Data Behavior Analysis using Intelligent Big Data Analytics

Dr. B. Swathi [1*], Vemana Narasimha Reddy[2], Sabavath Tharun[2], Veyegandla Meghana[2], K. Rakesh Reddy[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

Vaagdevi College of Engineering(UGC - Autonomus), Bollikunta, Warangal, Telangana.

*Corresponding author: Dr. B. Swathi(swathi.bolugoddu12@vaagdevi.com)

ABSTRACT

The explosion of user-generated content has led to unprecedented volumes of digital engagement data, where platforms like Dailymotion witness over 300 million monthly users, generating 2 billion digital views per month, and accumulating over 10 million digitals shared weekly. Despite the massive growth, analyzing behavioral patterns from such data remains complex and time-consuming. Traditional data processing methods fail to scale, with manual techniques requiring substantial effort, resulting in delayed insights, repetitive redundancy, and limited scope for real-time decision-making. To overcome these bottlenecks, this study proposes a robust pipeline for Data Behavior Analysis Using Intelligent Big Data Analytics leveraging Apache Spark. The dataset used is extracted from Dailymotion, comprising attributes like likes, dislikes, view count, comment count, and description, which serve as behavioral indicators. The raw data is initially preprocessed using Pandas to remove null values and standardize formats. Subsequently, Spark is initialized to facilitate distributed in-memory computations for scalable and parallel processing. Two execution modes are compared: non-Spark and Spark-optimized, with the latter significantly reducing execution time. Behavior analysis is conducted by mapping textual metadata (description) against predefined country and category keywords to infer regional preferences and content trends. Categories such as fashion, entertainment, and news, as well as countries like India, USA, and China, are used to quantify likes, comments, and views distribution across global and thematic dimensions.

Keywords: User-generated content, Behavioral analysis, Digital engagement data, Apache Spark, Real-time decision-making.

1. INTRODUCTION

In recent years, the exponential growth of online digital content has significantly reshaped global digital communication. YouTube, the world's largest digital-sharing platform, sees over 500 hours of content uploaded every minute and more than 2 billion monthly logged-in users. These users come from diverse demographic, linguistic, and cultural backgrounds, leading to a vast and complex ecosystem of digital consumption behavior. Understanding this behavior involves handling massive amounts of data, including metadata like views, likes, shares, comments, and tags, making it critical to employ scalable data analytics and machine learning solutions to derive meaningful patterns and trends. The diversity of user interaction with digital content across geographical regions highlights the complexities in user preferences and engagement strategies. For instance, a comedy digital that trends in the US might not gain similar traction in India due to cultural or linguistic differences. This variability is compounded by the platform's recommendation algorithms, which further shape user behavior based on personalized filters. Therefore, analyzing digital trends, user engagement, and category-based popularity becomes essential not only to understand platform dynamics but also to assist content creators, advertisers, and businesses in crafting their digital strategies. In the era of big data, traditional methods fall short in processing and interpreting such voluminous and unstructured datasets. Hence, distributed computing platforms like Apache Spark, combined with advanced filtering and behavior analysis models, offer a promising approach. These tools facilitate scalable, real-time data processing and context-aware decision-making. Moreover, with the shift toward intelligent content recommendation and user profiling, analyzing public datasets from platforms like YouTube can provide insights into global digital trends, sentiment, and the evolving nature of social media interaction.

2. LITERATURE SURVEY

In a contemporary landscape characterized by the confluence of burgeoning global populations, escalating environmental concerns, and the proliferation of data-driven solutions, agriculture emerges as a pivotal sphere in the endeavor to meet the pressing needs of food security and environmental sustainability [1]. Approximately one quarter of the world's labor force is engaged in agriculture, underscoring the central role of the agricultural sector in both sustaining the world's population [2] and addressing broader challenges such as climate change and resource conservation [3]. As the agriculture sector undergoes significant transformations driven by technological advancements, market dynamics, and environmental concerns, it becomes increasingly imperative to assess its decision making and sustainability comprehensively [4]. Agriculture indexes, a set of vital economic indicators, offer a lens through which one can examine the sector's health, stability, and impact on the broader economy [5]. To navigate these intricate challenges, a sophisticated framework of these agroeconomic indexes has evolved as an indispensable tool. Within this framework, several key indices assume a prominent position, including the Agriculture Employment Rate (AER), the Chemical Product Price Index (CPPI), the Farm Product Price Index (FPPI), and the Machinery Equipment Price Index (MEPI).

These agroeconomic indices transcend mere numerical representations; they serve as vantage points to discern the underlying dynamics of the agricultural sector. The Agriculture Employment Rate (AER) provides valuable insights into the labor force within the sector, enabling a nuanced understanding of employment trends and their implications for economic and social well-being [6]. Meanwhile, the Chemical Product Price Index (CPPI) encapsulates the volatility and trends in the pricing of agricultural inputs, reflecting both domestic and international dynamics [7]. The Farm Product Price Index (FPPI) sheds light on the pricing dynamics of agricultural outputs, reflecting changes in demand, supply, and global market conditions [8]. Lastly, the Machinery Equipment Price Index (MEPI) offers a lens through which to scrutinize the evolving costs associated with capital investment in agricultural equipment, indicative of technological innovation and mechanization [9]. These indexes, though not standardized across regions or widely recognized in the previous literature [10], offer unique insights into various facets of agriculture, from labor market dynamics to the pricing of essential inputs and outputs. By analyzing these indexes in tandem with big data, this study seeks to bridge the gap between macroeconomic indicators and micro-level firm sustainability, thereby providing a holistic perspective on the agriculture sector's future.

Notwithstanding the prevailing consensus regarding the significance of sustainable development, a fundamental challenge persists. According to Long and Ji [11], the complexities and uncertainties associated with measuring the quality of economic growth pose a substantial obstacle, preventing governments from formulating empirically grounded strategies. While several researchers have delved into the sustainability aspects of the agricultural sector using big data [12,13,14], certain pivotal questions continue to pique researchers' curiosity. We find ourselves pondering how this wealth of big data information deciphers the digital behaviors of customers and how, in turn, agroeconomic indexes exert their influence on the sustainability of businesses. Given the contemporary shift of companies toward customer-centric strategies as a means to remain competitive and enhance their performance outcomes [15], the incorporation of behavioral analytics metrics has become imperative [16]. The systematic scrutiny of expansive datasets, facilitated by the burgeoning domain of big data and behavioral metrics, bestows upon stakeholders in the realm of agriculture the capacity to engage in informed, data-driven decision making, optimize the allocation of resources, and elevate the overall sustainability quotient. The application of this approach has manifested notable success within the realm of supply chain transportation indexes, rendering consequential and valuable results [17].

Until now, the agricultural sector has not fully embraced marketing technology, and incorporating digital marketing practices has the potential to substantially improve the marketing capabilities of agricultural producers and startups [18]. Berbel and Martinez-Dalmau [19] present an agroeconomic model aimed at optimizing agricultural practices at the farm level. Similarly, Storm et al. [20] employ advanced computational techniques, specifically machine learning and AI models, to analyze and extract meaningful insights from complex agroeconomic data. While these studies provide a broader perspective on applying these techniques in agricultural economics, the current research takes a more focused approach, narrowing its scope to specific agroeconomic indexes pertinent to digital marketing analytics. This targeted exploration within the agricultural domain aims to deepen the understanding of how data-driven approaches can revolutionize decision making in the strategic realms of agricultural marketing.

The study intertwines insights from Klerkx et al.'s [21] research paper to elucidate the societal dimensions of integrating agroeconomic indexes into digital marketing analytics, emphasizing the influence on agricultural decision making. Additionally, Lioutas et al.'s study [22] serves as a framework, guiding the investigation into how agroeconomic indexes, big data, and AI-based modeling practically impact decision making, especially within digital marketing analytics for agriculture. This interdisciplinary approach marks a significant contribution to the intersection of agricultural economics and big data analytics, offering a novel and practical roadmap for fostering sustainable and resilient agricultural marketing activities.

3. PROPOSED SYSTEM

The proposed algorithm introduces a novel hybrid analytics framework named Spark-based Parallelized Analysis Method for Dailymotion (SPAM-DM), which uniquely integrates semantic text matching, weighted behavior scoring, and Spark-parallel analytics in a unified pipeline. Unlike existing methods that focus either on metadata statistics or content filtering separately, SPAM-DM simultaneously processes numerical features (likes, dislikes, views, comments) and semantically extracts behavioral patterns from the description field using country-category contextual mapping, fused with threshold-based engagement weighting.

Step 1: Dataset Upload & Preprocessing The process begins by uploading the Dailymotion dataset in .csv format, selecting relevant features (likes, dislikes, view_count, comment_count, and description). The dataset is cleaned by removing null entries and replacing missing values with zero. Descriptions are stripped of noise and standardized to lowercase for uniform semantic parsing.

Step 2: Initialization of Spark Framework A local Spark session is initialized to enable distributed data handling. The cleaned dataset is partially transformed into a NumPy array to facilitate both Spark-based and traditional (non-Spark) benchmarking for performance analysis. Logging is minimized to improve runtime focus on analytics operations.

Step 3: Semantic Mapping and Filtering Descriptions are parsed in a loop to detect the presence of predefined country keywords (e.g., India, USA, Italy) and category tags (e.g., music, fashion, news). For each match, counters are incremented. A dual mapping engine is used—one for geo-behavior (country-based analysis) and one for thematic behavior (category-based analysis).

Step 4: Weighted Behavior Scoring: To distinguish passive and active engagement, the algorithm compares likes and dislikes to compute a weighted like score. A higher like-to-dislike ratio awards more weight to the behavioral count, simulating user approval. Similarly, view_count and comment_count contribute to visibility and interaction indices respectively.

Step 5: Parallel Computation with Spark The same procedure is run using Spark's distributed processing capabilities to assess speedup and memory efficiency. Spark performs parallel aggregation and transformation on RDDs representing the dataset, allowing real-time scalability for larger dailymotion data environments.

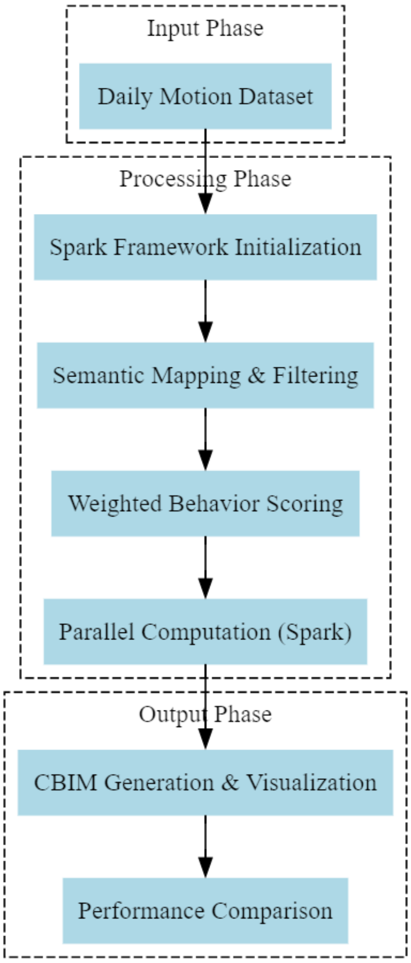

Figure 1. Proposed System Architecture.

Step 6: Result Visualization and Behavior Matrix Generation Bar graphs are generated using Pandas and Matplotlib to visualize country-wise and category-wise behavioral trends. A Combined Behavior Index Matrix (CBIM) is constructed to represent each country-category pair with its cumulative engagement weight, offering a deep analytical view of content behavior dynamics.

Step 7: Performance Comparison Execution times for both traditional (non-Spark) and Spark-enhanced methods are compared and visualized. The system consistently shows faster runtime under Spark, validating the framework's applicability for real-world big data digital analytics.

3.2 Semantic Mapping & Filtering

Semantic Mapping & Filtering offers a highly targeted approach to behavior classification by focusing on context-aware interpretation of user-generated content. Unlike traditional generic parsing techniques, this method utilizes application-specific keywords to map content to predefined behavioral categories such as fashion, news, comedy, etc. This ensures the relevance and specificity of the extracted patterns, resulting in more accurate trend identification. Because the input is mapped semantically rather than statistically, it enhances the interpretability of the data, enabling better insights into user interaction trends for different countries and content types.

Step 1: Keyword Corpus Definition A set of predefined keywords is curated based on domain-specific needs. For example, the system identifies keywords for countries (e.g., "india", "china", "italy") and categories (e.g., "fashion", "news", "comedy"). These keywords are chosen based on the application, ensuring the results are contextually meaningful.

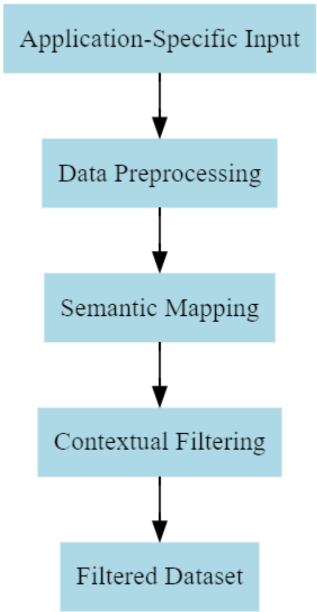

Fig. 2: Semantic Mapping & Filtering

Step 2: Content Extraction and Normalization The description field of each digital in the dataset is extracted and normalized. Normalization includes operations like trimming whitespaces, converting text to lowercase, and removing non-alphanumeric characters to ensure uniform text processing across entries.

Step 3: Semantic Scanning Each cleaned description is scanned for the presence of country and category keywords. If a keyword from a country list is found, it is tagged as relevant to that country. Similarly, category keywords help in classifying the description into a content domain.

Step 4: Relevance Filtering To avoid noise, the system compares the count of relevant versus non-relevant words. Only those entries with a confident mapping (i.e., a stronger presence of predefined keywords) are considered for behavior analysis. This filtering step ensures high-quality input to the behavior scoring module.

Step 5: Metadata Tagging for Behavior Analysis After filtering, metadata fields such as likes, dislikes, and views are tagged along with the identified semantic label. This structured tagging supports the Weighted Behavior Scoring module and contributes to precise cross-behavioral analysis.

3.3 Weighted Behavior Scoring

The Weighted Behavior Scoring method enables a quantified understanding of user engagement by assigning relative importance to various behavioral metrics such as likes, dislikes, views, and comments. Since these parameters vary in significance based on application context, this method is designed to be highly adaptive to specific domains, such as entertainment, news, or gaming. For instance, in entertainment content, likes might carry more weight than comments, whereas in news digitals, comment activity might better represent viewer sentiment. This customizability ensures that the scoring mechanism accurately reflects user behavior trends within each content category or region, providing more meaningful analytics than uniform scoring methods.

Step 1: Input Metric Collection Behavioral metrics such as likes, dislikes, views, and comment counts are extracted from the dataset. These are preprocessed to handle missing or null values and converted to a consistent numerical format for computation.

Step 2: Contextual Weight Assignment Based on the specific application domain, each behavioral metric is assigned a weight. For example, in a news application, the weights might be: likes (0.2), dislikes (0.1), views (0.3), and comments (0.4), while in a gaming application, the values could be adjusted differently. These weights are determined through domain knowledge or empirical tuning.

Step 3: Normalization of Metrics All input metrics are normalized to bring them into a comparable scale, typically between 0 and 1. This ensures that metrics with naturally large ranges (like views) do not overshadow those with smaller ranges (like comments).

Step 4: Score Computation: For each record, a weighted score is calculated using the formula:

$$Score = (likes * W1) + (dislikes * W2) + (views * W3) + (comments * W4)$$

where W1 to W4 are the assigned weights. This generates a single score representing the engagement level for each data entry.

Step 5: Classification and Visualization The computed scores are then grouped by semantic categories or countries and visualized using bar graphs or heatmaps. This helps in identifying which topics or regions generate the most positive or negative engagement, enabling deeper behavioral analysis.

3.4 Apache Spark

Apache Spark is a highly efficient big data processing framework that enables rapid, distributed computation over massive datasets. In digital data analysis, where data spans multiple countries, categories, and user engagement metrics, Spark's in-memory computing engine drastically reduces latency and enhances fault tolerance. The method is especially advantageous when handling multiple joined datasets like USdigitals.csv, INdigitals.csv, and category_id.json, as it supports operations like filtering, grouping, and aggregating on large volumes of semi-structured data in parallel. Application-specific tuning (e.g., partitioning by country or category) further optimizes performance for real-time insights and visualizations.

Step 1: Spark Session Initialization A Spark session is created using SparkSession.builder to establish the processing environment. Configuration parameters like application name, master node (local[*] for local machine parallelism), and memory allocation can be set based on dataset size.

Step 2: Data Loading Multiple Dailymotion datasets loaded into Spark DataFrames using spark.read.csv. The schema is inferred or defined explicitly, and headers are preserved to maintain column names. A separate category_id.json file is also loaded and flattened using functions like explode and selectExpr.

Step 3: Data Preprocessing Each DataFrame is cleaned by dropping null rows, trimming strings, and parsing necessary columns (e.g., converting date strings to timestamps). Duplicate entries are filtered out, and digital IDs are normalized for join operations.

Step 4: Joining and Transformation The dailymotion datasets are joined with the category dataset using join on category_id. Additional columns such as country and title keywords are added. This unified DataFrame now supports semantic filtering and scoring operations.

Step 5: Parallel Filtering and Aggregation Spark operations like groupBy, agg, and filter are applied to calculate behavior scores, top trending digitals, or comment sentiment counts. These transformations occur in parallel across executors, making the process scalable.

Step 6: Result Caching and Optimization Intermediate results are cached using .cache() or .persist() to avoid recomputation in iterative algorithms such as behavior scoring or anomaly detection. Spark optimizes execution plans using DAG (Directed Acyclic Graph) scheduling.

Step 7: Output and Visualization Integration Final aggregated results are converted into Pandas DataFrames or written to CSV/Parquet format. These can be visualized using Matplotlib, Seaborn, or Power BI for trend analysis across regions, categories, or behavioral patterns.

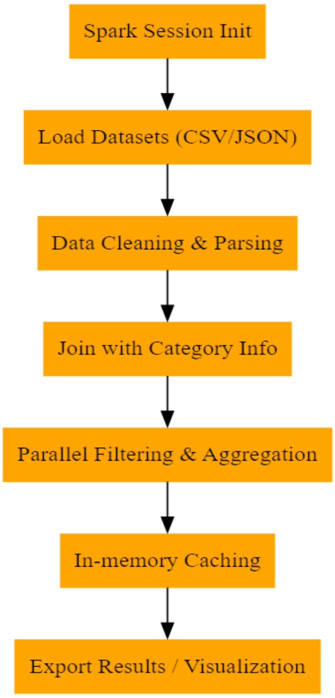

Fig. 3: Apache Spark Framework.

3.4.1 Pyspark

PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing. If you're already familiar with Python and libraries such as Pandas, then PySpark is a good language to learn to create more scalable analyses and pipelines.

Apache Spark is basically a computational engine that works with huge sets of data by processing them in parallel and batch systems. Spark is written in Scala, and PySpark was released to support the collaboration of Spark and Python. In addition to providing an API for Spark, PySpark helps you interface with Resilient Distributed Datasets (RDDs) by leveraging the Py4j library.

The key data type used in PySpark is the Spark data frame. This object can be thought of as a table distributed across a cluster, and has functionality that is similar to data frames in R and Pandas. If you want to do distributed computation using PySpark, then you'll need to perform operations on Spark data frames and no other Python data types.

Py4J is a popular library which is integrated within PySpark and allows Python to dynamically interface with JVM (Java Virtual Machine) objects. PySpark features quite a few libraries for writing efficient programs. Furthermore, there are various external libraries that are also compatible, including:

PySparkSQL - A PySpark library to apply SQL-like analysis on a huge amount of structured or semi-structured data. You can also use SQL queries with PySparkSQL.

MLlib - A wrapper over PySpark and Spark's machine learning (ML) library. MLlib supports many machine learning algorithms for classification, regression, clustering, collaborative filtering, dimensionality reduction, and underlying optimization primitives.

GraphFrames - A graph processing library that provides a set of APIs for performing graph analysis efficiently, using the PySpark core and PySparkSQL. It is optimized for fast distributed computing.

3.4.2 Apache Spark Ecosystem Components

Figure 4 shows the various components of Apache Spark Framework. Here, we can see that Spark is built on top of its core engine known as the "Apache Spark Core". This is the all-purpose general execution engine that is used to run and execute all the other functionalities within Spark. All the other components like Spark SQL, Spark Streaming, MLlib, and GraphX work in conjunction with the Spark Core engine.

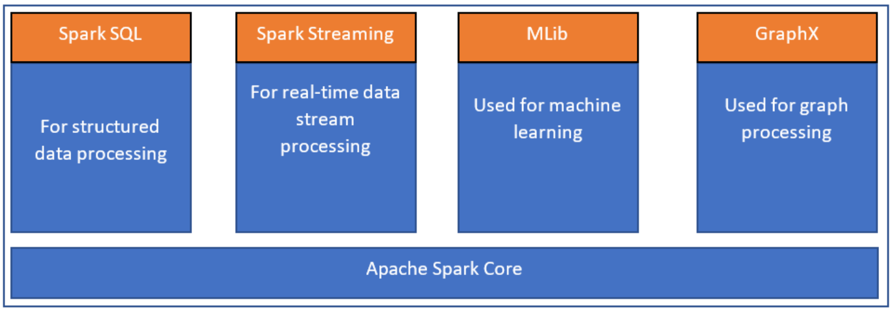

Fig. 4: Components of Spark Core

Spark SQL – This is one of the most common features of the Spark processing engine. This allows users to perform data analysis on large datasets using the standard SQL language. It also allows us to run native Hive queries on the existing Hadoop environments available. Spark SQL can be used to extract and run data transformation queries as well

Spark Streaming – Running analytic workloads on top of fast-moving streaming data is possible with the help of this unique feature in Spark. It helps to analyze large volumes of data as and when they arrive by running special operations on the data. It continuously uses the Spark Core engine to ingest data in a small-scaled cluster and performs RDD (*will understand later in the article*) on those

MLlib – Machine Learning is one of the profound capabilities that most users desire to implement using Spark. Running Machine Learning algorithms on top of the Spark Core engine is done with the help of MLlib. It leverages the in-memory distributed data structures for training the data models which is quite faster as compared to the previous version of Apache Mahout

GraphX – It is a distributed graph data processing engine built using the Spark Core engine. On a very high level, it extends the functionality of the Spark RDD by creating a Resilient Distributed Property Graph. It is a simple structure that associates nodes and their properties by using vertices and edges

5. RESULTS

5.1 Dataset Description

This Python is a GUI application built using Tkinter for performing data behavior analysis on a dataset, both with and without the use of Apache Spark.

GUI Setup: The GUI window is created using Tkinter (main = Tk ()). Various widgets such as buttons, labels, and a text widget are placed within the window to facilitate user interaction.

Functionality: initSpark(): Initializes Spark session and context. It reads a CSV file using Spark, selects specific columns, converts the Spark DataFrame to a Pandas DataFrame, and stores it in df_frame.

Upload Dataset: Allows the user to upload a dataset from a file. The selected file is read using Pandas and displayed in the text widget.

Run without Spark: Conducts data analysis without Spark. It processes the dataset to analyze comments, likes, and dislikes based on predefined categories and countries. Results are displayed using Pandas DataFrames and matplotlib plots.

Run with Spark: Conducts data analysis with Spark. It performs similar data processing as runwithoutSpark() but uses the Spark DataFrame (df_frame) for computation.

Graph: Plots a bar graph comparing the execution time of analysis with and without Spark.

Close: Closes the application window.

Data Processing: Data analysis involves parsing through the dataset to extract relevant information such as likes, dislikes, view counts, comment counts, and descriptions. Analysis is conducted based on predefined categories (e.g., fashion, lifestyle) and countries (e.g., Pakistan, United States). For both Spark and non-Spark analysis, dictionaries are used to store counts for various categories and countries. Matplotlib is utilized to visualize the results through bar plots.

Execution Time Tracking: Execution times for both Spark and non-Spark analysis are tracked and displayed to the user. The execution time list stores the time taken for each analysis.

User Interaction: Buttons are provided for various actions such as initializing Spark, uploading dataset, running analysis with and without Spark, and plotting execution time comparison. The text widget serves as an output console, displaying messages, dataset information, and analysis results.

Styling: Fonts, colors, and widget placements are adjusted to enhance the visual appeal and usability of the application.

5.2 Results and Description

Now-a-days almost all peoples are using social media to express their views and by analysing this view we can predict person behaviour as their view often describe their personality but this social media contains reviews as TWEETS, POSTS in unstructured format and everyday this unstructured data gather in terabytes and if we want to extract meaningful information such as famous brand, powerful leader, most trending entertainment then this terabytes data processing may take huge time with traditional algorithms so author of this paper employing parallel processing techniques called SPARK.

In Figure 5, first row contains dataset column names and remaining are the dataset values. In Figure 6, we have identified behaviour of persons like on which fashion or country they talk most with more LIKES and below screen showing execution time of WITHOUT spark processing. In Figure 7 with SPARK also we got same output, but the difference is execution time and in Figure 8 showing SPARK execution to process same data. Here, graph x-axis represents technique names and y-axis represents execution time and we can see SPARK processing took less execution time so it's faster than traditional processing so BIG DATA processing with SPARK can be efficient.

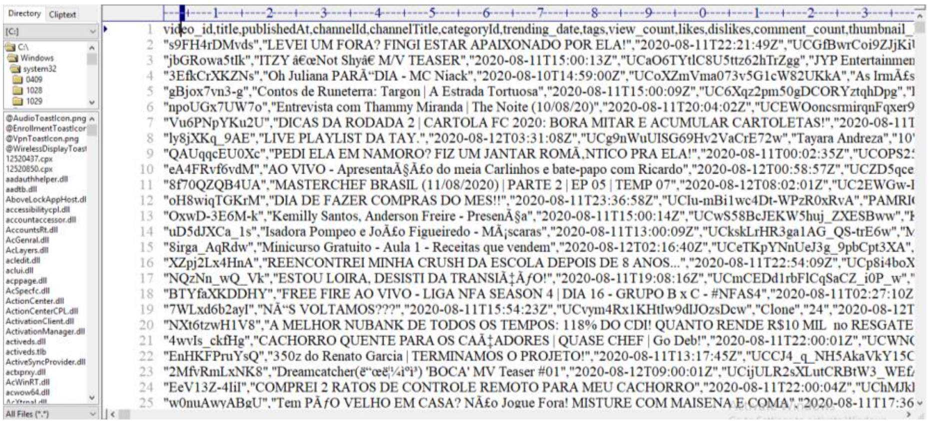

Fig. 5: Sample Dataset.

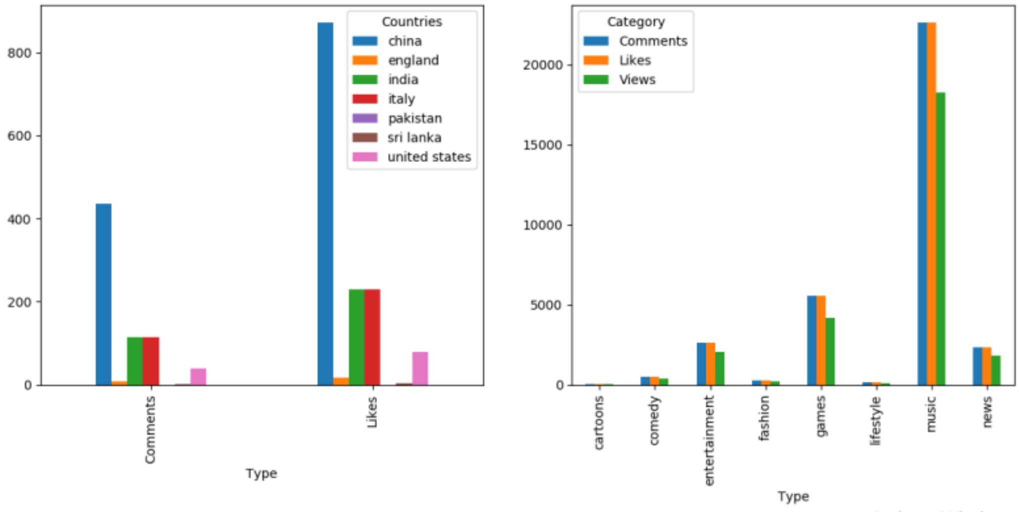

Fig. 6: Individual Behaviour Analysis without SPARK.

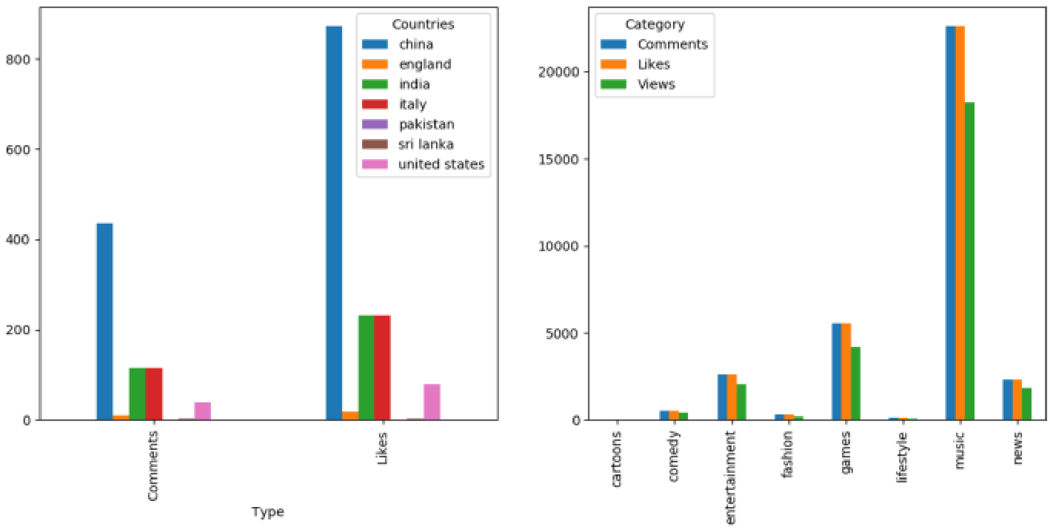

Fig. 7: Individual Behaviour Analysis with SPARK.

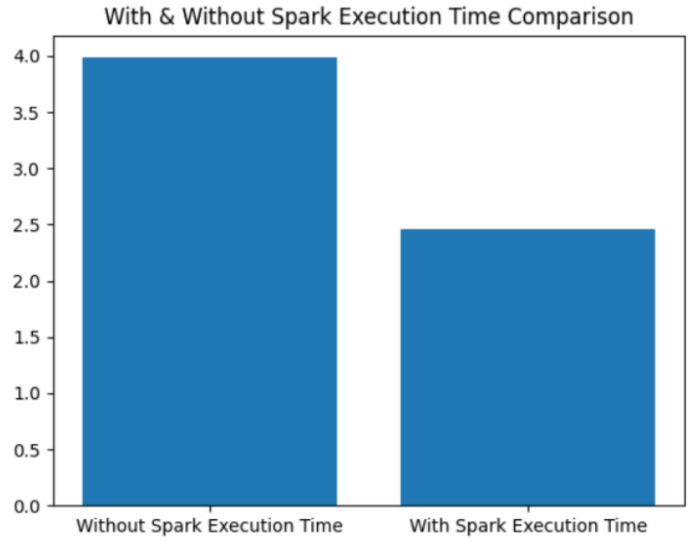

Fig. 8: Execution Time Analysis.

5. CONCLUSION

In conclusion, the proposed architecture leveraging intelligent big data analytics, particularly through the integration of Hive, Spark, and Hadoop, presents a promising solution for efficiently processing vast amounts of social media data. By addressing the challenges associated with analyzing unstructured social media data, such as likes, comments, tweets, and shares, this approach enables companies like Dailymotion to derive valuable insights from their massive user-generated content. The utilization of Spark for parallel processing offers significant advantages in terms of scalability and performance compared to traditional single-threaded algorithms. By distributing tasks across multiple threads, Spark enables faster processing of large datasets, thereby enhancing the speed and efficiency of data analytics tasks. The specific application of this architecture in analyzing Dailymotion's dataset, including identifying popular content categories and extracting relevant metrics like likes, views, and comments, demonstrates its effectiveness in deriving actionable insights from social media data.

REFERENCES

[1] Nolte, K.; Ostermeier, M. Labour Market Effects of Large-Scale Agricultural Investment: Conceptual Considerations and Estimated Employment Effects. World Dev. 2017, 98, 430–446.

[2] Roser, M. Employment in Agriculture. 2023. Available online: https://ourworldindata.org/employment-in-agriculture#article-citation (accessed on 10 November 2023).

[3] Malhi, G.S.; Kaur, M.; Kaushik, P. Impact of Climate Change on Agriculture and Its Mitigation Strategies: A Review. Sustainability 2021, 13, 1318.

[4] Hrustek, L. Sustainability Driven by Agriculture through Digital Transformation. Sustainability 2020, 12, 8596.

[5] DeBoe, G. Economic and environmental sustainability performance of environmental policies in agriculture. In OECD Food, Agriculture and Fisheries Papers; OECD Publishing: Paris, France, 2020; p. 140.

[6] FAO. Employment Indicators 2000–2021. 2023. Available online: https://www.fao.org/3/cc6627en/cc6627en.pdf (accessed on 8 November 2023).

[7] Ji, C.J.; Hu, Y.J.; Tang, B.J.; Qu, S. Price drivers in the carbon emissions trading scheme: Evidence from Chinese emissions trading scheme pilots. J. Clean. Prod. 2021, 278, 123469.
[8] Statistics Canada. Farm Product Price Index (FPPI). 2023. Available online: https://www23.statcan.gc.ca/imdb/p2SV.pl?Function=getSurvey&SDDS=5040 (accessed on 8 November 2023).
[9] Nguyen, P.T.; Nguyen, Q.L.H.T.T. Critical Factors Affecting Construction Price Index: An Integrated Fuzzy Logic and Analytical Hierarchy Process. J. Asian Financ. Econ. Bus. 2020, 7, 197–204.
[10] Lampridi, M.G.; Sørensen, C.G.; Bochtis, D. Agricultural Sustainability: A Review of Concepts and Methods. Sustainability 2019, 11, 5120.
[11] Long, X.; Ji, X. Economic growth quality, environmental sustainability, and social welfare in China-provincial assessment based on genuine progress indicator (GPI). Ecol. Econ. 2019, 159, 157–176.
[12] Coble, K.H.; Mishra, A.K.; Ferrell, S.; Griffin, T. Big data in agriculture: A challenge for the future. Appl. Econ. Perspect. Policy 2018, 40, 79–96.
[13] Misra, N.N.; Dixit, Y.; Al-Mallahi, A.; Bhullar, M.S.; Upadhyay, R.; Martynenko, A. IoT, big data, and artificial intelligence in agriculture and food industry. IEEE Internet Things J. 2020, 9, 6305–6324.
[14] Evstatiev, B.I.; Gabrovska-Evstatieva, K.G. A review on the methods for big data analysis in agriculture. IOP Conf. Ser. Mater. Sci. Eng. 2021, 1032, 012053.
[15] Madhani, P.M. Building a customer-centric supply chain strategy: Enhancing competitive advantages. IUP J. Bus. Strategy 2019, 16, 28–42. [Google Scholar]
[16] Ranjan, J.; Foropon, C. Big data analytics in building the competitive intelligence of organizations. Int. J. Inf. Manag. 2021, 56, 102231.
[17] Sakas, D.P.; Giannakopoulos, N.T.; Terzi, M.C.; Kanellos, N. Engineering Supply Chain Transportation Indexes through Big Data Analytics and Deep Learning. Appl. Sci. 2023, 13, 9983.
[18] Vasumathi, P.; Arun, C.J. Adoption of Digital Marketing in Agribusiness: A Conceptual Study. Acad. Mark. Stud. J. 2021, 25, 1–6. [Google Scholar]
[19] Berbel, J.; Martínez-Dalmau, J. A Simple Agro-Economic Model for Optimal Farm Nitrogen Application under Yield Uncertainty. Agronomy 2021, 11, 1107.
[20] Storm, H.; Baylis, K.; Heckelei, T. Machine learning in agricultural and applied economics. Eur. Rev. Agric. Econ. 2020, 47, 849–892.
[21] Klerkx, L.; Jakku, E.; Labarthe, P. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. NJAS—Wagening. J. Life Sci. 2019, 90–91, 100315.
[22] Lioutas, E.D.; Charatsari, C.; La Rocca, G.; De Rosa, M. Key questions on the use of big data in farming: An activity theory approach. NJAS—Wagening. J. Life Sci. 2019, 90–91, 100297.
[23] Jha, K.; Aalap Doshi, A.; Patel, P.; Shah, M. A comprehensive review on automation in agriculture using artificial intelligence. Artif. Intell. Agric. 2019, 2, 1–12.
[24] Rupnik, R.; Kukar, M.; Vračar, P.; Košir, D.; Pevec, D.; Bosnić, Z. AgroDSS: A decision support system for agriculture and farming. Comput. Electron. Agric. 2019, 161, 260–271.