IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

E-Commerce User Activity Log Analysis Using Hadoop Data Lake

Bharath Guru[1], Chinmay L[2], Darshan A M[3], Channabasava[4], Mrs. Kavitha M. G.[5]
B E, CSE, Kalpataru Institute of Technology, Tiptur, India[1-4]
Assistant Professor, CSE, Kalpataru Institute of Technology, Tiptur, India[5]

Abstract: *The rapid growth of e-commerce platforms has resulted in the generation of massive volumes of user activity data such as clicks, searches, views, and purchases. Analyzing this large-scale data using traditional database systems is inefficient due to limitations in scalability, storage, and processing speed. Big Data technologies provide a promising solution to overcome these challenges. This paper presents an end-to-end system for analyzing e-commerce user activity logs using a Hadoop Data Lake architecture. User interaction data is collected from a web-based e-commerce application, stored in Hadoop Distributed File System (HDFS), and analyzed using Apache Hive. The proposed system enables efficient storage, querying, and analysis of large datasets to extract meaningful insights into user behavior, popular products, and engagement patterns. Experimental results demonstrate that the Hadoop–Hive-based approach is scalable, cost-effective, and suitable for batch analytics in e-commerce environments.*

Keywords: E-commerce, Big Data, Hadoop, HDFS, Hive, Data Lake, User Activity Logs

I. INTRODUCTION

The rapid advancement of information and communication technologies has significantly transformed the way businesses operate, leading to the exponential growth of web-based applications and e-commerce platforms. In today's digital era, organizations increasingly rely on web applications to deliver services, sell products, and interact with customers across the globe. Every interaction between a user and a web application—such as page visits, clicks, searches, logins, and purchases—generates a large amount of data commonly referred to as user activity logs. These logs provide valuable insights into user behavior, preferences, and engagement patterns, which are critical for improving system performance, enhancing user experience, and supporting strategic business decisions.

With the growing popularity of e-commerce platforms, the volume of data generated by users has increased at an unprecedented rate. Millions of users simultaneously access web applications, resulting in high-velocity and high-volume data streams. Traditional data management systems, particularly relational database management systems (RDBMS), are not well-suited to handle such large-scale, continuously growing, and semi-structured data. These systems face challenges related to scalability, storage limitations, and processing overhead, especially when dealing with complex analytical queries over massive datasets. As a result, organizations struggle to extract meaningful insights from user activity data in a timely and cost-effective manner.

Web application analytics has emerged as a crucial domain that focuses on collecting, processing, and analyzing user interaction data to understand application usage patterns. Analytics enables organizations to track user journeys, identify popular features or products, detect anomalies, and evaluate system performance. In e-commerce environments, web analytics plays a vital role in understanding customer behavior, optimizing marketing strategies, improving product recommendations, and increasing conversion rates. However, the effectiveness of analytics largely depends on the underlying data processing architecture and the tools used to manage large-scale data.

To address the limitations of traditional data processing approaches, Big Data technologies have gained widespread adoption in recent years. Big Data is characterized by the three fundamental dimensions known as the "3Vs": Volume, Velocity, and Variety. E-commerce user activity logs perfectly align with these characteristics, as they are generated in massive volumes, at high speed, and in diverse formats. Big Data frameworks such as Apache Hadoop provide a robust

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

and scalable solution for storing and processing such data. Hadoop enables distributed data storage through the Hadoop Distributed File System (HDFS) and supports parallel data processing across clusters of commodity hardware, making it highly suitable for large-scale analytics applications.

Fig.1: Web Application Analytics Overall Working Model

The concept of a Data Lake has further revolutionized data management in Big Data ecosystems. Unlike traditional data warehouses that require data to be structured before storage (schema-on-write), a Data Lake allows raw data to be stored in its native format and processed later using schema-on-read techniques. This flexibility makes Data Lakes ideal for handling unstructured and semi-structured data such as web logs, clickstreams, and event data generated by web applications. By leveraging a Hadoop-based Data Lake, organizations can store vast amounts of raw user activity data and perform analytics as required without losing valuable information.

Apache Hive is an integral component of the Hadoop ecosystem that provides a high-level abstraction for querying and analyzing data stored in HDFS. Hive allows users to perform SQL-like queries, known as Hive Query Language (HiveQL), on large datasets without requiring in-depth knowledge of low-level distributed programming. This makes Hive particularly suitable for batch analytics and reporting tasks in e-commerce applications. By using Hive, analysts and administrators can easily extract insights such as most viewed products, peak usage times, frequently accessed categories, and user engagement trends.

In the context of web application analytics, an end-to-end data pipeline is required to efficiently collect, store, process, and analyze user activity data. Such a pipeline typically begins with data generation at the client side, where user interactions are captured through web interfaces using technologies such as JavaScript. The collected data is then transmitted to a backend server, where it is processed and stored. Integrating this pipeline with a Hadoop-based Data Lake enables scalable storage and efficient analytics, even as the volume of data continues to grow.

This research focuses on the design and implementation of a web application analytics system using a Hadoop Data Lake architecture. The proposed system captures user activity data from a web-based application, stores the data in HDFS, and analyzes it using Apache Hive to generate meaningful analytics reports. The system emphasizes batch-oriented analytics, which is well-suited for historical data analysis and decision support. Data Flow Diagrams (DFDs) are used to model the flow of data within the system, providing a clear understanding of how user interactions are transformed into analytical insights.

II. METHODOLOGY

1. Overview of the Proposed Methodology

The methodology adopted in this work aims to design and implement an efficient big data analytics framework for analyzing e-commerce user activity logs using a Hadoop-based Data Lake architecture. The proposed approach integrates data generation, storage, and analysis into a unified pipeline capable of handling large-scale and semi-structured data generated from user interactions on an e-commerce platform. The methodology combines system analysis, architectural design, and implementation strategies to achieve scalability, reliability, and analytical efficiency.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

2. System Architecture

The proposed system follows a three-layer architectural model consisting of a frontend layer, backend layer, and big data analytics layer. This layered structure ensures modularity and effective separation of responsibilities. The frontend layer simulates an e-commerce website where users interact with various product categories and interface elements. These interactions represent real-world user behavior such as browsing and clicking actions. The frontend is developed using standard web technologies including HTML, CSS, and JavaScript to ensure responsiveness and ease of interaction.

3. Data Collection and Backend Processing

The backend layer is implemented using Flask, a lightweight Python-based web framework responsible for capturing and managing user activity data. JavaScript event handlers embedded within the frontend track user actions and transmit them to the Flask server through HTTP POST requests in JSON format. Each log entry includes essential attributes such as action type, category information, and timestamp. The Flask backend processes incoming requests, validates the data, and appends the records to structured log files, ensuring accurate and consistent data collection.

4. Data Storage Using Hadoop Data Lake

After log generation, the user activity data is stored in JSON format and transferred to the Hadoop Distributed File System (HDFS). HDFS serves as the core storage component of the Data Lake, offering distributed, fault-tolerant, and scalable storage capabilities. Unlike traditional relational databases, HDFS efficiently handles large volumes of unstructured and semi-structured data. Data replication mechanisms in HDFS ensure high reliability and data availability even in the presence of node failures.

5. Data Processing and Analysis Using Hive

For analytical operations, Apache Hive is utilized as the query engine. Hive provides a SQL-like interface known as Hive Query Language (HiveQL), which allows efficient querying of large datasets stored in HDFS. External Hive tables are created to map directly to the stored log files, enabling schema-on-read processing. This flexibility allows the system to adapt to evolving data formats without modifying the underlying storage structure. Hive queries are executed to derive insights such as user activity counts, category-wise engagement, and temporal usage patterns.

6. Data Flow Mechanism

The data flow within the system follows a structured sequence beginning with user interaction at the frontend. User actions generate events that are captured by JavaScript and sent to the Flask backend for processing. The backend stores the events as log files, which are subsequently uploaded to HDFS. Hive accesses these stored logs and performs analytical queries, producing summarized results that support interpretation and decision-making. This seamless data flow ensures effective integration between web-based data generation and big data analytics.

7. Implementation Environment

The system is implemented using open-source tools including Python, Flask, Hadoop, and Hive, deployed in a pseudo-distributed environment using Windows Subsystem for Linux (WSL). This setup is suitable for academic experimentation and proof-of-concept validation. The modular architecture allows easy scalability and future extension to multi-node clusters or cloud-based platforms.

III. LITERATURE REVIEW

The rapid expansion of web applications and e-commerce platforms has led to an exponential increase in the volume of user-generated data. User activity logs, including clicks, page views, searches, and transaction records, have become a critical resource for understanding user behavior and improving system performance. As a result, extensive research has been conducted in the areas of web analytics, big data processing, and scalable data storage architectures. This section

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

reviews relevant literature related to web application analytics, traditional data processing techniques, big data frameworks, and analytics report generation systems.

A. Web Application Analytics

Web application analytics focuses on the collection and analysis of user interaction data to gain insights into application usage patterns. Early studies in web analytics primarily relied on server-side log analysis, where HTTP request logs were examined to understand page visits and access frequencies. These approaches provided basic metrics such as hit counts and session durations but lacked the ability to capture detailed user interactions.

Subsequent research introduced client-side analytics using JavaScript-based event tracking, enabling finer-grained data collection such as button clicks, mouse movements, and navigation paths. Researchers demonstrated that client-side tracking provides a more accurate representation of user behavior compared to traditional server logs. These techniques laid the foundation for modern web analytics systems used in e-commerce and large-scale web platforms.

B. Traditional Data Management Approaches

Traditional relational database management systems (RDBMS) such as MySQL, Oracle, and PostgreSQL have been widely used to store and analyze application data. These systems are optimized for structured data and transactional processing. Several studies reported that while RDBMS solutions are effective for small to medium datasets, they face serious limitations when handling large-scale, high-velocity web logs.

Researchers highlighted challenges such as limited horizontal scalability, high storage costs, and performance degradation when executing complex analytical queries on massive datasets. As web applications grew in scale, it became evident that traditional databases were insufficient for handling the volume and variety of modern web analytics data, leading to the exploration of alternative data processing frameworks.

C. Big Data Technologies for Analytics

The emergence of Big Data technologies marked a significant shift in analytics research. Apache Hadoop introduced a distributed computing model capable of processing large datasets across clusters of commodity hardware. Hadoop Distributed File System (HDFS) enabled fault-tolerant and scalable storage, making it suitable for storing massive web logs and clickstream data.

Several studies demonstrated the effectiveness of Hadoop-based systems in processing large-scale analytics workloads. Researchers emphasized Hadoop's ability to handle the three key characteristics of Big Data—volume, velocity, and variety. However, early Hadoop implementations relied heavily on MapReduce programming, which required complex coding and resulted in higher development effort.

D. Data Lake Architectures

The concept of a Data Lake was introduced to address the limitations of traditional data warehouses. Unlike data warehouses, which require data to be structured before storage, Data Lakes allow raw data to be stored in its native format. This schema-on-read approach provides greater flexibility for analytics, especially for semi-structured and unstructured data such as web logs.

Several authors discussed the advantages of Data Lake architectures in analytics applications, highlighting their ability to store diverse data types and support multiple analytical tools. In the context of web application analytics, Data Lakes enable organizations to preserve raw user activity data and apply different analytical techniques as business requirements evolve. However, researchers also pointed out challenges such as data governance and the risk of creating "data swamps" if proper management practices are not followed.

E. Analytical Query Processing Using Hive

Apache Hive was developed to simplify data analysis on top of Hadoop by providing a SQL-like interface known as Hive Query Language (HiveQL). Hive abstracts the complexity of distributed processing and allows analysts to perform batch queries on large datasets without deep knowledge of parallel programming.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

Multiple studies have reported the successful use of Hive in analyzing web logs, social media data, and e-commerce transactions. Researchers found that Hive significantly reduces development time and improves accessibility for non-programmers. Although Hive is not designed for real-time analytics, it is highly effective for batch-oriented reporting and historical data analysis, which aligns well with web application analytics requirements.

F. Analytics Report Generation Systems

Analytics report generation is a crucial component of decision support systems. Traditional reporting tools relied on periodic extraction of data from databases to generate static reports. With the growth of data volumes, researchers proposed automated report generation systems that integrate data storage, analytics, and reporting within a unified framework.

Studies have shown that separating raw data storage from report repositories improves system performance and maintainability. Batch analytics systems using Big Data frameworks have been successfully applied to generate summary reports, trend analysis, and usage statistics for large-scale applications. These systems enable administrators to make informed decisions based on data-driven insights.

G. Research Gap and Motivation

From the reviewed literature, it is evident that significant progress has been made in web analytics and big data processing. However, many existing approaches focus on isolated aspects such as data collection, storage, or analytics, rather than providing an integrated end-to-end solution. Additionally, some systems are complex to deploy and require substantial computational resources.

There is a need for a simplified yet scalable analytics framework that integrates web-based data collection with Big Data storage and analytical querying. This motivates the proposed work, which combines web application analytics with a centralized analytics data store and batch-oriented report generation. The use of Data Flow Diagrams further enhances clarity in system design and data movement.

IV. RESULTS AND DISCUSSION

The proposed web application analytics system was implemented and evaluated to assess its effectiveness in collecting, processing, and analyzing user activity data and generating analytical reports. This section presents the experimental results obtained from the system and discusses the observed outcomes with respect to performance, scalability, and analytical accuracy.

A. Experimental Setup

The system was tested using a simulated web application environment where multiple users interacted with the application over a period of time. User activities such as page visits, clicks, navigation events, and search operations were generated and captured through the client-side tracking mechanism. The backend server successfully received and processed these interactions, storing the activity logs in the analytics data store.

The analytics module was executed periodically to perform batch analysis on the collected data. Reports were generated based on predefined analytical queries and administrative inputs. The evaluation focused on data integrity, processing efficiency, and the usefulness of the generated reports.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

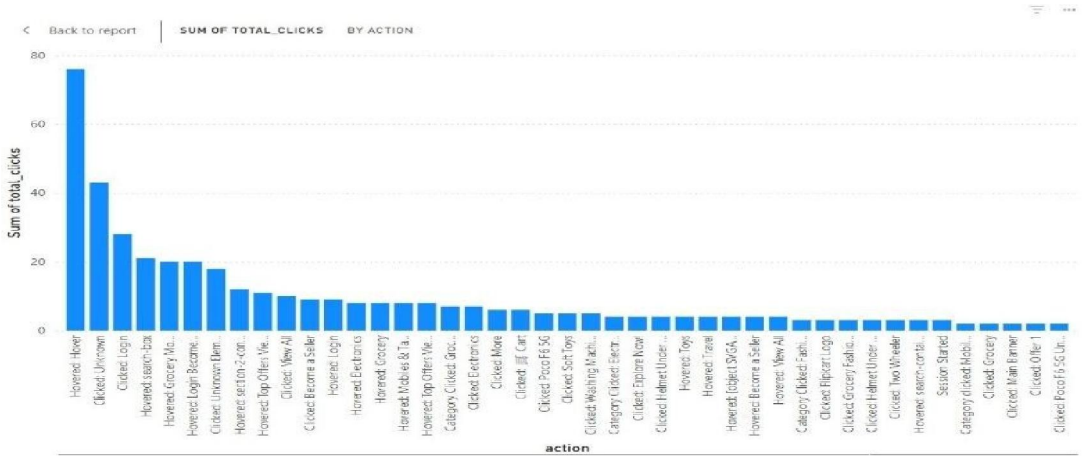

Fig.1: Result Visualization Graphs

B. User Activity Data Collection Results

The system successfully captured all user interactions without data loss. Each user action generated a corresponding log entry containing essential attributes such as activity type and timestamp. The structured format of the logs ensured consistency and simplified downstream processing. The results indicate that client-side tracking combined with backend logging provides reliable and accurate data collection for web analytics applications.

C. Data Processing and Storage Performance

The analytics data store efficiently handled the continuously growing volume of user activity logs. Batch ingestion of log data enabled smooth data storage without affecting the responsiveness of the web application. The separation of raw activity data and processed analytical data improved data organization and facilitated efficient retrieval during analysis.

The system demonstrated good scalability characteristics, as increasing the number of user interactions did not significantly impact processing time. This highlights the suitability of the adopted analytics architecture for large-scale web applications.

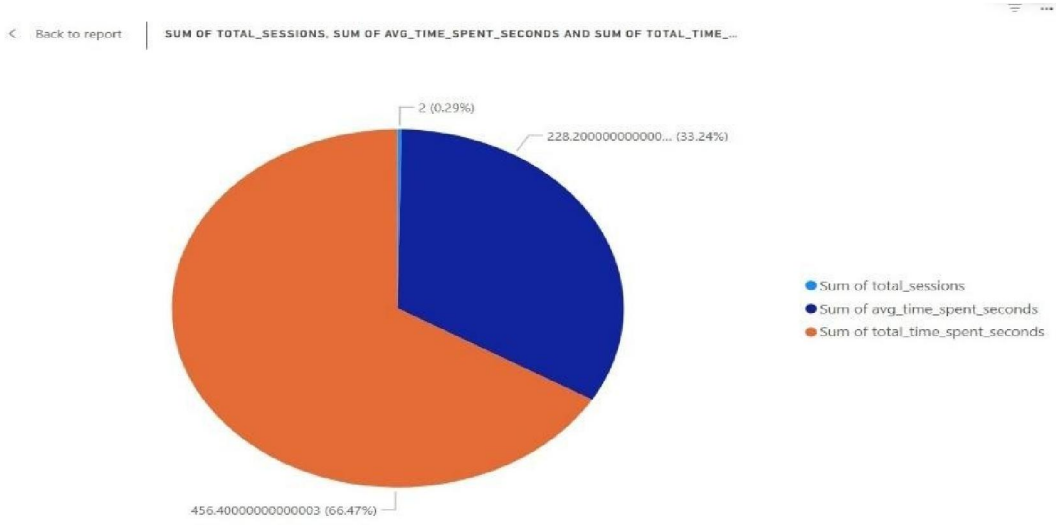

Fig. 2: Result Visualization Graphs

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

D. Analytics and Query Execution Results

Analytical queries were executed on the stored user activity data to derive insights such as most visited pages, frequently performed actions, and peak usage periods. The query results accurately reflected user behavior patterns observed during the experiment. Aggregation and filtering operations produced meaningful summaries that were useful for understanding application usage trends.

The batch-oriented analytics approach proved effective for historical data analysis and trend identification. Although real-time analytics was not implemented, the system efficiently supported periodic reporting and decision-making requirements.

E. Analytics Report Generation

The report generation module successfully transformed analytical results into structured reports. Reports were generated based on administrator-defined parameters and stored in the reports repository for future reference. The generated reports provided clear visibility into user engagement metrics and system usage patterns.

The separation of analytics processing and report storage reduced system complexity and improved maintainability. This design ensures that reports can be accessed independently of raw data processing, enhancing system usability.

F. Post-Analytics Report Evaluation and Discussion

After the analytics reports were generated, an evaluation was conducted to assess the effectiveness of the reporting outputs in conveying meaningful insights. The generated reports consolidated large volumes of user activity data into concise summaries, enabling administrators to quickly understand application usage behavior and interaction patterns. This post-analytics stage played an important role in validating the overall analytics workflow.

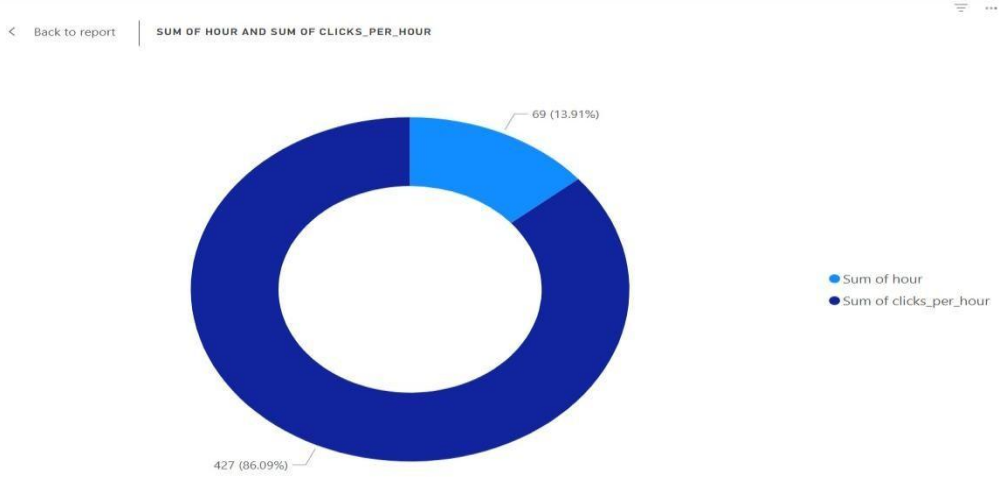

Fig. 3**:** ResultVisualization Graphs

G. Discussion of Results

The results demonstrate that the proposed web application analytics system effectively integrates data collection, processing, and reporting within a unified framework. The use of batch analytics ensures consistency and reliability of results, making the system suitable for decision support and performance analysis.

Compared to traditional analytics approaches, the proposed system offers improved scalability and flexibility by supporting large volumes of semi-structured data. The use of Data Flow Diagrams during system design further contributed to clear data movement and reduced implementation complexity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

However, the system primarily focuses on batch analytics and does not support real-time data processing. While this is sufficient for historical analysis and reporting, future enhancements could include real-time analytics to support immediate decision-making.

V. CONCLUSION

This research has presented a structured and efficient approach for web application analytics and analytics report generation through a well-defined data processing framework. The proposed system successfully captures user interaction data generated from a web application and processes it through a centralized analytics pipeline to derive meaningful insights. By organizing the analytics workflow into distinct stages—data collection, processing, storage, analysis, and reporting—the system ensures clarity, reliability, and scalability in handling large volumes of user activity data.The implementation results demonstrate that the system is capable of accurately recording user actions such as page visits, clicks, and navigation events without data loss. The structured handling of activity logs enables consistent data storage and simplifies subsequent analytical operations. The use of batch-based analytics ensures that historical data can be analyzed effectively, making the system suitable for applications where periodic reporting and trend analysis are required.The analytics report generation process plays a crucial role in transforming raw and processed data into meaningful information for administrators and decision-makers. The generated reports provide clear visibility into user behavior patterns, application usage statistics, and engagement trends. These insights can support informed decisions related to application optimization, performance evaluation, and user experience enhancement. The separation of analytics processing and report storage further improves system maintainability and accessibility.

Another significant outcome of this work is the effective use of Data Flow Diagrams to model the movement of data within the system. The DFDs provide a clear representation of how user interactions are transformed into analytical reports, improving the overall understanding of system design and data dependencies. This structured modeling approach reduces complexity and supports easier system validation and documentation.

Overall, the proposed web application analytics and report generation framework offers a reliable and scalable solution for analyzing user activity data. It addresses the limitations of traditional analytics approaches by supporting large-scale data handling and structured reporting. The system demonstrates the practical applicability of a well-organized analytics pipeline in modern web-based environments and serves as a strong foundation for data-driven analysis in web applications and e-commerce platforms.

REFERENCES

[1] White T, Hadoop The Definitive Guide, Fourth Edition, OReilly Media, Sebastopol, California, USA, 2015.
[2] Dean J and Ghemawat S, MapReduce Simplified Data Processing on Large Clusters, Communications of the ACM, Volume 51, Issue 1, Pages 107 to 113, 2008.
[3] Thusoo A, Sarma J, Jain N, Shao Z, Chakka P, Anthony S, Liu H and Murthy R, Hive A Warehousing Solution Over a MapReduce Framework, Proceedings of the VLDB Endowment, Volume 2, Issue 2, Pages 1626 to 1629, 2009.
[4] Dixon J, Pentaho Hadoop and Data Lakes, Pentaho Corporation Technical Report, 2010.
[5] Zhong Z, Fan J and You J, Big Data Analytics for E Commerce Applications, IEEE Access, Volume 4, Pages 784 to 797, 2016.
[6] Chen M, Mao S and Liu Y, Big Data A Survey, Mobile Networks and Applications, Volume 19, Issue 2, Pages 171 to 209, 2014.
[7] Chaudhuri S, Dayal U and Narasayya V, An Overview of Business Intelligence Technology, Communications of the ACM, Volume 54, Issue 8, Pages 88 to 98, 2011.
[8] Gandomi A and Haider M, Beyond the Hype Big Data Concepts Methods and Analytics, International Journal of Information Management, Volume 35, Issue 2, Pages 137 to 144, 2015.
[9] Kimball R and Ross M, The Data Warehouse Toolkit The Definitive Guide to Dimensional Modeling, Third Edition, Wiley Publishing, USA, 2013.
[10] Zikopoulos P, Eaton C, deRoos D, Deutsch T and Lapis G, Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data, McGraw Hill, New York, USA, 2012.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

[11] Abadi D J, Data Management in the Cloud Limitations and Opportunities, IEEE Data Engineering Bulletin, Volume 32, Issue 1, Pages 3 to 12, 2009.
[12] Hand D J, Mannila H and Smyth P, Principles of Data Mining, MIT Press, Cambridge, Massachusetts, USA, 2001.
[13] Russell S and Norvig P, Artificial Intelligence A Modern Approach, Third Edition, Pearson Education, 2010.
[14] Mankad S, Practical Data Analysis Using Hadoop and Hive, Wiley Publishing, USA, 2013.
[15] Han J, Kamber M and Pei J, Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, USA, 2011.
[16]Alpaydin E, Introduction to Machine Learning, Second Edition, MIT Press, Cambridge, USA, 2010.
[17] Laudon K C and Laudon J P, Management Information Systems Managing the Digital Firm, Twelfth Edition, Pearson Education, USA, 2014.
[18] Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C and Byers A H, Big Data The Next Frontier for Innovation Competition and Productivity, McKinsey Global Institute Report, 2011.
[19] Kune R, Konugurthi P K, Agarwal A, Chillarige R R and Buyya R, The Anatomy of Big Data Computing, Software Practice and Experience, Volume 46, Issue 1, Pages 79 to 105, 2016.
[20] Apache Software Foundation, Apache Hadoop Documentation, https://hadoop.apache.org
[21] Apache Software Foundation, Apache Hive Documentation, https://hive.apache.org
[22] Oracle Corporation, Big Data Handbook, Oracle Press, USA, 2014.