International Journal Software Engineering and Computer Science (IJSECS)
5 (3), 2025, 1307-1321
Published Online December 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i3.5419.

RESEARCH ARTICLE Open Access

Apache Spark for Business and Financial Data Engineering: A Systematic Literature Review

Ahmad Bilal Almagribi *
Department of Information Technology, Universitas Teknologi Digital Indonesia, Bantul Regency, Special Region of Yogyakarta, Indonesia.
Corresponding Email: abalmagribi@gmail.com.

Bambang Purnomosidi Dwi Putranto
Department of Information Technology, Universitas Teknologi Digital Indonesia, Bantul Regency, Special Region of Yogyakarta, Indonesia.
Email: bpdp@utdi.ac.id.

Received: August 24, 2025; Accepted: November 15, 2025; Published: December 1, 2025.

Abstract: This paper is an SLR that maps the application of Apache Spark in data engineering in the business and finance domains. Practitioners and researchers alike would find it interesting to know how Apache Spark has been applied to solve big data problems as organizations continue to deal with large volumes of data. By analyzing publications from the Scopus database for 2021-2025, we try to find trends and methodologies currently in use as well as gaps in research existing in the field. It was found that Apache Spark is primarily used for sentiment analysis and trend analysis on social media, particularly Twitter, since its real-time processing capability can help understand market dynamics and consumer behavior. The platform carries out predictive tasks like predicting customer churn or pricing financial assets (stocks, bonds, options), proving its versatility across different business applications. Also, this platform is popular for anomaly detection such as transaction fraud with efficiency and cost being the main drivers of adoption. The landscape is not monolithic since some studies propose alternative platforms indicating that Apache Spark may not be the best option for every scenario. Based on our findings, we suggest future research directions that would push the boundaries of the field: using social media data sources other than Twitter for more general market sentiment, applying more varied algorithms to improve prediction accuracy, and extending Spark's application into new areas like currency exchange rate forecasting, credit risk analysis, Anti-Money Laundering (AML) detection as well as Data Lakehouse architecture implementation. These recommendations are meant to steer researchers toward uncharted territories where significant value could be unlocked for business and finance with the help of Apache Spark.

Keywords: Apache Spark; Business; Finance; Data Engineering; Systematic Literature Review (SLR).

1. Introduction

As the quantity of data grows at an increasing rate, data storage and processing has become a major issue for many organizations around the world [1]. The quantitative explosion in digital data has forced researchers and practitioners to seek new methods of visualization and analysis, which in turn requires advanced tools for its efficient processing. These tools have become increasingly important in a wide range of areas—such as healthcare, engineering, banking, chemistry, finance as well as data mining and cloud computing—where timely insight from large datasets can mean the difference between winning or losing market share. In-memory computing frameworks like Apache Spark are increasingly seen as necessary for such efficient processing because they provide functionality that cannot be achieved with traditional systems. Spark is now widely recognized as an effective way to easily write distributed applications while providing classic processing libraries that enable complex data operations to be expressed very simply. Thanks to its excellent performance characteristics, Spark can process both disk-based and in-memory datasets; thus it is a state-of-the-art tool enabling us to reconcile speed with scalability [2]. The Apache Spark programming framework is an efficient open-source platform for organizing and testing big data at scale with high dependability. It offers programming interfaces supporting all ranges of Big Data applications from batch processing to real-time streaming analytics. The testing techniques provide optimal answers for some difficult business problems by processing very large volumes of data while still maintaining accuracy and reliability [3]. In contrast to traditional standalone machines, analytics using the distributed Apache Spark (AS) framework reduces training and testing time by factors usually measured in orders of magnitude [4].

Apache Hadoop and Apache Spark are two of the most popular distributed frameworks for big data applications. In many cases, these frameworks are used for business-critical processes and therefore it is important to be able to forecast the execution time of submitted applications with a high degree of accuracy when SLAs exist with end users and predictable performance is required [5]. Apache Spark has been used in various ways in business and finance, such as social media sentiment analysis, fraud prediction, security improvement, and operational efficiency improvement. However, there has not yet been a specific study that consolidates all these applications in order to identify gaps and new opportunities for future researchers. The existing scattered research does not help practitioners understand the full range of what Apache Spark can do well or poorly in the areas of business and finance. Therefore, we perform an SLR on the use of Apache Spark for business and finance data engineering based on the Scopus database to give a structured overview of this field. Our review tries to answer some key questions: What are the main application areas? Which methodologies work best? What are the gaps in current research? Where should future efforts focus for maximum impact? The rest of this paper is organized as follows: Section II describes the research method by explaining our systematic approach to literature selection and analysis; Section III presents descriptive results from the SLR organized by year, sub-topic clusters accompanied by recommendations for future research directions; finally, Section IV wraps up with a conclusion highlighting key findings along with their implications toward researchers as well as practitioners in this domain.

2. Related Work

Apache Spark has emerged as a leading framework for big data processing in business and finance applications due to its in-memory computing capabilities and distributed processing architecture. This section reviews key research areas where Spark has been applied over the past five years. In the area of data optimization and processing infrastructure, Aziz *et al.* (2021) provided a brief literature review on data optimization using shuffling and partitioning for in-memory computing in Apache Spark, conducting various simulations focused on optimizing the performance of data partitioning and shuffling. The authors highlighted how to properly tune the number of partitions and how to avoid expensive shuffling, which is critical for applications in finance, banking, data mining, cloud computing, healthcare, engineering, and chemistry [2]. Martinez-Mosquera *et al.* (2021) combined three methods to describe how Big Data tools interpret complicated XML schemas for practical applications, using three main techniques: positional explosion, deserialization, and cataloging. Test results validated the proposed method for various versions of Apache Hive and Apache Spark, showing that Apache Hive's external tables enabled faster query execution speeds for full data frame queries, while Apache Spark was less efficient for queries on specific values or attributes due to extra in-memory processing requirements [6]. Pallamala & Rodrigues (2022) showed how to successfully test unstructured and semi-structured data from Apache Spark data samples using the Testing Whiz tool. When compared to other automated testing solutions like Hadoop, Apache Spark produced faster testing results, improving accuracy, boosting efficiency, lowering expenses, and saving time [3].

For financial applications, Gu *et al.* (2021) proposed Alchemy, a distributed processing platform with a high-level programming interface created especially for financial quantitative analysis (FQA). Alchemy reduces

the learning process for financial quantitative analysts by providing distributed computing resources and a pipeline-based programming model similar to classic standalone systems. The effectiveness of Alchemy was assessed at Huatai Securities in China, and according to the findings, Alchemy could outperform current FQA systems by up to 300 times [1]. Zhou *et al*. (2021) suggested an intelligent and distributed big data technique for the identification of online financial fraud using the Node2Vec graph embedding algorithm to represent topological features in a financial network graph into low-dimensional dense vectors. Large datasets were processed in parallel using this distributed method on a Hadoop cluster with Apache Spark GraphX, and experimental results demonstrated improved precision, recall, F1-Score, and F2-Score rates [10]. Lin & Lin (2023) proposed a new approach to build a bond price predictive model based on technical indicators in the financial market, applying machine learning techniques on the Apache Spark framework. The experimental results showed that the proposed approach, which considers technical indicators and dimensionality reduction, outperformed the baseline results for bond price prediction in terms of MAE and RMSE [21]. Xiong *et al*. (2023) showed how the least-squares Monte Carlo (LSMC) method in American option pricing could be made more accurate and efficient by using distributed computing with a divide-and-conquer strategy carried out by Apache Spark. Distributed computing offered benefits such as lowering computational complexity, avoiding intricate matrix transformations, and improving data security and privacy [23].

In sentiment analysis and social media analytics, Lijo & Seetha (2021) offered a quick method of polarity detection using a multi-lexicon feature for Twitter sentiment analysis. The binary-valued multi-lexicon capability saves time and space when managing huge data, and the authors employed a modified Sequential Minimal Optimization (SMO) with Apache Spark. Experiments demonstrated that this approach enhanced polarity detection efficiency and offered great scalability [7]. Raviya & Mary Vennila (2021) performed sentiment analysis on social media big data to develop predictions and create business plans that preserve a company's reputation. A model known as Hybrid CNN-SVM was created on a pipelined Apache Spark ML architecture, and experimental results showed that the hybrid sentiment analysis model performed better than traditional models (Naive Bayes, SVM, Random Forest, Logistic Regression) across a range of evaluation metrics [8]. Rodrigues *et al*. (2021) examined the relative popularity of different hashtags and the fields with the highest voice share using LDA, cosine similarity, K-means clustering, and Jaccard similarity algorithms on Apache Spark. While Jaccard similarity generated an accuracy of 83% for static data, the LDA approach for trend analysis produced an accuracy of 74% 0. Abhijith & Gundad (2023) developed a system based on machine learning algorithms to analyze sentiment in large datasets sourced from social media sites, implementing Naive Bayes and Support Vector Machines classification techniques on Apache Spark. The algorithm proved quite effective in managing large sentiment datasets [19]. Patil *et al*. (2025) provided a thorough examination of social media sentiment analysis approaches, covering traditional ML methods (Naïve Bayes, SVM, Decision Trees, Random Forest) and sophisticated DL models (RNN, LSTM, CNN, BERT, GPT), as well as big data frameworks like Hadoop, Apache Spark, and Apache Flink [32].

For prediction and churn analysis, Jain *et al*. (2022) focused on the importance of features and feature engineering for telecommunication customer churn prediction models, employing a two-phase classification ensemble of Random Forest and Gradient Boosted Trees on the Cell2Cell dataset. With grid search for hyperparameter optimization, the model's accuracy was 95% with Random Forest and 97% with Gradient Boosted Trees [12]. Tariq *et al*. (2022) proposed a model to assist e-businesses in predicting user churn using a 2-D convolutional neural network (CNN) processed in a parallel setting using Apache Spark's distributed architecture. The suggested model obtained an accuracy score of 0.963 with training and validation loss of 0.004, and confusion matrix results indicated a 94% true-negative rate and a 95% true-positive rate [16]. In performance prediction and resource management, Ataie *et al*. (2022) proposed and validated a hybrid approach (Apache Hadoop and Apache Spark) for the performance prediction of big data applications running in the cloud using analytical modeling and machine learning approaches. The authors contrasted this method with Ernest, an ML-based method suggested by Spark's developers, and showed how their proposed method significantly reduced prediction error, especially when only a few experiments were conducted on the real system [5]. Karimian-Aliabadi *et al*. (2023) presented a precise performance prediction model based on stochastic activity networks (SANs) that takes into account multiple work queues for multi-queue YARN clusters running DAG-based Big Data applications. Tests using the TPC-DS benchmark demonstrated that the suggested model predicted Spark task execution times with an average inaccuracy of just 5.6% [24]. Mendes *et al*. (2024) proposed MAS-Cloud+, a new agent-based architecture for forecasting, allocating, and tracking optimal cloud computing resources with three reasoning models (heuristics, formal optimization, metaheuristics). Results showed that MAS-Cloud+ could lower the average cost of executing BigDataBench benchmarking workloads by almost 58% 0. For security and anomaly detection, Hasan *et al*. (2022) compiled a review from various researchers from 2010 to 2022, finding that the performance and precision of Spark ML are fundamentally better than Spark MLlib, using a dataset of bank client transactions for security-related data analysis [13]. Azeroual & Nikiforova (2022) presented security-relevant data analysis using the Apache Spark big data analytics engine incorporated into a business intelligence system. The k-means approach for clustering

analysis in Spark's MLlib was used to create a prototype intrusion detection system that uses machine learning to identify data anomalies [14]. Hagar & Gawali (2022) suggested three models (Apache Spark, LSTM, and CNN) to enhance the detection of all kinds of attacks using the CSE-CIC-IDS2018 dataset. After feature selection using Random Forest and dataset balancing, the Apache Spark model yielded the best results with an accuracy of up to 100% for every class and F1-score of 1.00 for most classes [18].

Several advanced applications have demonstrated Spark's versatility. Shrotriya *et al*. (2023) examined a case study on how Spark is used in the healthcare sector for remote patient monitoring, medical image analysis, tailored treatment, predictive analytics for disease outbreaks, and EHR management, while also identifying constraints pertaining to data security, infrastructure scalability, and talent shortages [20]. Ayub *et al*. (2022) provided a data pipeline system for processing video streams and images in a distributed setting using Apache Spark, Kafka, and OpenCV on commodity hardware, achieving effective video stream processing for face detection without requiring GPU-based hardware or cloud computing infrastructure [17]. Bachir Belmehdi *et al*. (2024) offered a novel solution using deep learning to predict the best virtual data model for queries on big datasets. The OPTIMA system, utilizing Apache Spark and GraphX, reduced query execution time by more than 40% for tabular model selections and more than 30% for graph model selections with an accuracy of 0.831 [27]. La Gatta *et al*. (2024) extended CASTLE (Cluster-Aided Space Transformation for Local Explanations) to manage high-volume datasets using Apache Spark. The method has a sub-linear rather than an exponential dependence on the size of the dataset, making it scalable for big data scenarios in financial, medical, and military sectors [28]. Aladib *et al*. (2025) suggested a novel runtime verification framework to incorporate Linear Temporal Logic (LTL) monitoring into stream processing applications like Apache Spark. A case study on real-time financial data monitoring showed that LTL-based monitoring can successfully identify safety and liveness property violations while preserving stable latency [29]. Ionescu *et al*. (2025) proposed a four-layer design (data sources, data integration, processing, and storage) to solve issues with financial data processing, with Apache Spark for real-time data analysis, Hadoop for batch processing, and ML infrastructure for predictive modeling [35]. Regarding alternative frameworks, Shaikh *et al*. (2022) introduced GeoFlink, an extension of Apache Flink that supports continuous queries on spatial data streams with grid-based indexing. Thorough experimental research demonstrated that GeoFlink achieved much greater query throughput compared to Apache Spark Streaming and Apache Samza [15]. Despite extensive research, gaps remain in exploring modern algorithms (Transformers, LSTM, GRU) for financial forecasting, Data Lakehouse implementations with Delta Lake, ethical AI considerations, and specialized applications like AML detection and CLV prediction.

3. Research Method

This research employs a systematic literature review (SLR) methodology to examine the available literature on Apache Spark's use in business and finance data engineering. We chose the SLR approach because it provides a rigorous, transparent, and replicable method for identifying, evaluating, and synthesizing existing research, thereby minimizing bias and ensuring comprehensive coverage of the field. Data were sourced from Scopus, a reputable scientific database that aggregates publications from major, high-quality publishers such as IEEE Xplore, Elsevier, Taylor & Francis, Wiley, Emerald, SAGE, and Springer. Scopus was selected due to its extensive coverage of peer-reviewed literature, consistent indexing standards, and robust search capabilities that enable precise query formulation. The systematic selection process followed a multi-stage filtering approach to ensure that only relevant, high-quality publications were included in the final analysis.

The document selection process in Scopus.com was conducted as follows (see Figure 1). On July 12, 2025, we performed an initial search with the query "apache spark AND business OR financ*" in the abstracts of Scopus-indexed scientific works, which yielded 231 documents. This broad search captured all publications mentioning Apache Spark in conjunction with business or finance-related terms. To focus on recent developments and current trends, we applied a temporal filter for the last five years (2021–2025), reducing the count to 97 documents. This timeframe was chosen to balance recency with sufficient publication volume, capturing the most relevant advancements in the field while excluding outdated methodologies. Finally, we filtered for journal publications only and restricted the language to English, resulting in 36 documents. Journal articles were prioritized over conference papers and other publication types because they typically undergo more rigorous peer review and provide more detailed methodological descriptions. The language restriction to English was necessary to ensure consistent analysis and interpretation across all selected publications.

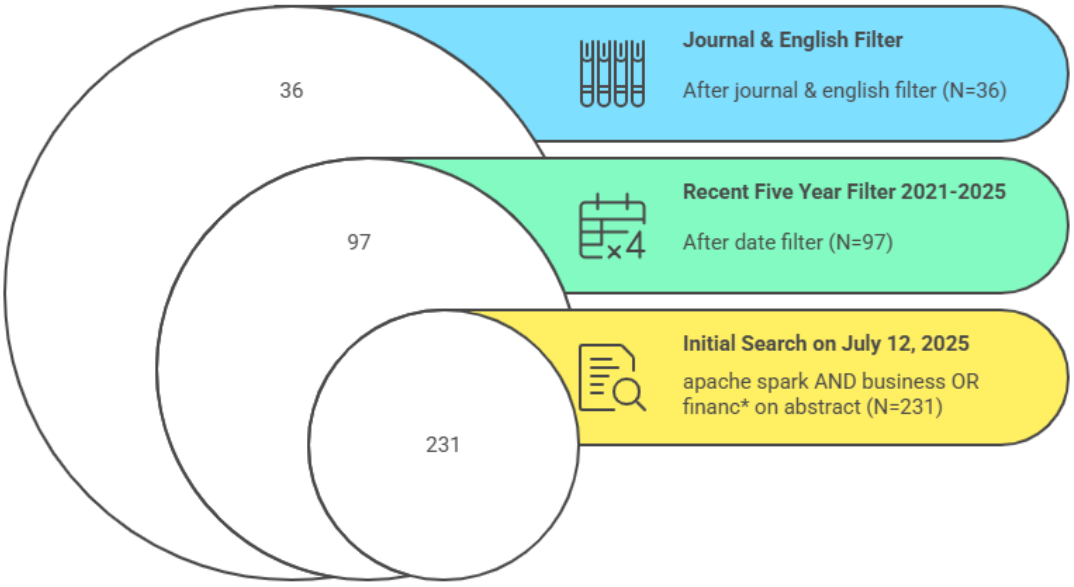

Figure 1. The Document Selection Criteria in Scopus

Ultimately, 36 documents obtained from Scopus on July 12, 2025, were selected for comprehensive analysis. The final search query after all filters were applied is: (ABS (apache spark) AND ABS (business OR financ*)) AND PUBYEAR > 2020 AND PUBYEAR < 2026 AND (LIMIT-TO (SRCTYPE , "j")). This query ensures reproducibility, allowing future researchers to verify our selection process or update the review with newer publications. The systematic literature review analysis was conducted in three distinct stages. First, we systematically described all 36 scientific articles that discuss Apache Spark's use in business and finance data engineering, extracting key information including research objectives, methodologies, datasets, algorithms, performance metrics, and main findings. Second, we clustered the sub-topics from the described articles through thematic analysis, identifying patterns and grouping similar research focuses to reveal the dominant application areas and methodological approaches. Third, based on the identified clusters and observed gaps in the literature, we recommended potential sub-topics that could offer novelty for future research, considering both underexplored areas and emerging trends that warrant further investigation. This structured analytical approach enables us to provide not only a comprehensive overview of the current state of research but also actionable guidance for advancing the field.

4. Result and Discussion

4.1 Results

4.1.1 Descriptive Analysis

The authors systematically describe 36 scientific articles that discuss the application of Apache Spark in business and finance data engineering, organized chronologically over the last five years, from works published in 2021 to 2025.

4.1.1.1 Year 2021:

Martinez-Mosquera *et al.* (2021) combined three methods to describe how Big Data tools interpret complicated XML schemas for practical applications. These XML files can be parsed using three main techniques: positional explosion, deserialization, and cataloging. The authors created a case study using actual datasets from the performance management of two mobile network suppliers. Test results validated the proposed method for various versions of Apache Hive and Apache Spark, obtaining query execution times for Apache Hive's internal and external tables as well as Apache Spark's data frames. In contrast to Apache Spark, Apache Hive's external tables enabled faster query execution speeds for full data frame queries. Additionally, for queries on specific values or attributes, Apache Spark was less efficient than Apache Hive's internal or external tables due to extra in-memory processing requirements. Complex XML schemas are utilized in many different applications, including social networks and financial data [6].

Lijo & Seetha (2021) offered a quick method of polarity detection using a multi-lexicon feature for Twitter sentiment analysis. When managing huge data, the binary-valued multi-lexicon capability saves time and space. To efficiently process big datasets, the authors employed a modified Sequential Minimal Optimization (SMO) with Apache Spark. Experiments demonstrated that this approach enhanced polarity

detection efficiency and offered great scalability [7]. Raviya & Mary Vennila (2021) performed sentiment analysis on social media big data to develop predictions and create business plans that preserve a company's reputation. This was done on a pipelined Apache Spark ML architecture to improve machine efficiency and speed up computation. A model known as Hybrid CNN-SVM was created. The effectiveness of the suggested system in a multi-node setting was assessed using Naive Bayes, SVM, Random Forest, and Logistic Regression classifiers. According to experimental results, the hybrid sentiment analysis model performed better than traditional models across a range of evaluation metrics. The proposed method permits the effective handling of huge sentiment datasets [8]. Rodrigues *et al*. (2021) examined the relative popularity of different hashtags and the fields with the highest voice share. Business, marketing, politics, sports, and entertainment all heavily rely on Twitter trends. LDA, cosine similarity, K-means clustering, and Jaccard similarity algorithms were used to analyze Twitter trends. While Jaccard similarity generated an accuracy of 83% for static data, the LDA approach for trend analysis produced an accuracy of 74%. These findings showed that real-time tweets may be evaluated comparatively more quickly with Apache Spark's Big Data tools [9]. Aziz *et al*. (2021) provided a brief literature review on data optimization using shuffling and partitioning for in-memory computing in Apache Spark. They also conducted various simulations focused on optimizing the performance of data partitioning and shuffling. The authors highlighted how to properly tune the number of partitions and how to avoid expensive shuffling, which is critical for applications in finance, banking, data mining, cloud computing, healthcare, engineering, and chemistry [2].

Zhou *et al*. (2021) suggested an intelligent and distributed big data technique for the identification of online financial fraud. This was accomplished by learning and representing topological features in a financial network graph into low-dimensional dense vectors using the Node2Vec graph embedding algorithm. This made it possible to use a deep artificial neural network to classify and predict data samples from massive datasets. Large datasets were processed in parallel using this distributed method on a Hadoop cluster with Apache Spark GraphX. With improved precision, recall, F1-Score, and F2-Score rates, the experimental results demonstrated that the suggested method could increase the effectiveness of online financial fraud detection [10]. Gu *et al*. (2021) proposed Alchemy, a distributed processing platform with a high-level programming interface created especially for financial quantitative analysis (FQA). Alchemy reduces the learning process for financial quantitative analysts by providing distributed computing resources and a pipeline-based programming model similar to classic standalone systems. Existing distributed computing frameworks, such as Apache Spark, require financial quantitative analysts to be proficient in distributed computing. The effectiveness of Alchemy was assessed at Huatai Securities in China. According to the findings, Alchemy could outperform current FQA systems by up to 300 times [1].

4.1.1.2 Year 2022:

Özgüven *et al*. (2022) presented a distributed system using the MapReduce paradigm for large-scale sports data aggregation and analytics. The authors described a Docker container-based big data architecture with Apache Spark and evaluated this architecture across four data-intensive case studies in sports analytics, including structured analysis, streaming, machine learning approaches, and graph-based analysis [11]. Jain *et al*. (2022) focused on the importance of features and feature engineering for telecommunication customer churn prediction models. The model employed a two-phase classification ensemble of Random Forest and Gradient Boosted Trees on the Cell2Cell dataset, which comprised 3,333 customers and 57 features. Python and Apache Spark were used for the implementation. A grid search was used for hyperparameter optimization. With Random Forest, the model's accuracy was 95%, while with Gradient Boosted Trees, it was 97% [12]. Pallamala & Rodrigues (2022) showed how to successfully test unstructured and semi-structured data from Apache Spark data samples using the Testing Whiz tool. When compared to Hadoop, Apache Spark produced faster testing results. The suggested method improved accuracy, boosted efficiency, lowered expenses, and saved time. The Extract, Transform, and Load (ETL) and Apache Spark processes were used to assess these approaches [3].

Jaya Lakshmi *et al*. (2022) suggested a health status prediction system to identify cardiovascular illnesses using patient tweets. Additional analytics were conducted on the distributed Apache Spark framework to decrease the time needed for training and testing. Streaming social media data was one of the main data sources. The program predicted cardiovascular risk by analyzing the characteristics of incoming user tweets. Two distinct corpora were used to assess the suggested framework's performance using an Extreme Learning Machine (ELM) Tree classifier. In terms of accuracy and time, the approach fared better than other classifiers including Decision Tree, Naïve Bayes, Linear SVC, and DNN [4]. Hasan *et al*. (2022) compiled a review from various researchers from 2010 to 2022. As the amount of data consistently generated reaches quintillions of bytes, it has become more crucial than ever to have a robust platform for powerful big data analysis. Apache Spark MLlib provides a range of exceptional capabilities for machine learning applications. A key finding of this study was that the performance and precision of Spark ML are fundamentally better than Spark MLlib. A dataset of bank client transactions was used in this correlation. The Apache Spark big data analysis engine

was used to launch a security-related data analysis [13]. Azeroual & Nikiforova (2022) presented security-relevant data analysis using the Apache Spark big data analytics engine incorporated into a business intelligence system. The k-means approach for clustering analysis in Spark's MLlib was used to create a prototype intrusion detection system that uses machine learning to identify data anomalies. Relevant data already owned by businesses and research institutions can be used to detect abnormal data [14].

Shaikh *et al*. (2022) introduced GeoFlink, an extension of Apache Flink that supports continuous queries on spatial data streams, spatial objects, and indexes. A grid-based index was provided to facilitate optimal data distribution and efficient spatial query processing. GeoFlink supported spatial range, spatial k-nearest neighbors (kNN), and spatial join queries on various spatial objects. In comparison to current state-of-the-art streaming platforms, GeoFlink achieved much greater query throughput. Besides Apache Flink, other state-of-the-art scalable stream processing platforms include Apache Spark Streaming and Apache Samza [15]. Tariq *et al*. (2022) proposed a model to assist e-businesses in predicting user churn using machine learning. A 2-D convolutional neural network (CNN) was used in the suggested model. Data was processed in a parallel setting using Apache Spark's distributed architecture. The Telco Customer Churn dataset from Kaggle was used as training data. The suggested model obtained an accuracy score of 0.963 out of 1. At 0.004, the training and validation loss was quite minimal. The confusion matrix results indicated a 94% true-negative rate and a 95% true-positive rate [16].

Ayub *et al*. (2022) provided a data pipeline system for processing video streams and images in a distributed setting using Apache Spark, Kafka, and OpenCV operating on commodity hardware. According to the experimental findings, the suggested method achieved effective video stream processing for face detection with increased throughput, scalability, and performance without requiring GPU-based hardware or cloud computing infrastructure [17]. Ataie *et al*. (2022) proposed and validated a hybrid approach (Apache Hadoop and Apache Spark) for the performance prediction of big data applications running in the cloud. This method used analytical modeling and machine learning approaches. The authors contrasted this method with Ernest, an ML-based method suggested by Spark's developers. According to the trials, Ernest was able to anticipate performance in interpolation scenarios with high accuracy, but was unable to do so in configurations with a growing number of cores. A comparison showed how the authors' proposed method significantly reduced prediction error, especially when only a few experiments were conducted on the real system [5]. Hagar & Gawali (2022) suggested three models (Apache Spark, LSTM, and CNN) to enhance the detection of all kinds of attacks. The CSE-CIC-IDS2018 dataset was used to train the prediction models for network-based intrusion detection. This dataset includes fourteen different types of attacks. Important features were chosen using Random Forest to reduce dimensionality; 19 out of 84 features were the outcome. Oversampling and undersampling methods were employed to lower the imbalance ratio. According to the trial results, the Apache Spark model yielded the best results in all 15 classes, with an accuracy of up to 100% for every class. For the F1-score, Apache Spark showed the highest result with 1.00 for most classes [18].

4.1.1.3 Year 2023:

Abhijith & Gundad (2023) developed a system based on machine learning algorithms to analyze sentiment in large datasets sourced from social media sites. A system leveraging Apache Spark was built and implemented using Naive Bayes and Support Vector Machines classification techniques. Accuracy was used to measure the performance of the algorithms. The algorithm proved quite effective in managing large sentiment datasets [19]. Shrotriya *et al*. (2023) examined a case study on how Spark is used in the healthcare sector to produce data-driven innovations and enhance patient care. The study explained how Spark is used in healthcare, including remote patient monitoring, medical image analysis, tailored treatment, predictive analytics for disease outbreaks, and EHR management. Spark has also improved cancer research and treatment, accelerated drug discovery, decreased hospital readmission rates, and detected sepsis earlier. Constraints pertaining to data security and privacy, infrastructure and scalability, data integration and quality, and talent shortages were also identified [20]. Lin & Lin (2023) proposed a new approach to build a bond price predictive model based on technical indicators in the financial market, applying machine learning techniques on the Apache Spark framework. This predictive model was constructed in three stages. First, the feature set was expanded by transforming the original price time series into technical indicators; features were then reduced using dimensionality reduction methods. Second, machine learning algorithms were used to build the predictive models. Finally, prediction results were compared by evaluating their MAE and RMSE. The data used was a competition dataset from Kaggle containing corporate bond transactions. The experimental results showed that the proposed approach outperformed the baseline results for bond price prediction [21].

Azeem *et al*. (2023) provided an introduction to deep learning in Mobile Big Data (MBD) analysis and verified the feasibility of a customizable learning architecture via Apache Spark. Each Spark worker trained a partial deep model on a shared MBD, and the parameters of all the partial models were then averaged to create a specialized deep model. Spark's architecture is especially well-suited for techniques that are very slow and computationally intensive, like deep learning [22]. Xiong *et al*. (2023) showed how the least-squares

Monte Carlo (LSMC) method in American option pricing could be made more accurate and efficient by using distributed computing with a divide-and-conquer strategy carried out by Apache Spark. This work was the first to investigate distributed computing technologies for LSMC improvement. Distributed computing offered benefits such as lowering computational complexity, avoiding intricate matrix transformations, and improving data security and privacy. Additionally, because pricing routes can be independently simulated and regressed, LSMC is appropriate for distributed computing [23]. Karimian-Aliabadi *et al.* (2023) presented a precise performance prediction model based on stochastic activity networks (SANs). The proposed model took into account multiple work queues, which is an essential component for attaining high accuracy in practical usage scenarios. The authors presented a monolithic analytical model that modeled each queue separately for a multi-queue YARN cluster running DAG-based Big Data applications. A fixed-point model was also offered to overcome the monolithic approach's limited scalability. A real-world cluster with the Apache Spark framework and the YARN scheduler was used to test the models. Tests using the TPC-DS benchmark demonstrated that the suggested model predicted Spark task execution times with an average inaccuracy of just 5.6%. The presented model enabled businesses to optimize their cluster configurations for specific workloads [24].

4.1.1.4 Year 2024:

Mendes *et al.* (2024) proposed MAS-Cloud+, a new agent-based architecture for forecasting, allocating, and tracking optimal cloud computing resources. Three reasoning models (heuristics, formal optimization, and metaheuristics) were implemented by MAS-Cloud+ for agents. By prioritizing user needs by taking time, cost, and resource waste into account, MAS-Cloud+ built Virtual Machines (VMs) while taking Service Level Agreements (SLAs) into account. A DNA sequence comparison application with varying workload sizes and a comparison with state-of-the-art work using Apache Spark benchmark programs running on AWS EC2 were utilized to validate MAS-Cloud+. The results demonstrated that while the heuristic model offered the best cost, the optimization model produced the best performance. These results showed that MAS-Cloud+ could lower the average cost of executing the WordCount, Sort, and PageRank BigDataBench benchmarking workloads by almost 58% [25]. Jose *et al.* (2024) compared the performance of Spark SQL and NoSQL databases, with an emphasis on the benefits of Spark SQL over NoSQL databases for streaming data exploration. Batch-stored streaming data was used to answer queries. The real-time streaming data from stock trading, website visitor activity, and mobile applications made this research crucial for business [26].

Bachir Belmehdi *et al.* (2024) offered a novel solution to the issue of choosing the best virtual data model for queries on big datasets. With the help of deep learning techniques, this method used features taken from SPARQL queries to predict the best virtual data model and was based on the principles of Ontology-Based Data Access (ODBA) to query and join large, heterogeneous data in a distributed fashion. Utilizing Apache Spark and GraphX, OPTIMA supported five out-of-the-box data source models: property graph, document-based, wide-column, relational, and tabular, stored in Neo4j, MongoDB, Cassandra, MySQL, and CSV, respectively. In-depth tests demonstrated that the method reduced query execution time by more than 40% for tabular model selections and more than 30% for graph model selections by returning the best virtual model with an accuracy of 0.831 [27]. La Gatta *et al.* (2024) extended CASTLE (Cluster-Aided Space Transformation for Local Explanations), a model-agnostic explainable AI methodology, to manage high-volume datasets. By combining local and global information, CASTLE sought to explain a predictive model's "black-box" behavior. To manage the volume, diversity, and velocity of big datasets, this extension made use of contemporary big data technologies such as Apache Spark. The outcomes showed that the suggested method effectively handled big datasets while preserving the high-quality explanations connected to CASTLE. Crucially, the method had a sub-linear rather than an exponential dependence on the size of the dataset, making it scalable for big data scenarios in financial, medical, and military sectors [28].

4.1.1.5 Year 2025:

Aladib *et al.* (2025) suggested a novel runtime verification framework to incorporate Linear Temporal Logic (LTL) monitoring into stream processing applications like Apache Spark. The method offered reusable LTL monitoring patterns that could be easily included into current streaming processes. LTL-based monitoring could successfully identify safety and liveness property violations while preserving stable latency, as shown by a case study applied to real-time financial data monitoring. Performance analysis showed that even though the method added computational overhead, it scaled well with growing data volumes. Beyond financial data processing, the suggested framework could be used in areas including industrial IoT analytics, financial fraud monitoring, and real-time equipment failure detection [29]. Esmaelizadeh *et al.* (2025) provided InfoMoD, a novel approach to machine learning model validation and debugging using current work in information-theoretic data summarization for model diagnostics. The authors used Apache Spark to develop InfoMoD in a distributed fashion to increase speed. It was demonstrated that InfoMoD produced expected performance indicators for every test example and provided a succinct explanation of model performance across multiple

data subsets based on four use cases, ranging from finance to computer vision and hate speech detection [30].

Trinh *et al*. (2025) provided an item-based collaborative filtering technique that made use of Apache Spark to overcome the scalability concerns of recommendation systems in cloud computing environments. The model maintained accuracy with a Root Mean Square Error (RMSE) of 0.9 while achieving a 7.6-fold training speedup over conventional single-machine methods by leveraging Spark's distributed computing capabilities. These findings showed how well distributed and parallel approaches work to create accurate and effective recommendation systems for extensive e-commerce applications [31]. Patil *et al*. (2025) provided a thorough examination of social media sentiment analysis approaches. These included both traditional machine learning methods like Naïve Bayes, Support Vector Machines, Decision Trees, and Random Forest, as well as more sophisticated deep learning models like Recurrent Neural Networks, Long Short-Term Memory Networks, Convolutional Neural Networks, and Transformer-based architectures like BERT and GPT. In addition, this review looked at natural language processing toolkits like NLTK, spaCy, TextBlob, and Stanford NLP, as well as big data frameworks like Hadoop, Apache Spark, and Apache Flink. The article also covered ML/DL frameworks like Scikit-learn, TensorFlow, PyTorch, and Keras, along with cloud and edge computing solutions [32].

Bompotas *et al*. (2025) presented CommC, a multifunctional commodity hardware cluster that reused unused computer capabilities to lower operating costs and increase hardware lifespan. A scalable, secure, and economical computing environment was produced by CommC through the integration of virtualization (KVM and Proxmox) with containerization (Kubernetes and Docker). The authors used big data engines like Apache Spark to illustrate CommC's adaptability. According to the cost analysis, by prolonging the life of current hardware, CommC lowered computing costs by up to 77.93% when compared to cloud-based alternatives. This greatly improved environmental sustainability [33]. Jiao (2025) explored through a comprehensive study how IoT devices—such as RFID tags, GPS trackers, and smart sensors—generated vast amounts of real-time data from enterprise assets including machinery, vehicles, and inventory. The data streams were collected and transmitted to a cloud-based platform, where big data tools like Apache Hadoop, Spark, and machine learning algorithms were used for processing and analysis. The author demonstrated how big data analytics enabled predictive insights, allowing for timely interventions in asset maintenance, lifecycle optimization, and precise cost allocation in accounting. Real-world case studies from various industries, including manufacturing and logistics, were examined. The author also discussed challenges such as data security, system interoperability, and the need for skilled personnel [34].

Ionescu *et al*. (2025) proposed a four-layer design to solve issues with financial data processing. Data sources, data integration, processing, and storage were among the layers in this design. Customer information, transaction records, and market feeds were all handled via the data ingestion process. Kafka recorded data in real time, and an ETL pipeline transformed it. The processing layer consisted of Apache Spark for real-time data analysis, Hadoop for batch processing, and a Machine Learning infrastructure that supported predictive modeling. The storage layer had a number of data tiering components to maximize access patterns. Test results demonstrated that environmental sustainability goals could be met through real-time processing of market data, compliance reporting, risk assessment, and customer analysis [35]. Vivek *et al*. (2025) proposed a new wrapper for online streaming feature subset selection (OSFSS) called OSFSS-W, intended especially for feature stream mining in the Apache Spark environment. The suggested approach included gradual learning, two vigilance tests to eliminate redundant and irrelevant characteristics, and a tolerance-based feedback mechanism that maintained and applied past knowledge. The authors also presented a logistic distribution-driven Bare Bones Particle Swarm Optimization (BBPSO-L) algorithm for optimization. Apache Spark parallelized BBPSO-L using an island-based methodology. The suggested algorithm's performance was assessed using datasets from the fields of finance, biology, and cybersecurity. The findings showed that the median AUC curve and the median cardinality were considerably improved across all datasets when two vigilance tests were used in conjunction with a tolerance-based feedback mechanism [36].

4.1.2 Sub-Topic Clustering

The authors cluster the sub-topics from the descriptions of the 36 scientific articles that were reviewed in the previous section regarding the use of Apache Spark in business and finance data engineering.

Table 1. The Sub-Topics Clustering

Topic	Sub-Topics and/or Algorithms Used	Author(s) and Year
Social Media Sentiment Analysis	Twitter with SMO-SVM	[7]
	Hybrid CNN-SVM	[8]
	Naive Bayes – SVM	[19]
	Multiple Algorithms	[32]
	Twitter with ELM Tree	[4]
Social Media Trend Analysis Prediction	Twitter	[9]
	Big Data in Cloud, Spark-Hadoop Integration	[5]
	Churn with Random Forest & Gradient Boosted Trees	[12]
	Churn with CNN	[16]
	Bond Price	[21]
	Option Price	[23]
	Performance/Workload/SLA with SAN	[24]
	Performance/Workload/SLA with MAS-Cloud+	[25]
	Performance/Workload/SLA with InfoMoD	[30]
Detection	Financial Fraud	[10]
	Anomaly	[14]
	Attacks	[18]
	Financial Security Breaches	[29]
	Bank Transaction Security	[13]
Other Advantages & Uses	Processing Semi-structured & Unstructured Data	[3]
	Efficient for Big Data	[17][22][28]
	Real-Time Updates with Spark SQL	[26]
	Healthcare Prediction & Detection	[20]
	Financial Data Processing	[35]
	Asset Accounting	[34]
	Recommendation Systems	[31]
	Efficiency with CommC	[33]
	Mining Feature Streams	[36]
	Data Accuracy and Timeliness	[11]
	Heterogeneous Data Access	[27]
	Data Shuffling/Partitioning Optimization	[2]
	Parsing XML	[6]
Alternatives to Spark	Alchemy	[1]
	GeoFlink with Apache Flink	[15]

4.2 Discussion

The systematic review of 36 articles published from 2021 to 2025 reveals significant trends and patterns in Apache Spark applications for business and finance data engineering. The chronological analysis demonstrates a clear evolution in research focus. Early works in 2021 concentrated on foundational optimization such as XML parsing [6], data shuffling and partitioning [2], and basic sentiment analysis [7][8]. By 2022-2023, research shifted toward specialized business applications including churn prediction [12][16], financial fraud detection [10], and financial instrument pricing [21][23]. Recent publications in 2024-2025 showcase advanced integration with emerging technologies such as runtime verification [29], explainable AI [28], IoT-based asset management [34], and modern data architectures [35].

Social Media Analytics emerged as a prominent research area with five studies [4][7][8][9][19], focusing on sentiment and trend analysis, predominantly using Twitter data. The algorithms employed ranged from traditional ML (SMO-SVM, Naive Bayes-SVM) to deep learning (CNN-SVM, ELM Tree), achieving 74-83% accuracy for trend analysis [9]. However, the concentration on Twitter represents a research gap for exploring other platforms such as Instagram, LinkedIn, TikTok, and Reddit. Prediction Tasks constituted the largest cluster, encompassing churn prediction [12][16], financial instrument pricing [21][23], and performance/workload prediction [5][24][25][30]. Ensemble methods (Random Forest + Gradient Boosted Trees) achieved 95-97% accuracy for churn prediction [12], while deep learning approaches (CNN) reached 96.3% [16]. The minimal accuracy difference suggests that simpler ensemble methods may be preferable due to better interpretability and lower computational requirements, especially in regulated industries requiring model explainability. Detection Applications focused on financial fraud [10], anomalies [14], cyber-attacks [18], and security breaches [13][29]. Notably, Hagar & Gawali (2022) achieved 100% accuracy using Apache

Spark for network intrusion detection after feature selection and dataset balancing, demonstrating Spark's effectiveness for security-critical applications [18]. Cost Optimization and Resource Management showed significant practical impact, with MAS-Cloud+ reducing costs by 58% [25], and CommC achieving 77.93% cost reduction compared to cloud alternatives [33], demonstrating that proper resource allocation strategies and commodity hardware utilization can substantially reduce operational expenses.

Several studies provided valuable comparative insights. Martinez-Mosquera *et al.* (2021) found Hive external tables faster for full queries, but Spark more flexible for complex queries despite requiring extra in-memory processing, suggesting framework selection should consider query complexity and data access patterns [6]. Hasan *et al.* (2022) identified Spark ML as fundamentally superior to MLlib in performance and precision, providing clear guidance for library selection in new implementations [13]. GeoFlink demonstrated higher query throughput than Spark Streaming for spatial data operations [15], while Alchemy achieved 300× speedup over conventional FQA systems for specialized financial analysis [1], indicating that domain-specific frameworks may outperform general-purpose Spark for specialized use cases. For churn prediction, the minimal accuracy difference between ensemble methods (95-97%) [12], and CNN (96.3%) [16], suggests that simpler ensemble methods may be preferable due to better interpretability and lower computational requirements.

Despite extensive research, several critical gaps remain. Sentiment analysis research heavily focuses on Twitter, neglecting other platforms that may provide different sentiment patterns and user demographics. There is limited exploration of state-of-the-art algorithms such as Transformer-based models (Temporal Fusion Transformer, Informer, Attention mechanisms) for financial time series forecasting. While mentioned conceptually [35], comprehensive empirical evaluation of Delta Lake implementation on Spark for financial data with ACID compliance is lacking. Insufficient research exists on Anti-Money Laundering (AML) detection using Spark GraphX for transaction network analysis, real-time credit risk scoring with Spark Structured Streaming, and Customer Lifetime Value (CLV) prediction using survival analysis or GLM. No studies explicitly addressed bias detection, fairness metrics (demographic parity, equalized odds), or ethical considerations in Spark-based ML models for financial services, despite increasing regulatory requirements. Additionally, there is an absence of research on privacy-preserving ML with federated learning on Spark for multi-institutional financial data collaboration.

The findings offer several actionable implications for practitioners. For framework selection, organizations should use Alchemy [1]or domain-specific platforms for specialized financial analysis with non-technical analysts, choose Spark for general-purpose big data processing with diverse use cases, and consider GeoFlink [15]for spatial data-intensive applications. Cost management can be achieved by implementing agent-based resource allocation (MAS-Cloud+) for 58% cost reduction [25], utilizing commodity hardware clusters (CommC) for 77.93% cost savings [33], and optimizing partition tuning while avoiding expensive shuffling [2]. For compliance and governance, organizations should integrate runtime verification frameworks [29] for regulatory compliance, adopt XAI frameworks like CASTLE [28] for model explainability requirements, and implement proper data lineage and audit trails for financial regulations. Algorithm selection should prefer ensemble methods (Random Forest, Gradient Boosted Trees) for churn prediction due to comparable accuracy and better interpretability, use feature selection techniques to reduce dimensionality and improve performance [18], and apply proper dataset balancing for imbalanced data [18]. Architecture design should implement four-layer architecture [35], for comprehensive financial data processing, use Kafka for real-time ingestion, Spark for real-time analysis, and Hadoop for batch processing, while considering Delta Lake for ACID compliance in financial data warehousing.

Regarding future research recommendations, the authors identify several potential sub-topics that could offer novelty. For sentiment and trend analysis on social media, previous researchers have already used hybrid CNN-SVM and Naive Bayes-SVM algorithms, with a focus on Twitter using SMO-SVM and Extreme Learning Machine (ELM) Trees [4][7][8][19]. Therefore, future research should investigate other social media platforms beyond Twitter and utilize other algorithms for training machine learning classification tasks. The same applies to sub-topics such as churn prediction, stock, bond, and option pricing, and the detection of fraud and anomalies, which should be explored with other algorithms that may yield better accuracy. Beyond existing topics, the utilization of Apache Spark in business and finance data engineering can be significantly expanded into more advanced areas. In the realm of predictive and risk analysis, research can be directed towards forecasting financial time series, such as stock prices or currency exchange rates, by implementing modern algorithms like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), or statistical models like ARIMA. Furthermore, Spark is highly reliable for in-depth credit risk analysis using models such as Logistic Regression and XGBoost to predict default probabilities. Another crucial topic is Anti-Money Laundering (AML) detection, where Spark's GraphX library can be used to map and analyze suspicious transaction networks, while algorithms like Isolation Forest can identify anomalies without requiring labeled data. In the domain of customer and marketing analytics, the focus can be shifted towards a more holistic understanding by exploring advanced customer segmentation using various clustering methods like K-Means, Gaussian Mixture Models

(GMM), or DBSCAN to group customers based on complex transactional behavior patterns. From there, predictive models can be built to calculate Customer Lifetime Value (CLV) using Generalized Linear Models (GLM), allowing companies to optimize retention and marketing strategies more effectively based on the long-term value of each customer. From a data engineering and architectural perspective, research can focus on constructing modern data infrastructures that serve as the foundation for all such analyses. A primary topic is the implementation of a Data Lakehouse architecture using technologies like Delta Lake on top of Spark. This approach combines the scalability of a data lake with the ACID transactional reliability of a data warehouse, enabling consistent, reliable, and auditable processing of financial data. This architecture, often powered by Spark Structured Streaming, ensures that data for risk analysis, fraud detection, and business prediction is always accurate and timely, addressing the challenges of data integrity at scale.

5. Conclusion

A Systematic Literature Review of the last five years (2021–2025) from Scopus reveals that previous authors have published scientific works on the use of Apache Spark in business and finance data engineering for sentiment and trend analysis of social media, with a particular dominance of Twitter. Apache Spark is also used for predicting customer churn, pricing stocks, bonds, and options, and detecting fraud and transactional anomalies, among other tasks. The review demonstrates a clear evolution from foundational optimization in 2021 toward specialized business applications in 2022-2023, and advanced integration with emerging technologies in 2024-2025. Apache Spark is widely used because it is a relatively low-cost and efficient platform, with studies showing cost reductions of up to 58-77.93% compared to cloud alternatives. Performance comparisons reveal that ensemble methods achieve 95-97% accuracy for churn prediction, while detection applications achieve up to 100% accuracy for network intrusion detection after proper feature selection and dataset balancing. Nevertheless, some authors have also presented other platforms as alternatives to Apache Spark, such as Alchemy, which achieved 300× speedup over conventional financial quantitative analysis systems, and GeoFlink, which demonstrated higher query throughput than Spark Streaming for spatial data operations.

Despite extensive research, several critical gaps remain. Sentiment analysis research heavily focuses on Twitter, neglecting other platforms that may provide different sentiment patterns. There is limited exploration of state-of-the-art algorithms such as Transformer-based models for financial time series forecasting. Comprehensive empirical evaluation of Delta Lake implementation on Spark for financial data with ACID compliance is lacking. No studies explicitly addressed bias detection, fairness metrics, or ethical considerations in Spark-based ML models for financial services, despite increasing regulatory requirements. Additionally, there is an absence of research on privacy-preserving ML with federated learning on Spark for multi-institutional financial data collaboration.

Future research is advised to investigate social media platforms other than Twitter and/or utilize other algorithms for training machine learning classification tasks. Similarly, other sub-topics should be explored with different algorithms that may yield better accuracy. Furthermore, the authors also recommend that Apache Spark be utilized for forecasting financial time series using modern algorithms like LSTM and GRU, in-depth credit risk analysis using models such as Logistic Regression and XGBoost, Anti-Money Laundering (AML) detection using Spark's GraphX library for transaction network analysis, advanced customer segmentation using clustering methods like K-Means, GMM, or DBSCAN, and Customer Lifetime Value (CLV) prediction using Generalized Linear Models. From a data engineering perspective, the implementation of Data Lakehouse architectures using technologies like Delta Lake on top of Spark is recommended, as this approach combines the scalability of a data lake with the ACID transactional reliability of a data warehouse, enabling consistent, reliable, and auditable processing of financial data. These recommendations aim to advance Apache Spark applications in business and finance toward greater accuracy, efficiency, interpretability, and regulatory compliance.

References

[1] Gu, R., Zhang, X., Gao, H., Huang, Z., Chen, H., & Wang, C. (2021). Alchemy: Distributed financial quantitative analysis system with high-level programming model. *Software: Practice and Experience*, *51*(8), 1676–1699. https://doi.org/10.1002/spe.2982

[2] Aziz, K., Zaidouni, D., & Bellafkih, M. (2021). Leveraging performance to optimize data shuffling/partitioning for in-memory computation in Apache Spark. *International Journal of Computer Science and Applications*, *18*(1), 69–84.

[3] Pallamala, R. K., & Rodrigues, P. (2022). An investigative testing of structured and unstructured data formats in big data application using Apache Spark. *Wireless Personal Communications*, *122*(1), 603–620. https://doi.org/10.1007/s11277-021-08915-0

[4] Jaya Lakshmi, A., Venkatramaphanikumar, S., & Kolli, V. K. K. (2022). Prediction of cardiovascular risk using extreme learning machine-tree classifier on Apache Spark cluster. *Recent Advances in Computer Science and Communications*, *15*(3), 443–455. https://doi.org/10.2174/2666255813999200904163404

[5] Ataie, E., Evangelinou, A., Gianniti, E., & Ardagna, D. (2022). A hybrid machine learning approach for performance modeling of cloud-based big data applications. *The Computer Journal*, *65*(12), 3123–3140. https://doi.org/10.1093/comjnl/bxab131

[6] Martinez-Mosquera, D., Navarrete, R., & Luján-Mora, S. (2021). Efficient processing of complex XSD using Hive and Spark. *PeerJ Computer Science*, *7*, 1–33. https://doi.org/10.7717/peerj-cs.652

[7] Lijo, V. P., & Seetha, H. (2021). Tweets sentiment analysis using multi-lexicon features and SMO. *International Journal of Embedded Systems*, *14*(5), 476–485. https://doi.org/10.1504/IJES.2021.120264

[8] Raviya, K., & Mary Vennila, S. (2021). An implementation of hybrid enhanced sentiment analysis systemusing Spark ML pipeline: A big data analytics framework. *International Journal of Advanced Computer Science and Applications*, *12*(5), 323–329. https://doi.org/10.14569/IJACSA.2021.0120540

[9] Rodrigues, A. P., Fernandes, R., Bhandary, A., Shenoy, A. C., Shetty, A., & Anisha, M. (2021). Real-time Twitter trend analysis using big data analytics and machine learning techniques. *Wireless Communications and Mobile Computing*, *2021*, Article 3920325. https://doi.org/10.1155/2021/3920325

[10] Zhou, H., Sun, G., Fu, S., Wang, L., Hu, J., & Gao, Y. (2021). Internet financial fraud detection based on a distributed big data approach with Node2vec. *IEEE Access*, *9*, 43378–43386. https://doi.org/10.1109/ACCESS.2021.3062467

[11] Özgüven, Y. M., Gönener, U., & Eken, S. (2022). A Dockerized big data architecture for sports analytics. *Computer Science and Information Systems*, *19*(2), 957–978. https://doi.org/10.2298/CSIS220118010O

[12] Jain, H., Khunteta, A., & Srivastava, S. (2022). Telecom churn prediction using an ensemble approach with feature engineering and importance. *International Journal of Intelligent Systems and Applications in Engineering*, *10*(3), 22–33.

[13] Hasan, Z., Xing, H.-J., & Magray, M. I. (2022). Big data machine learning using Apache Spark MLlib. *Mesopotamian Journal of Big Data*, *2022*, 1–11. https://doi.org/10.58496/MJBD/2022/001

[14] Azeroual, O., & Nikiforova, A. (2022). Apache Spark and MLlib-based intrusion detection system or how the big data technologies can secure the data. *Information*, *13*(2), Article 58. https://doi.org/10.3390/info13020058

[15] Shaikh, S. A., Kitagawa, H., Matono, A., Mariam, K., & Kim, K.-S. (2022). GeoFlink: An efficient and scalable spatial data stream management system. *IEEE Access*, *10*, 24909–24935. https://doi.org/10.1109/ACCESS.2022.3154063

[16] Tariq, M. U., Babar, M., Poulin, M., & Khattak, A. S. (2022). Distributed model for customer churn prediction using convolutional neural network. *Journal of Modelling in Management*, *17*(3), 853–863. https://doi.org/10.1108/JM2-01-2021-0032

[17] Ayub, U., Ahsan, S. M., & Qureshi, S. M. (2022). Scalable big data pipeline for video stream analytics over commodity hardware. *KSII Transactions on Internet and Information Systems*, *16*(4), 1146–1165. https://doi.org/10.3837/tiis.2022.04.004

[18] Hagar, A. A., & Gawali, B. W. (2022). Apache Spark and deep learning models for high-performance network intrusion detection using CSE-CIC-IDS2018. *Computational Intelligence and Neuroscience*, *2022*, Article 3131153. https://doi.org/10.1155/2022/3131153

[19] Abhijith, G. S. V., & Gundad, A. K. V. (2023). Data mining for emotional analysis of big data. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(3s), 271–279.

[20] Shrotriya, L., Sharma, K., Parashar, D., Mishra, K., Rawat, S. S., & Pagare, H. (2023). Apache Spark in healthcare: Advancing data-driven innovations and better patient care. *International Journal of Advanced Computer Science and Applications*, *14*(6), 608–616. https://doi.org/10.14569/IJACSA.2023.0140665

[21] Lin, S.-Y., & Lin, H.-Y. (2023). Bond price prediction using technical indicators and machine learning techniques. *Journal of Information Science and Engineering*, *39*(2), 439–455. https://doi.org/10.6688/JISE.202303_39(2).0012

[22] Azeem, M., Abualsoud, B. M., & Priyadarshana, D. (2023). Mobile big data analytics using deep learning and Apache Spark. *Mesopotamian Journal of Big Data*, *2023*, 16–28. https://doi.org/10.58496/MJBD/2023/003

[23] Xiong, L., Luo, J., Vise, H., & White, M. (2023). Distributed least-squares Monte Carlo for American option pricing. *Risks*, *11*(8), Article 145. https://doi.org/10.3390/risks11080145

[24] Karimian-Aliabadi, S., Aseman-Manzar, M.-M., Entezari-Maleki, R., Ardagna, D., Egger, B., & Movaghar, A. (2023). Fixed-point iteration approach to Spark scalable performance modeling and evaluation. *IEEE Transactions on Cloud Computing*, *11*(1), 897–910. https://doi.org/10.1109/TCC.2021.3119943

[25] Mendes, A. H. D., Rosa, M. J. F., Marotta, M. A., Araujo, A., Melo, A. C. M. A., & Ralha, C. G. (2024). MAS-Cloud+: A novel multi-agent architecture with reasoning models for resource management in multiple providers. *Future Generation Computer Systems*, *154*, 16–34. https://doi.org/10.1016/j.future.2023.12.022

[26] Jose, B., Rajesh, N., & Joseph, L. (2024). Enhanced query performance for stored streaming data through structured streaming within Spark SQL. *Indonesian Journal of Electrical Engineering and Computer Science*, *35*(3), 1744–1750. https://doi.org/10.11591/ijeecs.v35.i3.pp1744-1750

[27] Bachir Belmehdi, C. B., Khiat, A., & Keskes, N. (2024). Predicting an optimal virtual data model for uniform access to large heterogeneous data. *Data Intelligence*, *6*(2), 504–530. https://doi.org/10.1162/dint_a_00216

[28] La Gatta, V., Moscato, V., Postiglione, M., & Sperlì, G. (2024). An eXplainable artificial intelligence methodology on big data architecture. *Cognitive Computation*, *16*(5), 2642–2659. https://doi.org/10.1007/s12559-024-10272-6

[29] Aladib, L., Su, G., & Yang, J. (2025). Real-time monitoring of LTL properties in distributed stream processing applications. *Electronics*, *14*(7), Article 1448. https://doi.org/10.3390/electronics14071448

[30] Esmaelizadeh, A., Cotterill, S., Hebert, L., Golab, L., & Taghva, K. (2025). InfoMoD: Information-theoretic machine learning model diagnostics. *Distributed and Parallel Databases*, *43*(1). https://doi.org/10.1007/s10619-024-07450-8

[31] Trinh, T., Nguyen, V.-H., Nguyen, N., & Nguyen, D.-N. (2025). Product collaborative filtering based recommendation systems for large-scale e-commerce. *International Journal of Information Management Data Insights*, *5*(1), Article 100322. https://doi.org/10.1016/j.jjimei.2025.100322

[32] Patil, S. S., Suryawanshi, V. P., Patil, S. M., Girase, S. P., & Bhagat, D. A. (2025). Review of sentiment analysis in social media using big data: Techniques, tools, and frameworks. *International Journal of Basic and Applied Sciences*, *14*(2), 34–48. https://doi.org/10.14419/mhv83077

[33] Bompotas, A., Kalogeropoulos, N.-R., & Makris, C. (2025). CommC: A multi-purpose commodity hardware cluster. *Future Internet*, *17*(3), Article 121. https://doi.org/10.3390/fi17030121

[34] Jiao, S. (2025). Utilization of the Internet of Things and big data for enterprise asset management and accounting. *International Journal of High Speed Electronics and Systems*. https://doi.org/10.1142/S0129156425402505

[35] Ionescu, S.-A., Diaconita, V., & Radu, A.-O. (2025). Engineering sustainable data architectures for modern financial institutions. *Electronics*, *14*(8), Article 1650. https://doi.org/10.3390/electronics14081650

[36] Vivek, Y., Ravi, V., & Krishna, P. R. (2025). Online feature subset selection for mining feature streams in big data via incremental learning and evolutionary computation. *Swarm and Evolutionary Computation*, *94*, Article 101896. https://doi.org/10.1016/j.swevo.2025.101896.