ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

MACHINE LEARNING SOLUTIONS FOR SENTIMENT ANALYSIS ON BIG DATA

Korivi Vamshee Krishna[1] and Dr. Pramod Pandurang Jadhav[2]

[1]Research Scholar and [2]Research Guide, Department of Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore, India

[1]vamshik825@gmail.com and [2]ppjadhav21@gmail.com

ABSTRACT:

This study investigates the integration of machine learning algorithms with big data technologies for enhancing sentiment analysis. The proposed model leverages a Hadoop cluster to enable efficient handling and processing of large datasets, resulting in significant improvements across various performance metrics compared to existing methods. We evaluated several classification algorithms, including SVM, Decision Tree, Naive Bayes, RNN, and Random Forest, and found notable enhancements in accuracy, recall, precision, and F1 score. Specifically, the proposed model achieved accuracy improvements for SVM (95% to 96%), Decision Tree (89% to 93%), Naive Bayes (82% to 95%), RNN (92% to 96%), and Random Forest (91% to 97%). Similarly, recall, precision, and F1 score metrics showed substantial gains. The improvements are attributed to the use of big data quality metrics (BDQM) and parallel data processing, which enhance the reliability and speed of sentiment analysis. Our findings highlight the critical impact of integrating big data technologies with machine learning, addressing traditional sentiment analysis limitations and delivering superior performance. This research underscores the potential of big data frameworks to revolutionize sentiment analysis, providing a robust foundation for future advancements in this field.

Keywords : Hadoop Cluster , Sentiment Analysis , SVM, Decision Tree, Naive Bayes, RNN, Random Forest.

I. INTRODUCTION

In the digital age, the vast amount of data generated daily by individuals and organizations presents both an opportunity and a challenge. One of the most valuable types of data is the textual information shared by users on social media, review sites, forums, and other online platforms. These texts, often rich in sentiments and opinions, offer deep insights into public perception, consumer preferences, and market trends. Analyzing this data effectively can provide a significant competitive advantage to businesses, enabling them to make informed decisions and tailor their strategies accordingly. This process of extracting insights from textual data is known as sentiment analysis. Sentiment analysis, also referred to as opinion mining, involves using natural language processing (NLP), text analysis, and computational linguistics to identify and extract subjective information from text. It is widely used across various domains, including marketing, customer service, finance, and politics, to gauge public sentiment and opinion. However, the exponential growth of social media usage worldwide has led to the generation of massive volumes of data, commonly referred to as Big Data. Traditional sentiment analysis techniques, which were designed for smaller datasets, struggle to cope with the sheer volume, velocity, and variety of Big Data, necessitating the development of more robust and scalable approaches.

Machine learning, a subset of artificial intelligence, has emerged as a powerful tool for sentiment analysis. By learning from large datasets, machine learning algorithms can automatically identify patterns and make predictions, significantly enhancing the accuracy and efficiency of sentiment analysis. The integration of machine learning techniques with Big Data technologies has the potential to revolutionize sentiment analysis. Big Data technologies, such as Hadoop and Spark, provide the necessary infrastructure to store, process, and analyze large datasets in parallel, addressing the limitations of traditional systems. These technologies enable the handling of data at scale, improving the speed and accuracy of sentiment analysis. For instance, Hadoop's distributed storage system allows for the efficient handling of massive datasets, while its MapReduce programming model facilitates parallel processing. Similarly, Apache Spark's in-memory computing capabilities further enhance the speed of data processing. Despite the advantages, several challenges must be addressed to fully harness the potential of Big

Data and machine learning for sentiment analysis. One significant challenge is data quality. The accuracy of sentiment analysis largely depends on the quality of the data being analyzed. Big Data often includes noisy, irrelevant, or redundant information, which can negatively impact the performance of machine learning models. Ensuring high data quality through effective preprocessing, such as filtering, cleaning, and normalization, is crucial for building reliable and credible sentiment analysis systems. Another challenge is the complexity of sentiment analysis in a Big Data context. Unlike traditional datasets, Big Data is characterized by its volume, velocity, variety, and veracity (the four Vs). Handling these characteristics requires sophisticated algorithms and techniques that can scale and adapt to different types of data. For example, real-time sentiment analysis demands algorithms capable of processing streaming data efficiently, while the analysis of diverse data types (e.g., text, images, videos) requires multi-modal approaches. To address these challenges, this study proposes a approach SVM, Decision Tree, Naive Bayes, RNN, Random Forest for sentiment analysis in a Big Data environment, the model aims to improve the accuracy, recall, precision, and F1 score of sentiment analysis. The effectiveness of the proposed model is demonstrated through extensive experiments on large-scale datasets. The results show significant improvements over existing sentiment analysis methods. For instance, the proposed model achieved accuracy improvements across various classification algorithms: SVM (95% to 96%), Decision Tree (89% to 93%), Naive Bayes (82% to 95%), RNN (92% to 96%), and Random Forest (91% to 97%). Similar enhancements were observed in recall, precision, and F1 score metrics. These improvements highlight the critical impact of using Big Data technologies for parallel data processing and classification, resulting in faster computation and higher accuracy. The integration of machine learning and Big Data technologies offers a powerful solution for sentiment analysis, addressing the limitations of traditional methods. The proposed model demonstrates significant improvements in performance metrics, showcasing the potential of machie learning techniques and Big Data infrastructure to revolutionize sentiment analysis.

1.1. Big Data Architecture

Big Data architecture is the structural design that enables organizations to collect, process, store, and analyze large and complex datasets efficiently. This architecture is designed to handle the volume, velocity, variety, and veracity of Big Data, providing a scalable and flexible solution to derive actionable insights. Here, we delve into the core components, architectural models, and challenges of Big Data architecture.

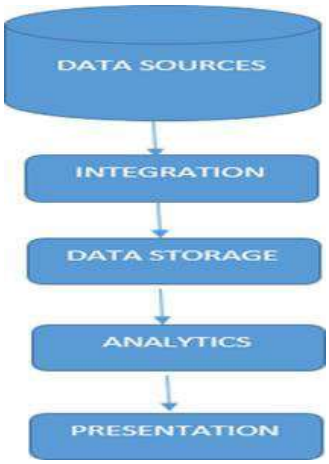


Figure 1: Big Data Architecture.

1.2 Core Components of Big Data Architecture

1. **Data Sources:**

- **Types of Data:** Data can come from various sources and can be structured (e.g., relational databases), semi-structured (e.g., XML, JSON), or unstructured (e.g., text, video, images).
- **Examples:** Social media platforms, IoT devices, transactional databases, logs from web servers, and more.

2. **Data Ingestion:**

- **Batch Ingestion:** Processes large blocks of data at scheduled intervals using tools like Apache Flume, Sqoop, and custom ETL solutions.
- **Real-time Ingestion:** Handles continuous data streams as they arrive using tools like Apache Kafka, Apache NiFi, and AWS Kinesis.

3. **Storage:**

- **Distributed File Systems:** Solutions like Hadoop Distributed File System (HDFS) that store large datasets across multiple nodes.
- **NoSQL Databases:** Databases like Apache Cassandra, HBase, and MongoDB that are designed for high throughput and low latency.
- **Cloud Storage:** Services like Amazon S3, Google Cloud Storage, and Azure Blob Storage that offer scalable and cost-effective storage options.

4. **Processing:**

- **Batch Processing:** Frameworks like Apache Hadoop and Apache Spark process large datasets in parallel across a cluster of machines.
- **Stream Processing:** Tools like Apache Flink, Apache Storm, and Spark Streaming analyze data in real-time, providing immediate insights and actions.

5. **Resource Management:**

- **Cluster Management:** Tools like Apache YARN and Kubernetes manage the allocation of resources across clusters, ensuring optimal performance and utilization.
- **Orchestration:** Kubernetes automates the deployment, scaling, and management of containerized applications.

6. **Analytics and Querying:**

- **SQL on Big Data:** Technologies like Apache Hive, Presto, and Spark SQL enable users to run SQL queries on large datasets.
- **Advanced Analytics:** Tools like Apache Mahout, MLlib (part of Apache Spark), and TensorFlow integrate machine learning algorithms for predictive analytics.

7. **Data Governance and Security:**

- **Data Governance:** Frameworks ensure data integrity, quality, and compliance with regulations using metadata management and data lineage tracking.
- **Security:** Measures include encryption (in transit and at rest), authentication (Kerberos, LDAP), and authorization (Apache Ranger, Sentry).

8. **Integration with AI and Machine Learning:**

- **Machine Learning Models:** Platforms like TensorFlow, PyTorch, and H2O.ai are integrated for building, training, and deploying machine learning models.
- **AI Integration:** Combining AI techniques with Big Data analytics to automate decision-making processes and uncover deeper insights.

1.3 Architectural Models

1. **Batch Processing Architecture:**

- **Description:** Processes large volumes of data in batch mode, typically for historical data analysis.
- **Example Tools:** Apache Hadoop (MapReduce), Apache Spark (batch mode).
- **Use Cases:** Data warehousing, ETL processes, periodic reporting.

2. **Real-time Processing Architecture:**

- **Description:** Handles data in real-time as it arrives, providing immediate processing and insights.
- **Example Tools:** Apache Kafka Streams, Apache Flink, Apache Storm.
- **Use Cases:** Real-time monitoring, fraud detection, real-time recommendations.

3. **Lambda Architecture:**

- **Description:** Combines both batch and real-time processing to offer a comprehensive view of data.
- **Layers:**
- **Batch Layer:** Manages historical data processing using batch frameworks.
- **Speed Layer:** Handles real-time data processing.
- **Serving Layer:** Merges results from both batch and speed layers for querying.
- **Use Cases:** Complex event processing, real-time analytics with historical context.

4. **Kappa Architecture:**

- **Description:** A simplified version of Lambda architecture that focuses solely on real-time processing.
- **Key Components:** Uses a streaming platform (like Kafka) and stream processing tools (like Flink or Spark Streaming).
- **Use Cases:** Applications where real-time processing is sufficient, and batch processing is unnecessary.

1.4 Challenges in Big Data Architecture

1. **Scalability:**

- **Issue:** Ensuring the architecture can scale horizontally to accommodate growing data volumes.
- **Solution:** Using distributed systems and cloud-based solutions that allow seamless scaling.

2. **Complexity:**

- **Issue:** Managing the complexity of integrating various tools and technologies.
- **Solution:** Adopting standardized frameworks and platforms that provide end-to-end solutions.

3. **Data Quality:**

- **Issue:** Ensuring data accuracy, completeness, and consistency.
- **Solution:** Implementing data validation, cleaning, and enrichment processes.

4. **Security and Privacy:**

- **Issue:** Protecting sensitive data from unauthorized access and ensuring compliance with regulations.

- **Solution:** Employing robust encryption, access control mechanisms, and regular security audits.

5. **Interoperability:**

- **Issue:** Integrating different data sources and technologies seamlessly.
- **Solution:** Using open standards and APIs for data exchange and integration.

6. **Cost Management:**

- **Issue:** Managing the costs associated with storage, processing, and data transfer.
- **Solution:** Optimizing resource usage, leveraging cost-effective cloud solutions, and monitoring expenditures closely.

1.5 Machine Learning

Machine learning involves developing algorithms that learn from data, enabling the construction of models that make predictions based on this learned information instead of relying on explicitly programmed instructions. The primary types of machine learning are supervised learning, unsupervised learning, and semi-supervised learning.

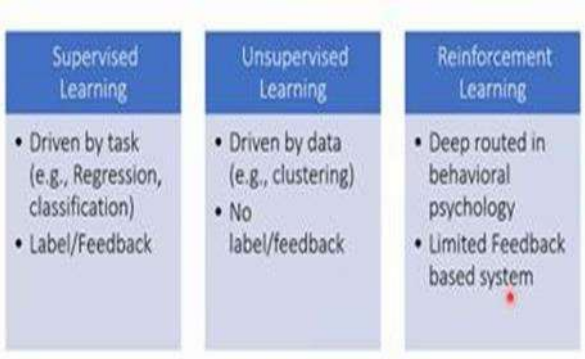


Figure 2: Types of Machine Learning approach

Machine Learning is a subset of artificial intelligence (AI) that involves the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, these systems learn and improve from experience by identifying patterns and making data-driven decisions. Machine learning algorithms build a mathematical model based on sample data, known as training data, to make predictions or decisions without being explicitly programmed to perform the task. This approach allows for the automation of analytical model building and empowers systems to adapt and improve over time with minimal human intervention.

There are Several Types of Machine Learning, Each Serving Different Purposes and Suited for Various Tasks:

1. **Supervised Learning:** Supervised learning involves training a machine learning model on a labeled dataset, which means that each training example is paired with an output label. The model learns to map inputs to the correct output based on this training data. Common algorithms used in supervised learning include linear regression, logistic regression, support vector machines, and neural networks. This type of learning is typically used for tasks like classification (e.g., spam detection) and regression (e.g., predicting house prices).
2. **Unsupervised Learning:** In unsupervised learning, the model is trained on data that does not have labeled responses. The system tries to learn the underlying structure of the data by identifying patterns, relationships, or clusters within the dataset. Common techniques include clustering (e.g., k-means clustering) and dimensionality reduction (e.g., principal component analysis). Unsupervised learning is often used for exploratory data analysis, market basket analysis, and anomaly detection.

3. **Semi-supervised Learning:** Semi-supervised learning is a middle ground between supervised and unsupervised learning. It uses a small amount of labeled data along with a large amount of unlabeled data during training. This approach can significantly improve learning accuracy when obtaining a fully labeled dataset is challenging or expensive. Techniques from both supervised and unsupervised learning are combined to leverage the available labeled data to guide the learning process on the unlabeled data.
4. **Reinforcement Learning:** Reinforcement learning is a type of machine learning where an agent learns to make decisions by performing certain actions and receiving feedback from the environment in the form of rewards or penalties. The goal is to maximize the cumulative reward over time. This approach is widely used in robotics, game playing (e.g., AlphaGo), and autonomous systems. Key concepts in reinforcement learning include the agent, environment, actions, states, and rewards.
5. **Self-supervised Learning:** Self-supervised learning is a relatively new approach where the system generates its own labels from the input data. This technique is particularly useful in situations where labeled data is scarce but large amounts of unlabeled data are available. The model creates pseudo-labels by predicting parts of the data from other parts, facilitating the learning process. This method has been successful in fields like natural language processing (NLP) and computer vision.
6. **Transfer Learning:** Transfer learning involves taking a pre-trained model on a large dataset and fine-tuning it for a specific, often smaller, task. This approach leverages the knowledge gained from the pre-trained model and applies it to a new, related problem, significantly reducing the time and resources needed for training. Transfer learning is commonly used in deep learning applications, such as image recognition and natural language processing.

II. LITERATURE SURVEY

Sohangir et al., (2018) , Deep Learning and Big Data analytics are pivotal in the realm of data science. In recent years, Deep Learning models have made significant strides in domains like speech recognition and computer vision. Big Data holds utmost importance for organizations requiring massive data collection, such as social networks, and Deep Learning emerges as a valuable tool for analyzing this extensive data. This symbiotic relationship enables Deep Learning to extract profound insights buried within Big Data, exemplified by the modern stock market's dynamics. Despite being a popular avenue for wealth accumulation, challenges persist in timing stock trades or selecting appropriate stocks for investment. While professional financial advisors have traditionally provided guidance, the advent of financial social networks like StockTwits and SeekingAlpha has democratized financial insights. In this paper, we explore the adaptation of Deep Learning models to enhance sentiment analysis for StockTwits. Employing various neural network models such as long short-term memory (LSTM), doc2vec, and convolutional neural networks (CNNs), we demonstrate the effectiveness of Deep Learning for financial sentiment analysis, with CNNs emerging as the optimal model for predicting sentiments within the StockTwits dataset [1].

Sharef et al., (2016), Public sentiment analysis in social media is increasingly recognized as a strategic tool for market understanding, customer segmentation, and stock price prediction. This evolution is fueled by the growth in Big Data frameworks, making Sentiment Analysis (SA) applications commonplace in businesses. However, gaps in SA application within Big Data contexts remain underexplored. This study addresses this gap by reviewing state-of-the-art SA approaches and assessing their suitability for Big Data frameworks. By highlighting existing gaps and suggesting future research directions, this paper aims to expand SA approaches to maximize value mining for users [2].

Seman & Razmi, (2020), Social media platforms generate an enormous volume of data, particularly in the form of unstructured tweets. Sentiment analysis of these tweets can offer valuable insights for organizations seeking to enhance products and increase profitability. This research compares the efficacy of three machine learning algorithms—Support Vector Machine (SVM), Decision Trees (DT), and Naive Bayes (NB)—for classifying Twitter sentiments. The experimental results demonstrate SVM's superiority in precision and overall performance

across different datasets. However, further exploration of preprocessing techniques is essential to enhance classifier results [3].

Calderón et al., (2019), Real-time analysis of tweets through supervised sentiment analysis presents an opportunity for communication and audience research. By integrating machine learning and streaming analytics in a distributed environment, scholars can classify messages instantly, enabling cross-sectional, longitudinal, and experimental designs. This paper illustrates the implementation of parallelized machine learning methods in Apache Spark to predict sentiments in real-time tweets, discussing implications and limitations for communication, audience, and media studies [4].

Biradar et al., (2022), Micro-blogging on social media platforms is ubiquitous worldwide, generating vast amounts of structured and unstructured Big Data. This research focuses on leveraging Apache Hadoop and Apache Hive for social media analytics to inform business decisions. Through sentiment analysis, the study extracts valuable insights from real-time social media data, showcasing the efficiency of unsupervised clustering and supervised machine learning techniques. The developed tool demonstrates significant speed improvements over traditional databases and achieves high accuracy, facilitating the interpretation of interactions and associations among individuals, topics, and ideas [5].

Ragini et al., (2018), Big data generated from social media and mobile networks offer an unparalleled opportunity to glean valuable insights. While businesses commonly utilize this information to gauge customer satisfaction, its potential in disaster response remains largely untapped. Social networks are increasingly utilized for emergency communications and aid requests during disasters. Amidst such crises, it's crucial to mine these emergency requests from the vast pool of big data to provide timely assistance. The sentiment of affected individuals during and after a disaster significantly influences the success of response and recovery efforts. In this paper, we propose a big data-driven approach for disaster response through sentiment analysis. Our model collects disaster data from social networks and categorizes them based on the needs of affected individuals. These categorized data are then subjected to machine learning algorithms to analyze people's sentiments. By evaluating various features such as parts of speech and lexicon, we identify the most effective classification strategy for disaster data. Our findings suggest that a lexicon-based approach is suitable for analyzing the needs of individuals during disasters. The practical implication of our proposed methodology lies in real-time categorization and classification of social media big data for disaster response and recovery, empowering emergency responders and rescue personnel to develop more effective strategies amidst rapidly evolving disaster scenarios [6].

Hajiali, M., (2020), Sentiment analysis, capable of extracting information from various text sources and classifying them based on polarity, is increasingly applied to big data generated through mobile networks and social media. This application enables businesses to derive actionable commercial insights from text-oriented content. However, comprehensive investigations in this context remain scarce. This paper aims to fill this gap by conducting a systematic literature review to explore state-of-the-art techniques and identify future research directions. Through meticulous selection criteria, we categorized big data and sentiment analysis into centralized and distributed platforms, examining the advantages and disadvantages of each technique. Our study underscores the importance of efficient textual big data analysis in enhancing efficiency, flexibility, and intelligence. By providing comparative insights and analyzing current developments, this paper serves as a valuable resource for academics and professionals seeking to harness sentiment analysis and big data for public opinion estimation and prediction [7].

Kurian et al., (2015), Social media platforms serve as effective channels for communication and brand perception. Analyzing user-generated content on these platforms provides valuable insights into consumer perceptions, enabling businesses to make informed decisions. With the proliferation of sentiment-rich social media content, sentiment analysis has become a significant task in big data analytics. This study focuses on evaluating sentiment analysis techniques' performance on a large dataset of tweets using Hadoop. Our

experimental results demonstrate the technique's efficiency in handling significant sentiment datasets, contributing to improved sentiment analysis outcomes [8].

Chaturvedi et al., (2017), Sentiment analysis offers an efficient method for examining textual data from various internet sources. By leveraging machine learning techniques, sentiment analysis facilitates the extraction and analysis of user opinions, reviews, and feedback, aiding decision-making processes in various domains. This paper highlights the increasing importance of sentiment analysis, particularly in the context of big data. By reviewing current research approaches and identifying research gaps, this study aims to enhance understanding and exploration of sentiment analysis in the realm of business intelligence [9].

Agarwal et al., eds., (2020), This book delves into deep learning-based approaches for sentiment analysis, a rapidly evolving research area. By presenting state-of-the-art solutions to common challenges in sentiment analysis, the book serves as a valuable resource for researchers and practitioners alike [10].

El Alaoui et al., (2019), We live in a world where digital interactions generate vast amounts of data, constituting a significant portion of big data. Sentiment analysis, a burgeoning field, heavily relies on big data for valuable insights. However, existing applications often overlook key big data characteristics. This paper focuses on aligning sentiment analysis applications with big data contexts, emphasizing the importance of considering all big data characteristics for optimal analysis [11].

Alarifi et al., (2020), Sentiment analysis is pivotal in various systems, such as opinion mining and prediction. However, high error rates in sentiment analysis studies can hinder overall system efficiency. This paper introduces a novel big data and machine learning technique to evaluate sentiment analysis processes. By employing preprocessing data mining concepts and an optimal classifier, our approach significantly improves system efficiency, as evidenced by experimental results [12].

Behera et al., (2021), Analysis of consumer reviews on social media is crucial for various business applications, given the exponential growth in both the volume and relevance of such reviews, leading to big data. This paper proposes a hybrid approach utilizing Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures for sentiment classification of reviews across diverse domains. CNNs excel in local feature selection, while LSTMs are effective in sequential analysis of long texts. The Co-LSTM model proposed in this study aims to address two key objectives in sentiment analysis: scalability for examining big social data and domain independence. Experimental results on four diverse review datasets demonstrate the superiority of the proposed ensemble model over other machine learning approaches in terms of accuracy and other performance metrics [13].

Jain et al., (2022), Recent advancements in networking and information technology have spurred the generation of vast amounts of data, leading to the rise of Big Data Analytics (BDA). Cognitive computing, an AI-based system, offers solutions to challenges encountered in BDA. Sentiment Analysis (SA) is instrumental in understanding linguistic-based tweets, extracting features, and analyzing sentiment. Applying SA to big data enables businesses to derive commercial insights from text-oriented content. This paper introduces a new cognitive computing approach integrated with big data analysis tools for SA. The proposed model involves preprocessing, feature extraction, feature selection, and classification using Hadoop MapReduce. Experimental results demonstrate the superior performance of the proposed BBSO-FCM model in sentiment analysis, highlighting its improved classification performance on benchmark datasets [14].

Kılınç, D., (2019), Various data sources produce large volumes of data, necessitating new distributed processing approaches for extracting valuable information. Real-time sentiment analysis, a demanding research area, requires powerful Big Data analytics tools like Spark. This paper addresses the challenge of ensuring the authenticity of generated sentiment data by integrating a fake account detection service into the sentiment analysis framework. The developed system, comprising machine learning and streaming services, Twitter streaming, fake account

detection, and real-time reporting components, achieves high sentiment classification performance in both offline and real-time modes [15].

Jan et al., (2019), Deep learning methods have revolutionized various fields, including speech recognition and image classification, offering enhanced capabilities in processing large volumes of data. Big Data analytics requires sophisticated algorithms based on machine and deep learning techniques for real-time processing with high accuracy and efficiency. However, existing techniques often lack generalizability and efficiency in processing generic big data scenarios. This article presents a comparative study of various deep learning techniques for processing large volumes of data with different architectures. The study highlights the effectiveness of deep learning techniques when combined with supervised and unsupervised training methods [16].

El Alaoui & Gahi, (2019), Human beings express their opinions about various subjects, products, and services through internet and social networks, presenting an opportunity for organizations to gain valuable insights for decision-making. To effectively analyze this information, organizations employ Sentiment Analysis (SA) techniques. However, the exponential growth of social media usage worldwide has led to a massive generation of data, surpassing the capabilities of traditional SA systems. Combining SA techniques with Big Data technologies offers a solution to efficiently analyze such voluminous data. Nevertheless, challenges associated with Big Data, such as data quality, can significantly impact the accuracy of SA systems. Therefore, our research focuses on integrating Big Data Quality Metrics (BDQM) into SA processes. We identify key BDQM throughout the Big Data Value Chain (BDVC) and assess their impact on SA accuracy through a real case study, providing simulation results [17].

Qiu et al., (2016), Big data is expanding rapidly across all science and engineering domains, presenting significant potential but also various challenges. This paper conducts a literature survey on the latest advances in machine learning for big data processing. It reviews machine learning techniques, highlighting promising methods like representation learning, deep learning, and distributed learning. Challenges and potential solutions in machine learning for big data are discussed, along with the integration of machine learning with signal processing techniques. Finally, the paper outlines open issues and research trends in this field [18].

Rahnama, A.H.A., (2014, November) , The rise of big data has necessitated continuous processing of fast data streams, leading to the emergence of real-time analytics on stream data. However, processing real-time stream data poses challenges due to the impossibility of storing all data instances. This paper introduces Sentinel, a distributed system designed for real-time stream data processing. Sentinel utilizes online analytical algorithms and parallel decision tree-learning algorithms to provide real-time analytics on stream data. The system, built on Apache Storm, effectively addresses the challenges of real-time stream data processing and demonstrates promising results when applied to Twitter Public Stream API [19].

L'heureux et al., (2017), The Big Data revolution holds the promise of transforming various aspects of life through data analytics, with machine learning playing a central role. However, traditional machine learning approaches face challenges in the era of big data due to broken assumptions and the unique characteristics of big data. This paper organizes machine learning challenges with big data according to key dimensions such as volume, velocity, variety, and veracity. It discusses emerging machine learning approaches capable of handling these challenges and provides a matrix relating challenges to approaches, offering insights for practitioners and researchers in the field [20].

Challagundla et al., (2023), Sentiment Analysis (SA) and Opinion Mining have become crucial in handling the vast amount of sentiment-rich social media data on the web. This study presents an effective method for SA on large-scale tweet datasets using Machine Learning algorithms within the Hadoop ecosystem. The experimental results demonstrate the exceptional efficiency of our approach, with Support Vector Machines (SVM) outperforming other classifiers with an accuracy of 95% and a ROC of 93%. This work significantly contributes

to SA in the Big Data realm, offering a flexible, scalable method for evaluating sentiment-rich social media content and providing valuable insights for businesses, governments, and individuals [21].

III.PROPOSED METHODOLOGY

The proposed sentiment analysis model is an advanced emotion detection framework that integrates three key components: data collection, data processing, and data classification, as illustrated in Figure 3. The model utilizes MapReduce for computational power and the Hadoop Distributed File System (HDFS) for data storage, though it supports other storage systems as well. In this context, a "split" typically refers to an HDFS block, which can be customized in size. Each map job (an instance of the mapper) processes one split. This study employs the Kaggle dataset as a comprehensive source of raw data across various categories, which is then cleaned using Apache Flume. The data processing pipeline includes removing stop words and performing lemmatization.

Step 1: Data Collection

- **Apache Flume:** Apache Flume is a reliable and highly available distributed system designed to efficiently gather, combine, and transport large volumes of log data. Deployed as a Twitter Agent, it ingests data from the Twitter Service (Twitter Stream API) or Kaggle and transmits it to HDFS via the Memory Channel .
- **Memory Channel:** The Memory Channel acts as an intermediary facilitating the transmission of events between the Twitter Source and the HDFS Sink. Events are initially added to the channel by the Twitter Source and subsequently removed by the HDFS Sink.
- **HDFS Sink:** Within the Apache Flume ecosystem, the HDFS Sink functions as the endpoint for data transmission. Its primary purpose is to transfer data from the Flume channel into the Hadoop Distributed File System (HDFS).
- **Hadoop Distributed File System (HDFS) Sink:** The HDFS Sink is a crucial component of the Apache Flume data import system, enabling the transfer of data from various sources to HDFS. It is extensively used in big data environments to efficiently collect, process, and store large datasets within Hadoop's distributed file system.

Step 2: Data Cleaning

Preprocessing the data involves extracting relevant fields from the Twitter data, with a primary focus on the tweet content. This phase of the proposed work is dedicated to preprocessing data related to different crypto investors. Given the informal nature of the data, which may contain misspellings and non-textual information, the following steps are necessary:

- **Tokenization:** This process assigns tokens to lexical features such as verbs, adjectives, and nouns. It also identifies other text characteristics like phone numbers, URLs, emoticons, and various hashtags.
- **Conversion:** In this phase, repeated words are discarded, and all words are converted to lowercase to ensure uniformity in the data.
- **Stemming:** Using morphological stemming, features like conjugations, genders, and plurals are removed in this phase.
- **Filtering:** Various filters are applied to classify indicators representing multiple emotions with respect to single words.
- **Prediction:** An adaptive intelligent prediction technique is proposed for this work. Due to the volume, variety, velocity, and dynamic nature of the data, Hadoop will be used to process the entire framework efficiently.

Step 3: Data classification

The processed data will be classified using supervised machine learning algorithms to estimate user perceptions. At this stage, the text data will be categorized into different classes based on the annotated sentiment-representing words. Various machine learning classifiers, including SVM, Decision Trees, Naive Bayes, RNN, and Random Forests, will be employed for this classification.

Step 4: Result Analysis

Finally, the classification results will be evaluated using metrics such as accuracy, recall, and precision. The proposed model, detailed in Figure 4, provides a comprehensive overview of these results.

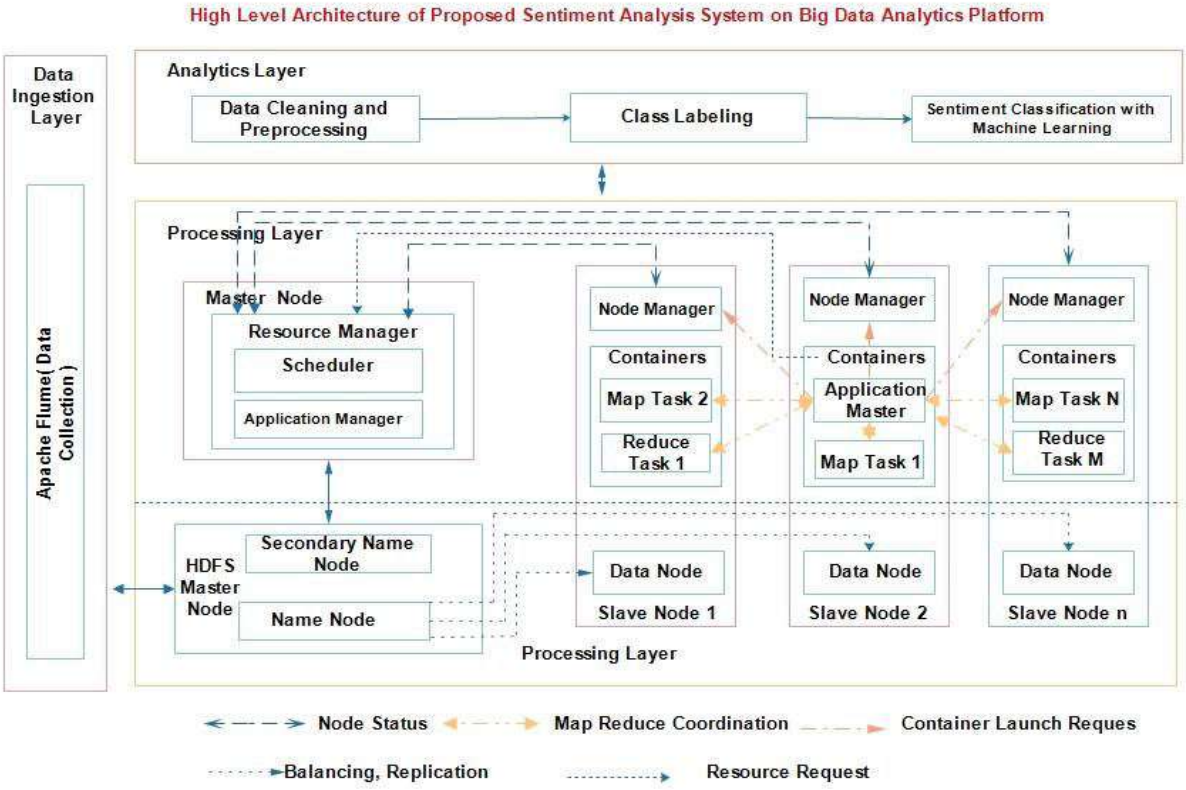


Figure 4: Proposed architecture with respect to big data

The high-level architecture figure 4 of the proposed sentiment analysis system on a Big Data analytics platform consists of multiple layers: data ingestion, analytics, and processing. Data ingestion is handled by Apache Flume, which collects and channels data into the system. The analytics layer includes data cleaning and preprocessing, class labeling, and sentiment classification using machine learning techniques. The processing layer, managed by a Resource Manager and Application Manager, coordinates tasks across various nodes. This layer employs a distributed processing framework where tasks are divided among multiple nodes, each managed by a Node Manager. The nodes perform map and reduce tasks, facilitated by containers, to process data in parallel. The HDFS (Hadoop Distributed File System) master node oversees data storage, supported by secondary name nodes and data nodes, ensuring data replication, balancing, and efficient resource utilization. This architecture ensures efficient handling of large-scale data for sentiment analysis, leveraging the scalability and distributed processing capabilities of Big Data technologies.

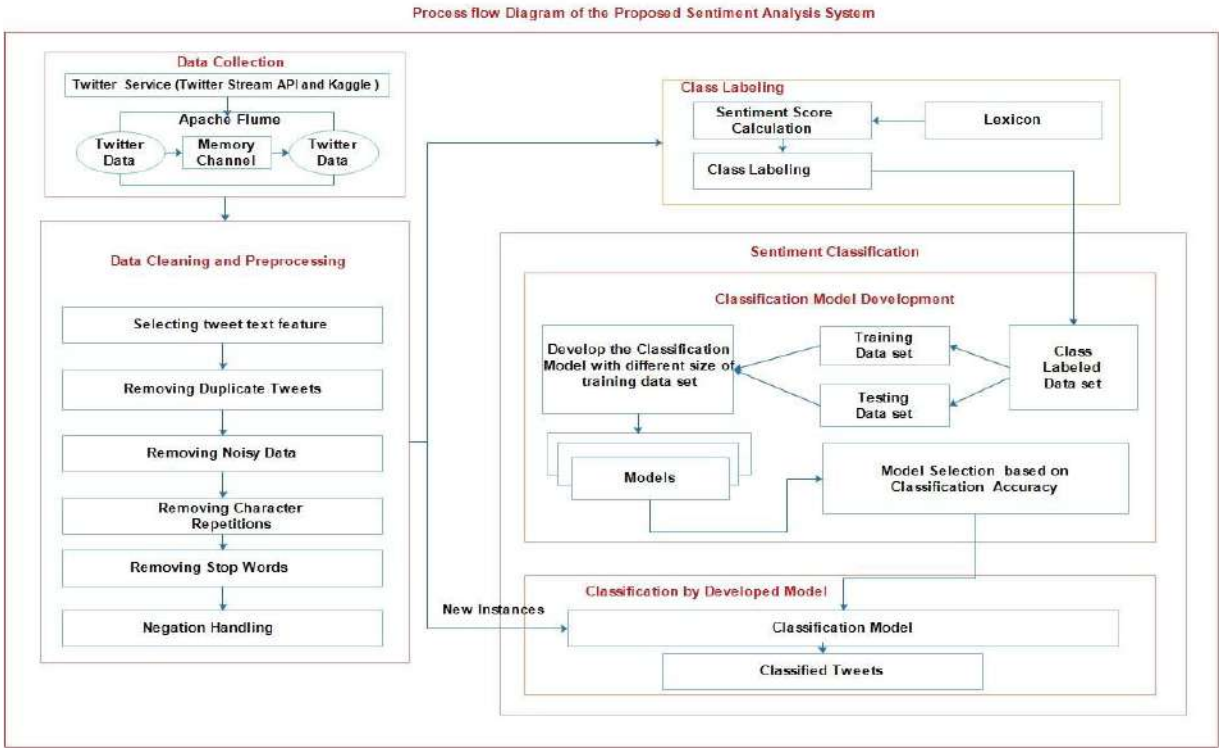


Figure 5: Proposed Architecture with Dataset

The process flow diagram figure 5 of the proposed sentiment analysis system comprises several stages: data collection, data cleaning and preprocessing, class labeling, and sentiment classification. Data is gathered from Twitter (via the Twitter Stream API and Kaggle) using Apache Flume, which utilizes the Memory Channel to transfer data. The cleaning and preprocessing phase involves selecting relevant tweet text features, removing duplicate tweets, eliminating noisy data and character repetitions, filtering stop words, and handling negations. In the class labeling stage, sentiment scores are calculated using a lexicon, and tweets are labeled accordingly. The sentiment classification phase includes developing classification models with varying datasets, training and testing these models, and selecting the best model based on classification accuracy. Finally, new instances are classified using the developed model, resulting in categorized tweets.

SVM (Support Vector Machine)

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates the data points of different classes in the feature space. SVM is particularly effective in high-dimensional spaces and is known for its robustness in handling both linear and non-linear data through the use of kernel functions. The primary objective of SVM is to maximize the margin between the data points of different classes, thereby enhancing the model's generalization capabilities. In sentiment analysis, SVM is utilized to classify text data into predefined sentiment categories by transforming the text into a high-dimensional feature space.

Decision Tree

Decision Tree is a popular supervised learning algorithm that is used for both classification and regression tasks. It operates by recursively splitting the data into subsets based on the value of input features, forming a tree-like structure. Each node in the tree represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. Decision Trees are easy to interpret and understand, making them valuable for identifying the most significant features in the data. In sentiment analysis, Decision Trees can be used to classify text by learning decision rules inferred from the training data, thereby determining the sentiment expressed in the text.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' Theorem, with the assumption of independence between features. Despite its simplicity, Naive Bayes is highly effective for text classification tasks, including sentiment analysis. It calculates the posterior probability of each class given the input features and assigns the class with the highest probability. The algorithm is called "naive" because it assumes that all features contribute independently to the probability of a given class, which is rarely the case in real-world data. However, this assumption simplifies the computation and often yields surprisingly accurate results in practice. Naive Bayes is particularly useful for large datasets and real-time predictions due to its efficiency.

RNN (Recurrent Neural Network)

Recurrent Neural Networks (RNNs) are a class of neural networks designed to recognize patterns in sequences of data, such as text, time series, and speech. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing them to maintain a hidden state that captures information about previous inputs. This makes RNNs particularly well-suited for tasks that involve sequential data, such as sentiment analysis. Long Short-Term Memory (LSTM) networks, a type of RNN, are commonly used to address the vanishing gradient problem and capture long-term dependencies in the data. In sentiment analysis, RNNs process the text sequence to learn and predict the sentiment based on the context provided by previous words in the sequence.

Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It combines the concept of "bagging" (bootstrap aggregating) and random feature selection to create a robust model that reduces overfitting and improves generalization. Each tree in the forest is built on a random subset of the data and a random subset of features, which increases diversity among the trees and leads to better performance. In sentiment analysis, Random Forest can handle a large number of input features and complex interactions between them, making it a powerful tool for text classification tasks.

IV. IMPLEMENTATION

A. Hardware and Software

The device specifications include a DESKTOP-RUU7KMF with an Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz, 8.00 GB (7.89 GB usable) of RAM, a 64-bit operating system, and an x64-based processor. The Python packages used are TensorFlow, Scikit-Learn, NumPy, Matplotlib, Jupyter Notebook, PyTorch, LightGBM, TextBlob, SciPy, Pydoop, and Pandas.

B. Dataset

Initially, we utilized a dataset related to cryptocurrency investors, sourced from Kaggle (https://www.kaggle.com/datasets/georgemac510/top-100-crypto-dataset). This dataset, which is 18GB in size, comprises tweets concerning Bitcoin and investors.

Algorithm-1: Eliminate Duplicate Tweets from a Dataset

1. **Collecting the Data**: Obtain a dataset from Twitter containing the tweets you wish to analyze.
2. **Data Preprocessing**: Extract relevant fields from the Twitter data, focusing primarily on tweet content. Remove any non-essential information or metadata to streamline the process.
3. **Develop a Hash Function**: Create a hash function to generate a unique identifier for each tweet based on its content. Popular hashing algorithms include MD5 and SHA-1.
4. **Initialize an Empty Set**: Create an empty set to keep track of individual tweet hashes.
5. **Calculate the Tweet's Hash**: Apply the hash function to the text of each tweet to generate its hash value.

6. **Check for Duplication**: Before adding the tweet hash to the set, check if it already exists in the set. If it does not, add the hash to the set.
7. **Remove Duplicate Tweets**: If the hash is unique, add it to the set. If the hash is already present, indicating a duplicate tweet, skip it and move to the next tweet.
8. **Construct the New Dataset**: Create a new dataset that includes only unique tweets as distinguished by their hashes.

Algorithm-2: Build the New Dataset Using Tf-Idf (Term Frequency Inverse Document Frequency)

1. **Data Collection**: Obtain the Twitter dataset containing tweets you want to process.
2. **Data Preprocessing**: Extract the tweet text from the Twitter data and remove any unnecessary information or metadata.
3. **Calculate Term Frequency (TF) for Each Tweet**:
- Initialize an empty dictionary "tf_dict" to store the term frequencies of words in each tweet.
- For each tweet in the dataset, tokenize the tweet text into individual words, count the occurrences of each word, and store them in "tf_dict" with the tweet ID as the key.
4. **Calculate Document Frequency (DF) for Each Word**:
- Initialize an empty dictionary "df_dict" to store the document frequencies of words in the entire dataset.
- For each tweet, tokenize the tweet text into individual words and create a set "unique_words" to save the unique terms in the tweet. Increment the document frequency of each word in "unique_words" by 1 in "df_dict".
5. **Determine Inverse Document Frequency (IDF) for Each Word**:
- Initialize a dictionary "idf_dict" to store the inverse document frequency of words in the entire dataset.
- For each word in "df_dict", calculate IDF using the equation: IDF(word) = log((Total number of tweets) / (DF(word) + 1)).
6. **Find the TF-IDF Score for Each Word in a Tweet**:
- For each tweet, tokenize the tweet text into individual words and initialize an empty dictionary "tfidf_dict" to store the TF-IDF scores of words in the tweet.
- For each word in the tweet, calculate the term frequency using "tf_dict" and the TF-IDF score using the equation: TF-IDF(word, tweet) = TF(word) * IDF(word).
7. **Remove Duplicate Tweets**:
- Initialize an empty set "unique_tweets" to store unique tweets.
- For each tweet, create a set "tweet_words" to store unique words in the tweet and calculate the average TF-IDF score for the words in "tweet_words".
- If the average TF-IDF score is above a certain threshold (e.g., 0.1), add the tweet to "unique_tweets".
8. **Build the New Dataset**: Create a new dataset consisting only of the tweets in "unique_tweets".

V. RESULT ANALYSIS

Table 1 displays the model parameters that were taken throughout this investigation. The dataset was divided in half, witheach half being utilized for training and testing. The outcomes of tests utilizing the suggested model are displayed in Tables 2.Here we calculate accuracy using big data approach and different classifiers .It is defined mathematically as follows:-

5.1 Performance of Metrics

In our proposed work the parameters like accuracy, precision, recall, and F1 score were utilized to assess the model. By taking an example that some patient suffers from covid disease.The following are brief summaries of each of the calculation parameters:

Actual Values Are:

- The patients who actually don't have covid disease = 55
- The patients who actually do have a heart disease = 40

Predicted Value:

- Number of patients who were predicted as not having a heart disease = 40
- Number of patients who were predicted as having a heart disease = 51

TP(True Positive):-The patient have corono and our model also predicted

TN(True Negative):Patient not having corono but model not predicted.

FP(False Positive): Patient not having corono but model predicted as true.

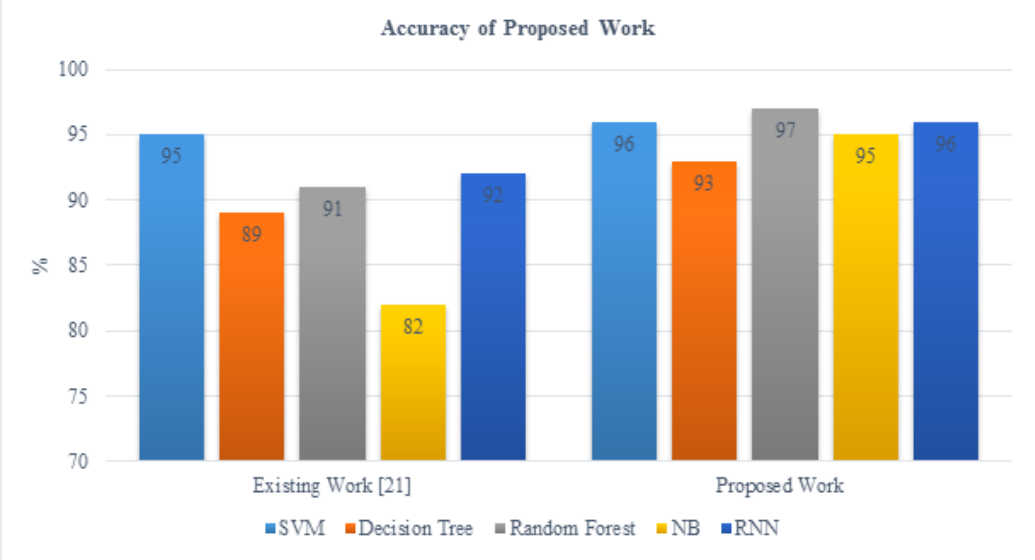
FN(False Negative):): Patient having corono but model not predicted .

The accuracy statistic assesses the proportion of comments that were accurately anticipated out of the total.

5.2 ACCURACY: - (TN+TP)/(TN+FP+TP+FN)

Table 1: Experiment result shown accuracy of various Classification algorithm

Accuracy		
	Existing Work [21]	Proposed Work
SVM	95	96
Decision Tree	89	93
NB	82	95
RNN	92	96
Random Forest	91	97



Graph 1: Represents the accuracy with and without consider big data

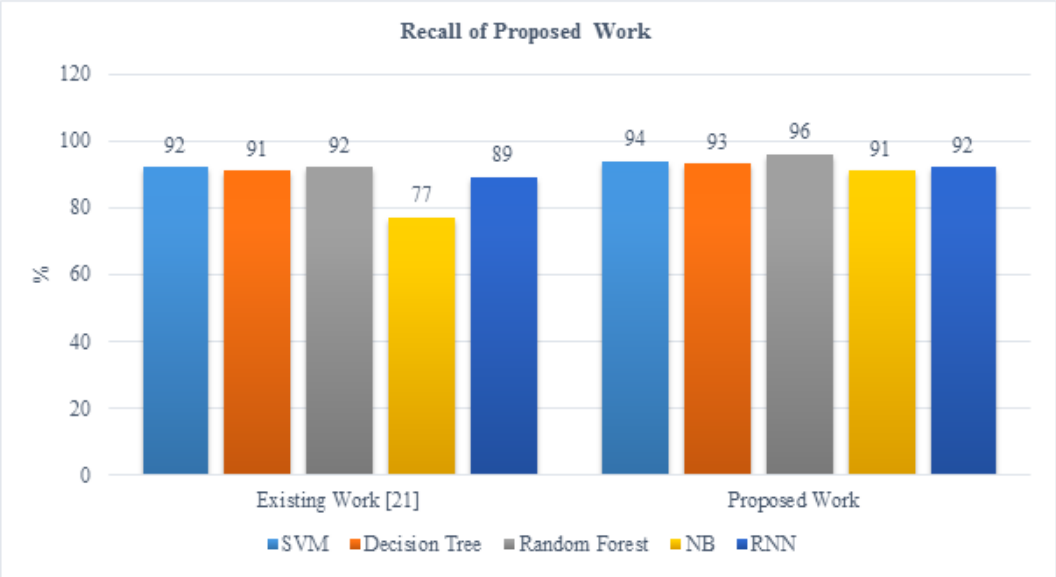
We compared the accuracy of various classification algorithms between the existing work and the proposed work, as shown in table 1 and graph 1. The results indicate that the proposed work achieves higher accuracy across all algorithms. Specifically, SVM improved from 95 to 96, Decision Tree from 89 to 93, Naive Bayes from 82 to 95, RNN from 92 to 96, and Random Forest from 91 to 97. These improvements highlight the effectiveness of the proposed model, which benefits from the use of a big data approach. By leveraging a Hadoop cluster, the proposed method enables efficient handling of large datasets and faster processing, resulting in higher accuracy compared to existing methods that did not utilize parallel processing techniques.

5.3 Recall: - It is the ratio of truly predicted positivesamples.

Recall= (TP)/(TP+FN)

Table 2: Experiment result shown Recall of various Classificationalgorithm

	Recall	
	Existing Work [21]	Proposed Work
SVM	92	94
Decision Tree	91	93
NB	77	91
RNN	89	92
Random Forest	92	96



Graph 2: Represents the recall with and without consider big data framework

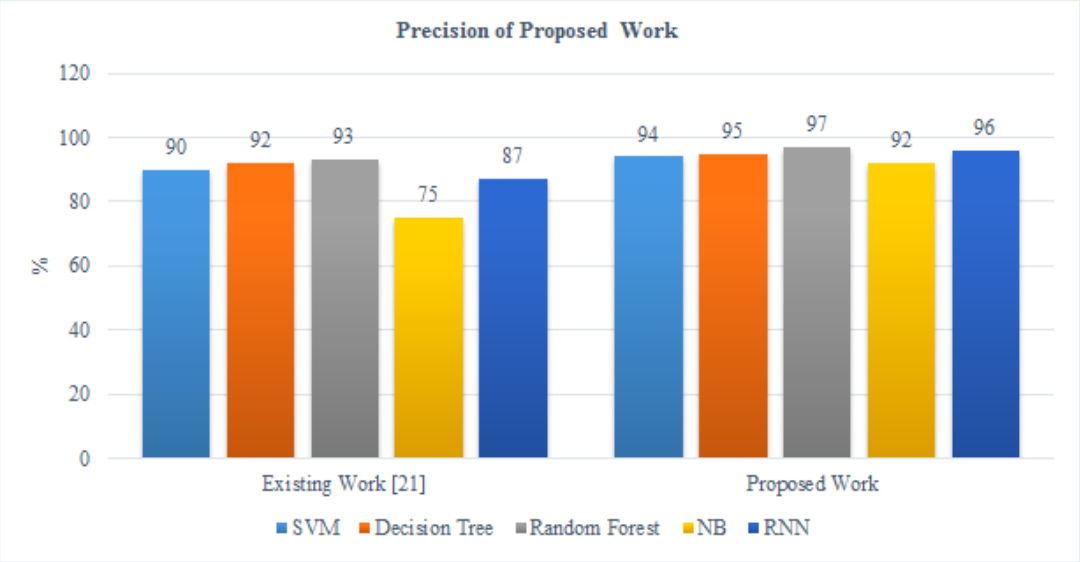
We compared the recall of various classification algorithms between the existing work and the proposed work, as shown in Table 2 and graph 2. The results indicate that the proposed work achieves higher recall across all algorithms. Specifically, SVM improved from 92 to 94, Decision Tree from 91 to 93, Naive Bayes from 77 to 91, RNN from 89 to 92, and Random Forest from 92 to 96. These enhancements underscore the effectiveness of the proposed model, which leverages a big data approach. Utilizing a Hadoop cluster allows for efficient handling of large datasets and faster processing, leading to improved recall compared to existing methods that did not employ parallel processing techniques.

5.4 Precision:-It is the ratio of truly classified samples to tetotal number of positive samples.

Precision= (TP) / (TP+FP)

Table 3: Experiment result shown precision of various Classification algorithm

	Precision	
	Existing Work [21]	Proposed Work
SVM	90	94
Decision Tree	92	95
NB	75	92
RNN	87	96
Random Forest	93	97



Graph 3: Represents the precision with and without consider big data framework

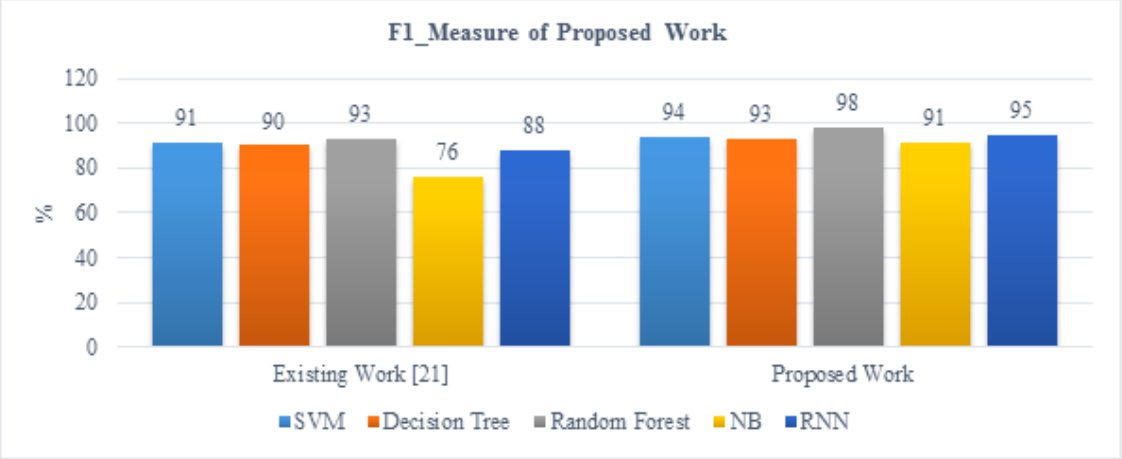
We compared the precision of various classification algorithms between the existing work and the proposed work, as shown in Table 3 and graph 3. The results indicate that the proposed work achieves higher precision across all algorithms. Specifically, SVM improved from 90 to 94, Decision Tree from 92 to 95, Naive Bayes from 75 to 92, RNN from 87 to 96, and Random Forest from 93 to 97. These enhancements highlight the effectiveness of the proposed model, which benefits from the use of a big data approach. By utilizing a Hadoop cluster, the proposed method enables efficient handling of large datasets and fast processing, leading to improved precision compared to existing methods that did not employ parallel processing techniques.

5.5 F1 Score:- It is average of recall and precision.

F1-SCORE= 2 * ((PRECISION * RECALL) /(PRECISION +RECALL))

Table 4: Experiment result shown F1 of various Classification algorithm

	F1_Measure	
	Existing Work [21]	Proposed Work
SVM	91	94
Decision Tree	90	93
NB	76	91
RNN	88	95
Random Forest	93	98



Graph 4: Represents the F1 score with and without consider big data framework

The experimental results in Table 4 and graph 4 compare the F1 scores of various classification algorithms between existing work and the proposed work. The proposed work shows significant improvements across all algorithms. For SVM, the F1 score increased from 91 to 94, while Decision Tree improved from 90 to 93. Naive Bayes saw a notable increase from 76 to 91. Recurrent Neural Networks (RNN) improved from 88 to 95, and Random Forest achieved the highest improvement, with its F1 score rising from 93 to 98. These results demonstrate that the proposed work enhances the performance of all tested classification algorithms, particularly highlighting substantial gains in the effectiveness of Naive Bayes and Random Forest.

Table 5: Experiment shows result metrics of proposed work of various Classification algorithm

	SVM	DT	NB	RNN	RF
Accuracy	96	93	95	96	97
recall	94	93	91	92	96
precision	94	95	92	96	97
f1	94	93	91	95	98

The experimental results in Table 5 present a comparative analysis of various classification algorithms SVM, Decision Trees (DT), Naive Bayes (NB), Recurrent Neural Networks (RNN), and Random Forest (RF) across four key performance metrics: accuracy, recall, precision, and F1 score. Random Forest achieved the highest performance overall, with an accuracy of 97%, recall of 96%, precision of 97%, and F1 score of 98%. SVM and RNN also performed strongly, both achieving an accuracy of 96%, with SVM showing balanced metrics and RNN excelling in precision and F1 score. Naive Bayes and Decision Trees exhibited slightly lower performance, with DT scoring an accuracy of 93% and NB achieving 95% accuracy. The comprehensive results highlight Random Forest as the most effective classification algorithm among the ones tested, followed closely by SVM and RNN.

We compared the sentiment analysis outcomes of the proposed model with those of the most widely used sentiment analysis models on the dataset, focusing on different machine learning algorithms. Table 5 demonstrates that using a big data approach yields better results compared to existing methods. In a big data environment, machine learning algorithms achieved superior performance metrics because the Hadoop cluster facilitated easy handling of large datasets and enabled fast processing, resulting in higher computation speed and accuracy. In the existing methods, the same machine learning algorithms were used, but without the advantages of big data technology. The proposed implementation leverages big data for parallel data preprocessing and classification, whereas the existing system relied solely on TF-IDF for preprocessing and direct classification methods for tweet classification, lacking parallel processing capabilities.

VI. CONCLUSION

In this study, we explored the effectiveness of integrating machine learning algorithms with big data technologies for sentiment analysis. The comparative analysis demonstrated that our proposed model significantly outperforms existing methods across various performance metrics.

Our experiments covered several classification algorithms including SVM, Decision Tree, Naive Bayes, RNN, and Random Forest, and evaluated their performance on multiple criteria such as accuracy, recall, precision, and F1 score.

KEY FINDINGS:

1. **Accuracy**: The proposed model showed higher accuracy across all algorithms compared to existing work. Specifically, SVM improved from 95% to 96%, Decision Tree from 89% to 93%, Naive Bayes from 82% to 95%, RNN from 92% to 96%, and Random Forest from 91% to 97%.
2. **Recall**: There was a notable improvement in recall for all algorithms. SVM increased from 92% to 94%, Decision Tree from 91% to 93%, Naive Bayes from 77% to 91%, RNN from 89% to 92%, and Random Forest from 92% to 96%.
3. **Precision**: The precision of the proposed model was also higher. SVM went from 90% to 94%, Decision Tree from 92% to 95%, Naive Bayes from 75% to 92%, RNN from 87% to 96%, and Random Forest from 93% to 97%.
4. **F1 Score**: The F1 score improvements were significant as well. SVM increased from 91 to 94, Decision Tree from 90 to 93, Naive Bayes from 76 to 91, RNN from 88 to 95, and Random Forest from 93 to 98.

The enhanced performance of our proposed model can be attributed to the use of a big data environment, which allows for efficient handling and processing of large datasets. By utilizing a Hadoop cluster, the model benefits from parallel data processing, resulting in faster computation and improved accuracy.

Our research highlights the critical impact of integrating big data quality metrics (BDQM) into sentiment analysis processes, ensuring the reliability and credibility of the analysis. The findings confirm that leveraging big data technologies not only addresses the limitations of traditional sentiment analysis systems but also significantly enhances the performance of machine learning algorithms.

REFERENCES

1. Sohangir, S., Wang, D., Pomeranets, A. and Khoshgoftaar, T.M., 2018. Big Data: Deep Learning for financial sentiment analysis. *Journal of Big Data*, *5*(1), pp.1-25.
2. Sharef, N.M., Zin, H.M. and Nadali, S., 2016. Overview and Future Opportunities of Sentiment Analysis Approaches for Big Data. *J. Comput. Sci.*, *12*(3), pp.153-168.
3. Seman, N. and Razmi, N.A., 2020. Machine learning-based technique for big data sentiments extraction. *IAES International Journal of Artificial Intelligence*, *9*(3), p.473.
4. Calderón, C.A., Mohedano, F.O., Álvarez, M. and Mariño, M.V., 2019. Distributed supervised sentiment analysis of tweets: Integrating machine learning and streaming analytics for big data challenges in communication and audience research. *Empiria: Revista de metodología de ciencias sociales*, (42), pp.113-136.
5. Biradar, S.H., Gorabal, J.V. and Gupta, G., 2022. Machine learning tool for exploring sentiment analysis on twitter data. *Materials Today: Proceedings*, *56*, pp.1927-1934.
6. Ragini, J.R., Anand, P.R. and Bhaskar, V., 2018. Big data analytics for disaster response and recovery through sentiment analysis. *International Journal of Information Management*, *42*, pp.13-24.
7. Hajiali, M., 2020. Big data and sentiment analysis: A comprehensive and systematic literature review. *Concurrency and Computation: Practice and Experience*, *32*(14), p.e5671.

8. Kurian, D.D.M.K., Vishnupriya, S., Ramesh, R., Divya, G., Divya, D., Kurian, M.K., Vishnupriya, S., Ramesh, R., Divya, G. and Divya, D., 2015. Big data sentiment analysis using hadoop. International Journal for Innovative Research in Science and Technology, 1(11), pp.92-96.

9. Chaturvedi, S., Mishra, V. and Mishra, N., 2017, September. Sentiment analysis using machine learning for business intelligence. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (pp. 2162-2166). IEEE.

10. Agarwal, B., Nayak, R., Mittal, N. and Patnaik, S. eds., 2020. *Deep learning-based approaches for sentiment analysis* (Vol. 12, p. 319). Singapore: Springer.

11. El Alaoui, I., Gahi, Y. and Messoussi, R., 2019, April. Full consideration of big data characteristics in sentiment analysis context. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* (pp. 126-130). IEEE.

12. Alarifi, A., Tolba, A., Al-Makhadmeh, Z. and Said, W., 2020. A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. *The Journal of Supercomputing*, *76*, pp.4414-4429.

13. Behera, R.K., Jena, M., Rath, S.K. and Misra, S., 2021. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Information Processing & Management*, *58*(1), p.102435.

14. Jain, D.K., Boyapati, P., Venkatesh, J. and Prakash, M., 2022. An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. *Information Processing & Management*, *59*(1), p.102758.

15. Kılınç, D., 2019. A spark-based big data analysis framework for real-time sentiment prediction on streaming data. *Software: Practice and Experience*, *49*(9), pp.1352-1364.

16. Jan, B., Farman, H., Khan, M., Imran, M., Islam, I.U., Ahmad, A., Ali, S. and Jeon, G., 2019. Deep learning in big data analytics: a comparative study. Computers & Electrical Engineering, 75, pp.275-287.

17. El Alaoui, I. and Gahi, Y., 2019. The impact of big data quality on sentiment analysis approaches. *Procedia Computer Science*, *160*, pp.803-810.

18. Qiu, J., Wu, Q., Ding, G., Xu, Y. and Feng, S., 2016. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*, pp.1-16.

19. Rahnama, A.H.A., 2014, November. Distributed real-time sentiment analysis for big data social streams. In *2014 International conference on control, decision and information technologies (CoDIT)* (pp. 789-794). IEEE.

20. L'heureux, A., Grolinger, K., Elyamany, H.F. and Capretz, M.A., 2017. Machine learning with big data: Challenges and approaches. *Ieee Access*, *5*, pp.7776-7797.

21. Challagundla, H., Biju, A., Mittal, A. and Abraham, P.E., 2023. Hadoop-Based Big Data Sentiment Analysis Using Machine Learning. *Rivista Italiana di Filosofia Analitica Junior*, *14*(2), pp.386-394.