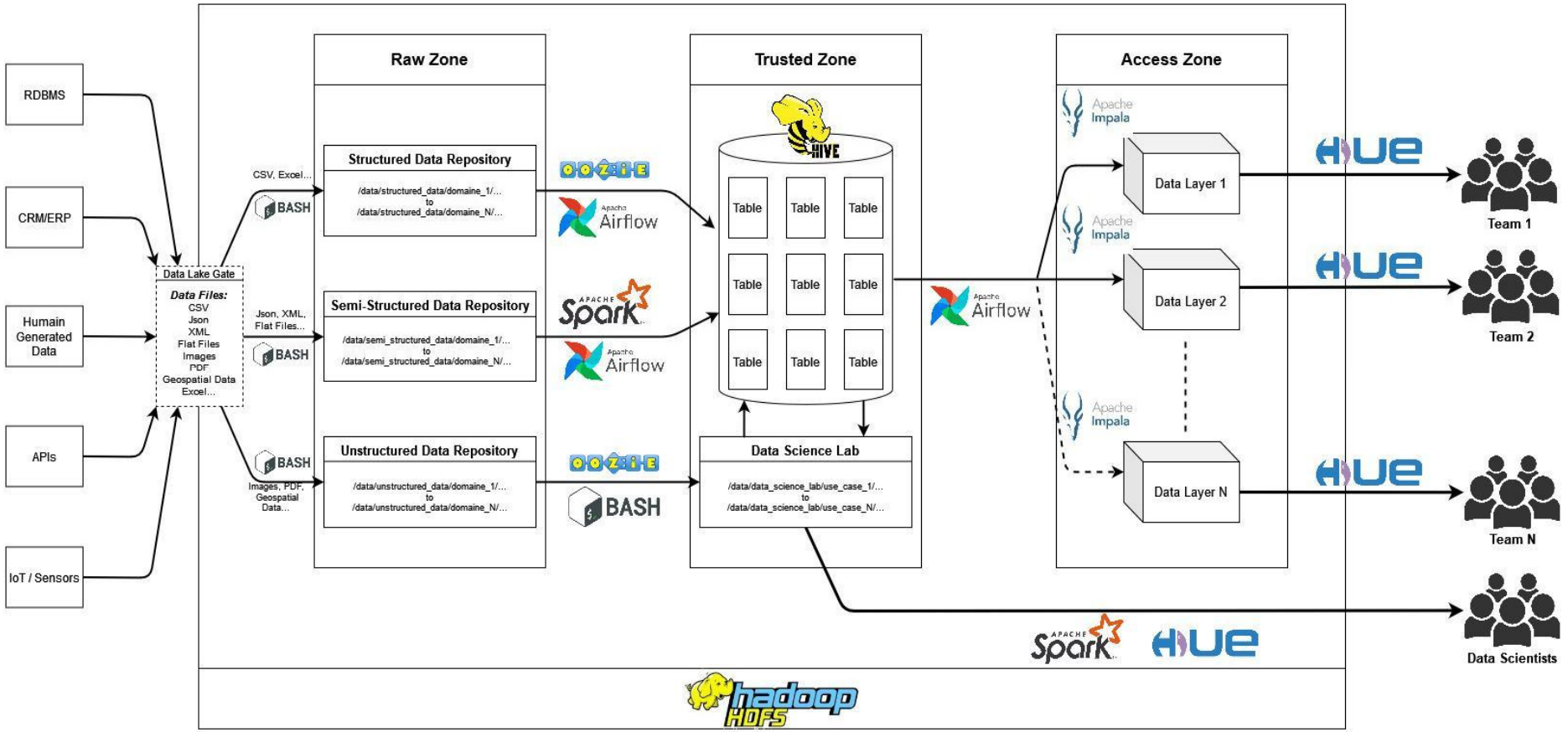

Figure 2. Fish Farming Data Lake Technical Architecture.

In order to ease the implementation of this architecture and make it widely accessible, the fish farming data lake was designed according to six main guidelines:

1. It should support handling the three types of data (structured, semi-structured, unstructured)
2. The process of collecting and ingesting data should be automated
3. Depending on the provenance and type, the data should be classified
4. It should support handling historical/backup data in raw format
5. Providing a dedicated data access layer for data analyst teams
6. Providing a data science laboratory for advanced analytics

In the same perspective, the proposed fish farming data lake architecture is multi-zone architecture with three main zones; namely, raw zone (RZ), trusted zone (TZ), and access zone (AZ).

The data could be gathered from different sources such as relational data base management systems (RDBMS), CRM/ERP, human-generated data such as flat files, application programming interfaces (APIs) like weather and IoT/Sensors for data like Ph, temperature, and pressure. Then, the data is directly submitted to the data lake edge node as it is considered the entry point.

5.1. Raw Zone

The gateway represents the entry point of the data lake. The main objective of the gateway machine is extracting data from multiple sources before loading it into the Hadoop distributed file system. Each data source has access to this server; however, this server should not be confused with HDFS. The received data files follow a nomination pattern that indicates their source and date. The reason behind using this approach is to allow keeping track of the collected data.

The data loader is the mechanism that allows the loading of data to HDFS. It is a developed job that reads the previously received data files from the gateway and stores them in the data lake file system. It relies on a defined naming pattern to identify the source, domain and time of data batches then it loads each one in the appropriate HDFS repository. In addition, the data loader can distinguish between structured, semi-structured, and unstructured data based on the data file extension. On the HDFS side, three repositories are created: structured, semi-structured, and unstructured data inside each directory, data is organized by source, and we have different domains inside each source. These three HDFS repositories constitute the raw zone. It must be noted that the data is stored as-is from the source at this level.

5.2. Trusted Zone

To pass this data to the trusted zone, we rely on different jobs depending on the data type. For structured data, it is directly stored in structured tables. Apache hive is a data warehouse system based on two types of tables: External and managed. External tables point directly to a remote data directory with a defined structure (fields, delimiter, etc.). Moreover, it has to be noted that Hive does not manage the storage of the external table; it only manages the metadata, which means, in case of deleting the external table, the data remains in HDFS, and only the table definition is deleted. For managed tables, also known as internal tables, both the storage and the metadata are managed by Hive, and deleting the table implies deleting the data from HDFS. The table creation is a one-time action. Still, the data ingestion into the conformed zone is an automated process using an orchestrator such as Apache Oozie that allows the execution of sequential and parallel tasks with a defined frequency to match the frequency of data collection.

Semi-structured data is not collected in a conventional or tabular form but is not completely unstructured. It contains some tags or key values that can be used to analyze that data. In our data lake use case, semi-structured data may refer to JSON files, XML files, or other markup language-based files. This data is transformed using dedicated Apache Spark jobs to match a defined structure at the level of the conformed zone. A dedicated

spark job is developed for each semi-structured data source to transform and store data values in the conformed zone in a structured hive table. The spark jobs are executed systematically using the orchestrator to ensure a continuous semi-structured data integration.

In the fish farming data lake use case, unstructured data, such as images and geospatial data, is usually used for data science use cases. For this purpose, data transition scripts are developed to create dedicated data science repositories for each use case.

5.3. Access Zone

The access zone objective is to allow easy access to all data available on the data lake for reporting, dashboarding, or data analysis purposes. In this optic, a dedicated layer is created for each need with controlled data access. These layers are either a group of hive tables or specific directories containing unstructured data. The data access is controlled using Apache Ranger policies. By this architecture, we can ensure that all external tools/teams can plug into the data lake and read the specified data.

6. Discussion

The agricultural sector produces massive data that can enhance the sector as a whole and anticipate market needs. This data is either produced by captors in the crops/tanks or collected from commercial activity. In addition, through big data technologies, advanced data analytics platforms are designed to facilitate the process of extracting valuable information from stored data for farmers and researchers [36]. For instance, weather forecasting can be performed based on the collected data from APIs since it is considered a key factor for fish farming. Moreover, it is very interesting for fish farmers to continuously measure tank temperature and Ph, which can be easily retrieved from data. In addition, big data technologies offer huge opportunities for farmers to adopt better fish management by analyzing different stages of fish farming, such as fingerling, nutrition, and production time [37].

Another critical point is to analyze the market information to obtain profit from fish production. It has to be noted that many parameters can constitute market data, such as input cost, price trends, farming cost, demand and supply, and transportation cost. Furthermore, this data may be accessible to other public and private parties [38].

Given these points, the proposed fish farming data lake architecture is designed to initiate the adoption of a data-driven strategy based on the results of analyzing huge data existing on the data lake. This architecture is designed to collect data existing in different farms with different formats and types. Then, it is directly saved into the RZ without applying any transformation to it, with the aim of forming a solid historical repository of the received data. Following that is the TZ, where lean transformations are applied to make data ready for further analysis. Finally, the AZ is made up of a dedicated layer with the required data for each team.

7. Conclusions

In this paper, we discussed the positive impact that data-driven strategies have when adopted in a domain. In addition, we take several domains, such as marketing, healthcare, and the industry of construction, as a reference to confirm that making data the center of a strategy allows extracting valuable information to support decision-making and predictive analysis. However, we shed light on the agricultural domain as it does not sufficiently benefit from the advantage of adopting a data-driven strategy, especially in the fish farming field.

In addition, designing a dedicated data lake architecture is the cornerstone of adopting a fish farming data-driven strategy. Furthermore, we propose a complete fish farming big data architecture for collecting, handling, and processing data, based on a multi-zones data lake architecture (RZ, CZ, and AZ), allowing huge data analysis with high efficiency

and scalability. Moreover, we present the previously proposed data lake functional architecture from a technical perspective by demystifying each Hadoop ecosystem component used while implementing this solution. On top of that, we explain the technical use of each one of the mentioned data lake zones.

The motive behind our study is to initiate a data-driven strategy that benefits the whole fish farming domain in an area or region and not focus only on a farm level. This idea translates into collecting and transforming huge data sets from multiple farms in order to extract more value and enhance not only production in each individual fish farm but also decision making and analytics of the domain in the whole area. Our contributions consist of:

- Proposing a data-driven strategy for the fish farming domain as a whole in a country or region;
- Presenting different big data technologies to use to collect, store, transform, and expose data in order to extract value from it;
- Proposing a technical architecture using these technologies to initiate a data-driven strategy.

Ultimately, our future work will focus on implementing this proposed dedicated fish farming data lake architecture to provide a proof of concept (POC) and simulate the data collection, transformation, and exposition process based on raw generated fish farming data.

Author Contributions: Conceptualization, S.B. and M.E.M.E.A.; methodology, S.B. and M.E.M.E.A.; validation, S.B., M.E.M.E.A., Y.L. and S.E.H.B.A.; formal analysis, S.B. and M.E.M.E.A.; writing—original draft preparation, S.B. and M.E.M.E.A.; writing—review and editing, S.B. and M.E.M.E.A., and Y.L.; supervision, Y.L. and S.E.H.B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sawant, N.; Himanshu, S. Big data application architecture. In *Big data Application Architecture Q & A.*; Apress: Berkeley, CA, USA, 2013; pp. 9–28.
2. Nachiappan, R.; Javadi, B.; Calheiros, R.N.; Matawie, K.M. Cloud storage reliability for Big Data applications: A state of the art survey. *J. Netw. Comput. Appl.* **2017**, *97*, 35–47.
3. Maru, A.; Berne, D.; De Beer, J.; Ballantyne, P.; Pesce, V.; Kalyesubula, S.; Fourie, N.; Addison, C.; Collett, A.; Chaves, J. Digital and data-driven agriculture: Harnessing the power of data for smallholders. *F1000Research* **2018**, *7*, 525.
4. Li, X.; Li, J.; Wang, Y.; Fu, L.; Fu, Y.; Li, B.; Jiao, B. Aquaculture industry in China: Current state, challenges, and outlook. *Rev. Fish. Sci.* **2011**, *19*, 187–200.
5. Elgendy, N.; Elragal, A. Big data analytics: A literature review paper. In *Proceedings of the Industrial Conference on Data Mining, St. Petersburg, Russia, 16–20 July 2014*; Springer: Cham, Switzerland; pp. 214–227.
6. Pramanik, P.K.D.; Pal, S.; Mukhopadhyay, M. Healthcare big data: A comprehensive overview. *Res. Anthol. Big Data Anal. Archit. Appl.* **2022**, 1, 119–147.
7. Panimalar, Arockia, Varnekha Shree, and Veneshia Kathrine. "The 17 V's of big data." International Research Journal of Engineering and Technology (IRJET) **2017**, *4*, 3–6.
8. Mahesh, B. Machine learning algorithms-a review. *Int. J. Sci. Res. (IJSR)* **2020**, *9*, 381–386.
9. Wang, C.; Li, Z.; Wang, T.; Xu, X.; Zhang, X.; Li, D. Intelligent fish farm—The future of aquaculture. *Aquac. Int.* **2021**, *29*, 2681–2711.
10. Sagiroglu, S.; Duygu, S. Big data: A review. In *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems (CTS) San Diego, CA, USA, 20–24 May 2013*; IEEE: New York, NY, 2013.
11. Coulthart, S.; Riccucci, R. Putting Big Data to Work in Government: The Case of the United States Border Patrol. *Public Adm. Rev.* **2021**, *82*, 280–289.
12. Li, C.; Chen, Y.; Shang, Y. A review of industrial big data for decision making in intelligent manufacturing. *Eng. Sci. Technol. Int. J.* **2022**, *19*, 101021.
13. Yoon, J.; Joung, S. A big data based cosmetic recommendation algorithm. *J. Syst. Manag. Sci.* **2020**, *10*, 40–52.

14. Cao, G.; Tian, N.; Blankson, C. Big data, marketing analytics, and firm marketing capabilities. *J. Comput. Inf. Syst.* **2022**, *62*, 442–451.
15. Rehman, A.; Naz, S.; Razzak, I. Leveraging big data analytics in healthcare enhancement: Trends, challenges and opportunities. *Multimed. Syst.* **2022**, *28*, 1339–1371.
16. Hussein, A.; Ahmad, F.K.; Kamaruddin, S.S. Cluster Analysis on covid-19 outbreak sentiments from twitter data using K-means algorithm. *J. Syst. Manag. Sci.* **2021**, *11*, 167–189.
17. Lusch, R.F.; Nambisan, S. Service innovation. *MIS Q.* **2015**, *39*, 155–176.
18. Rajaraman, V. Big data analytics. *Resonance* **2016**, *21*, 695–716.
19. Mouzakitis, S.; Tsapelas, G.; Pelekis, S.; Ntanopoulos, S.; Askounis, D.; Osinga, S.; Athanasiadis, I.N. Investigation of common big data analytics and decision-making requirements across diverse precision agriculture and livestock farming use cases. In *International Symposium on Environmental Software Systems*; Springer: Cham, Switzerland, 2020.
20. Nambiar, A.; Mundra, D. An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management. *Big Data Cogn. Comput.* **2022**, *6*, 132.
21. Aissi, E.; El Mehdi, M.; Benjelloun, S.; Loukili, Y.; Lakhrissi, Y.; Boushaki, A.E.; Chougrad, H.; Elhaj Ben Ali, S. Data Lake Versus Data Warehouse Architecture: A Comparative Study. In *WITS 2020*; Springer: Singapore, 2022. pp. 201–210.
22. Ravat, F.; Zhao, Y. "Data lakes: Trends and perspectives. In *Proceedings of the International Conference on Database and Expert Systems Applications Linz, Austria, 26–29 August 2019*; Springer: Cham, Switzerland, 2019.
23. Benjelloun, S.; El Aissi, M.E.M.; Loukili, Y.; Lakhrissi, Y.; Ali, S.E.B.; Chougrad, H.; El Boushaki, A. Big data processing: Batch-based processing and stream-based processing. In *Proceedings of the 2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS) Fez, Morocco, 21–23 October 2020*; IEEE: New York, NY, USA, 2020.
24. Benjelloun, S.; Aissi, M.E.M.E.; Lakhrissi, Y.; Ali, S.E.H.B. Big Data Technology Architecture Proposal for Smart Agriculture for Moroccan Fish Farming. *WSEAS Trans. Inf. Sci. Appl.* **2022**, *19*, 311–322.
25. Vohra, D. *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools*; Apress: Berkeley, CA, USA, 2016.
26. Monteith, J.Y.; McGregor, J.D.; Ingram, J.E. Hadoop and its evolving ecosystem. In Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO 2013) Potsdam, Germany, 11 June 2013; Volume 50.
27. Oussous, A.; Benjelloun, F.Z.; Lahcen, A.A.; Belfkih, S. Big Data technologies: A survey. *J. King Saud Univ.-Comput. Inf. Sci.* **2018**, *30*, 431–448.
28. Condie, T.; Conway, N.; Alvaro, P.; Hellerstein, J.M.; Elmeleegy, K.; Sears, R. MapReduce online. *Nsdi* **2010**, *10*, 20.
29. Shaw, S.; Vermeulen, A.F.; Gupta, A.; Kjerrumgaard, D. Hive architecture. In *Practical Hive*; Apress: Berkeley, CA, USA, 2016; pp. 37–48.
30. Prasad, B.R.; Agarwal, S. Comparative Study of Big Data Computing and Storage Tools: A Review. *Int. J. Database Theory Appl.* **2016**, *9*, 45–66.
31. Elasticsearch, B.V. Elasticsearch. 2018. Available online: https://www. elastic. co/pt/ (accessed on 12 September 2019).
32. Lakhe, B. Implementing SQOOP and Flume-based Data Transfers. In *Practical Hadoop Migration*; Apress: Berkeley, CA, USA, 2016; pp. 189–205.
33. Salloum, Salman, et al. "Big data analytics on Apache Spark." International Journal of Data Science and Analytics 1.3 (2016): 145-164.
34. Bandi, A.; Hurtado, J.A. Big data streaming architecture for edge computing using kafka and rockset. In *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 29–31, March 2022*; IEEE: New York, NY, USA, 2021.
35. Islam, M.K.; Srinivasan, A. *Apache Oozie: The Workflow Scheduler for Hadoop*; O'Reilly Media, Inc.: Sebastopol, CA, USA,, 2015.
36. Lokers, R.; Knapen, R.; Janssen, S.; van Randen, Y.; Jansen, J. Analysis of Big Data technologies for use in agro-environmental science. *Environ. Model. Softw.* **2016**, *84*, 494–504.
37. Bendre, M.R.; Thool, R.C.; Thool, V.R. Big data in precision agriculture: Weather forecasting for future farming. In Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 4–5 September 2015; pp. 744–750.
38. Sarker MN, I.; Islam, M.S.; Murmu, H.; Rozario, E. Role of big data on digital farming. *Int. J. Sci. Technol. Res.* **2020**, *9*, 1222–1225.