Effective Machine Learning Garbage Data Filtering Algorithm for SNS Big Data Processing

Sukanya Ledalla[1*], *Saiharini* Akkenapally[1], *Rishika Reddy* Baluri[1], *Kalyani* Chittipolu[1], *Anvitha* Burri[1], and *Sujana* Kolepalli[1]

[1]Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Abstract. Social network services (SNS) are used more often today, which results in more SNS data being generated. Furthermore, greater emphasis is being placed on extracting various sorts of information through the collection, processing, and analysis of massive volumes of SNS data. Although big data processing can extract a lot of information from SNS data, it takes a long time and a lot of resources. As a result, gaining insights from SNS data necessitates a significant investment of time and money. In this section, we propose a data filtering approach for removing unnecessary SNS data from the data stream. To improve filtering accuracy, the suggested method employs Random Forest, Decision Tree, and XGBoost. Research shows that the suggested algorithm filters the experimental keywords by more than 70%.

1 Introduction

Due to the fast growth of social network services (SNS), the number of users has recently increased. As the number of mobile devices grows, so does the volume of data gathered on social networking sites. SNS is frequently used for friendship and social interactions, but in recent years, its secondary usage for collecting, analysing, and acquiring various bits of information from large datasets on SNS has significantly increased. Therefore, by examining the data on SNS, it is possible to deduce information about a variety of flows and opinions on topics such as society, the economy, and politics. However, because the data on SNS is a mixture of relevant data, data from advertisements, and beneficial data for the research itself, it is highly difficult and time-consuming to analyse it successfully. Studies on stable data collection and storage as well as effective data processing with constrained computing resources have been done recently as interest in big-data processing has grown. The value of large data prior to processing, however, is the subject of less research and study.

*Corresponding author:ledalla.sukanaya@gmail.com

As a result, this work explores effective garbage data filtering from big data to enhance the correctness and speed of real-world big-data processing analysis. By introducing machine learning into the process of removing useless data, this study explicitly aims to increase filtering accuracy. Consequently, in this paper, we present a method that increases garbage data filtering accuracy using cyclic learning, and we use experiments to demonstrate the programme's efficacy.

2 Literature Survey

We review and analyse studies pertaining to big data filtering in this section. The studies already done on this subject are listed below. First off, machine learning is one method for more effectively analysing massive amounts of data. Recent trends in machine learning studies for big data processing are described by Qiu et al. in their article [1]. There is discussion of the most recent techniques, such as deep learning, distributed and parallel learning, and representation learning. Technologies for big data networking come with a number of problems and challenges, as Suthanharan et al. [2] highlighted. The 3C factors of complexity, continuity, and cardinality are used to define the features of big data in order to better understand the analysis in this study. Jarrah et al.'s [3] investigation into effective machine learning methods for handling massive data sets.

Using implementation engines including MapReduce, Spark, Flink, Storm, and H2O, Xing et al. [4] analysed and contrasted the Hadoop environment, a typical machine learning architecture. A look was also given to machine learning frameworks and libraries, including Samoa, Mahout, and MLlib. For use with big data in healthcare, Chen et al. [5] suggested a method for disease prediction based on machine learning, and trials showed the proposed algorithm to be effective. A massive data processing method using multiple machine learning techniques is also suggested by certain publications [6-8]. Filtering is a different approach for effectively processing large amounts of data. The research that has already been done on this topic is listed below. Ou et al. [9] published a method for filtering text message spam characters using a sentence similarity metric. However, no suggestions for cyclic machine learning-based data filtering methods were made. Using API call data and machine learning, Cho et al. [10] examined the classification of Windows executable files. The success of execution file classification is assessed using a variety of API call information refinement techniques and machine learning algorithms. Sadly, only Windows executables are included in the classification.

Choi and colleagues [11] looked at the use of machine learning techniques to analyse security patterns, including item categorization, positive or negative appraisal, and core keyword association. There was no further method of pattern recognition in this experiment. Yang et al.'s [12] study on social media big data filtering resulted in the creation of an early warning system for adverse drug interactions. In this study, data filtering was accomplished using supervised learning techniques and trained classifiers.

Hu and colleagues [13] proposed a collaborative filtering approach based on clustering for big data applications. This approach uses two steps to speed up data processing: clustering and collaborative filtering. Numerous studies have also suggested various data filtering methods for handling vast amounts of data. Comparing the study to past research, the following characteristics are present: Second, the recommended method may increase the precision of SNS data analysis by applying machine learning to categorise information into garbage, ads, and particular data. During the learning process, initial data are used as inputs, and cyclic learning improves classification accuracy.

3 System Architecture

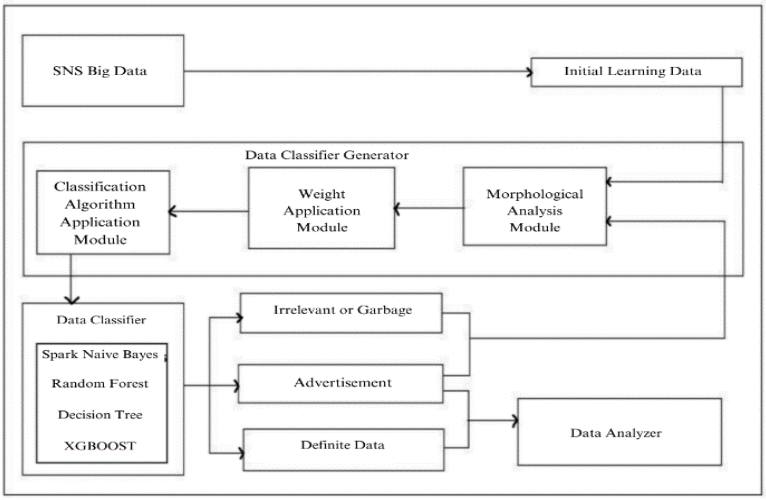

Fig. 1. System Architecture

4 Methodologies

SNS information includes content relevant to viewpoints being expressed in a variety of contexts, including economics, society, and culture. So it is possible to learn about various flows and ideas on subjects like society, the economy, and politics by analysing the data on SNS. The data on SNS is a mixture of useful data, data from advertisements, and positive data that is helpful to the actual study; therefore, it is highly challenging and time-consuming to analyse it well. Studies on trustworthy data collection, storage, and processing with constrained computing resources have been carried out as an interest in big-data processing has grown. On the utilisation of huge amounts of data before processing, there are, nevertheless, few studies and researches available.

1. It is quite challenging.
2. Time intensive

This work investigates the effective removal of junk data from enormous amounts of data to improve the accuracy and speed of data analysis in actual big-data processing. By introducing machine learning into the process of removing useless data, this study explicitly aims to increase filtering accuracy. As a result, we provide a technique in this research that uses cyclic learning to improve the SNS large data trash filtering accuracy. We also use experiments to show the programme's effectiveness.

1. Effectively eradicating garbage data from big data
2. Increasing the data analysis's precision and speed in the process

To implement the project author has used following modules

1) Upload SNS Dataset: Using this module, we will upload the Social Network Services dataset to the application.
2) Data Classifier Generator: Using this module, we will read all dataset tweets and then calculate the weight of each word by using its occurrence in the tweets.
3) Data Classifier Generator: Using this module, we will read all dataset tweets and then calculate the weight of each word by using its occurrence in the Tweets.

4) Data Classifier using SPARK Naive Bayes: by using tweets weights, we will train the SPARK Naive Bayes algorithm, perform prediction on test data, and then calculate its prediction accuracy.
5) Run Extension Random Forest: Using this module, we will train the Extension Random Forest algorithm, and then perform prediction on test data, and then calculate its prediction accuracy.
6) Run Extension Decision Tree: Using this module, we will train the Extension Decision Tree algorithm and then perform prediction on test data, and then calculate its prediction accuracy.
7) Run Extension XGBOOST: Using this module, we will train the Extension XGBOOST algorithm, and then perform prediction on test data, and then calculate its prediction accuracy.
8) Data Analyzer: Using this module, we will upload test data, and then the trained model will classify tweets into one of the 3 groups called 0 (Garbage), 1 (advertisement), or 2 (definite).
9) Accuracy Comparison Graph: Using this module, we will plot an accuracy comparison graph between all algorithms.

5 Implementation

Nowadays, practically everyone uses social networking services to publish their opinions on a variety of subjects, including politics, online purchases, and a wide range of other issues. These tweets or posts frequently contain irrelevant information known as advertisements, garbage (meaningless), and definitive (important or relevant posts), and because of the large number of users, huge amounts of data are gathered (big data), making it challenging to process relevant information and ignore garbage data. The author of the previous paper uses big data technologies like Hadoop, Spark, or Mahout, which can process data more quickly and aid in retrieving useful data more quickly, to address the aforementioned problems. Using the Naive Bayes machine learning algorithm, posts or tweets are divided into three categories: garbage, advertisements, and definitive (relevant posts).

Each post will have its own morphological weight (the average frequency of each job, also known as weight), which aids machine learning in classifying groups of posts. If a post contains garbage or advertising, the same word can appear more frequently, increasing its weight. If the weight increases, the post will be regarded as garbage or advertising using machine learning. On the basis of tweets data from various groups, the Naive Bayes algorithm will be trained to produce a trained model.

When we apply test data to a trained model, Naive Bayes calculates weight and categorises the post as either garbage, an advertisement, or definitive based on the weight value. We choose to include advanced algorithms, including Decision Tree, Random Forest, and XGBOOST, which perform better than Naive Bayes. We downloaded our own tweets from various categories.

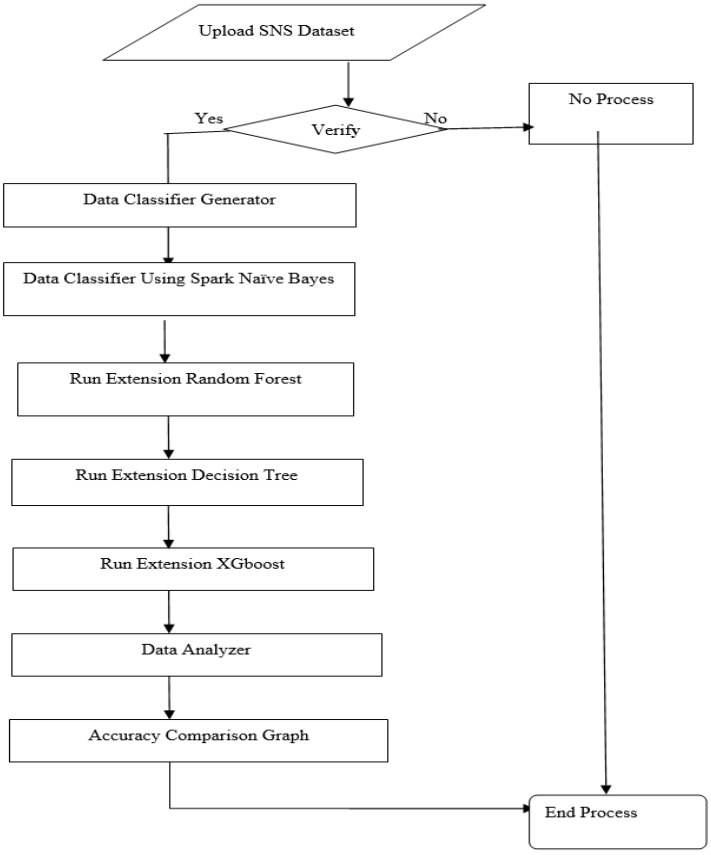

Fig. 2. Dataflow diagram

Algorithms: A decision tree is a diagram of a tree where the branches are alternatives and the leaves are associated risks, expenses, outcomes, or probabilities. Using this graphic, judgements can be made in computer programming or business.

Random Forest: In classification and regression problems, supervised machine learning methods like random forest are often used. Using their average in the case of regression and their majority vote in the case of classification, it builds decision trees from multiple samples.

Naive Bayes: A naive Bayes classifier is an algorithm that uses the Bayes theorem to classify objects. The naive Bayes classifier makes the assumption that the attributes of the data points are strongly or naively independent. Examples of typical uses for naive Bayes classifiers include spam filters, text analysis, and medical diagnosis.

XGBOOST: "Extreme gradient boosting" is referred to as "XGBoost." It selects the best tree model by making use of closer approximations. Boosting: Using random sampling and replacement, N new training data sets are generated from the original dataset; certain observations may appear in several training datasets.

6 Experimental Results and Discussions

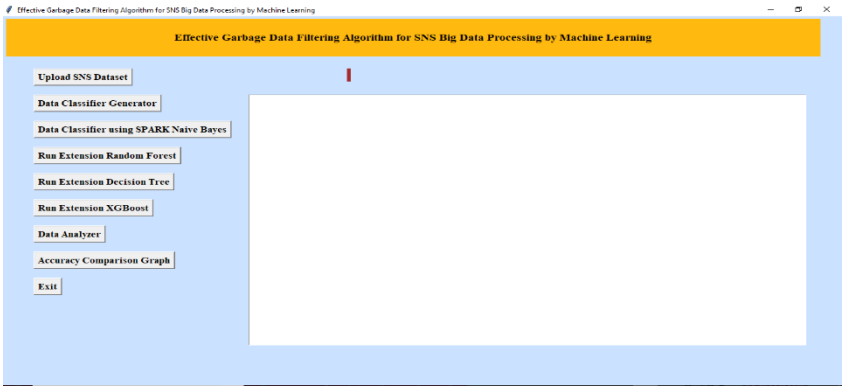

Fig. 3. Home Screen

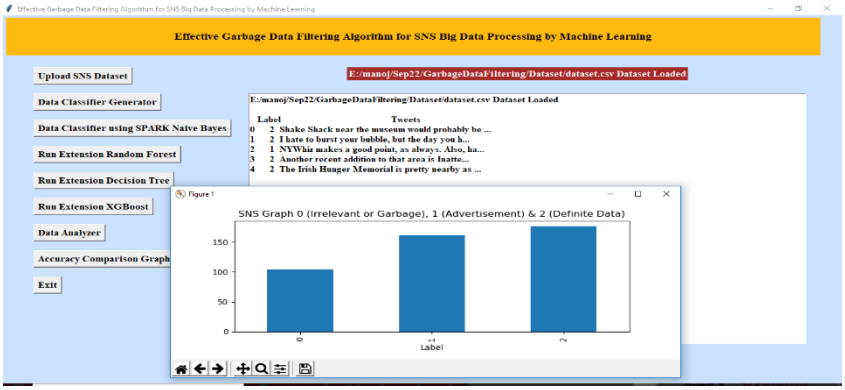

Fig. 4. Dataset Classification

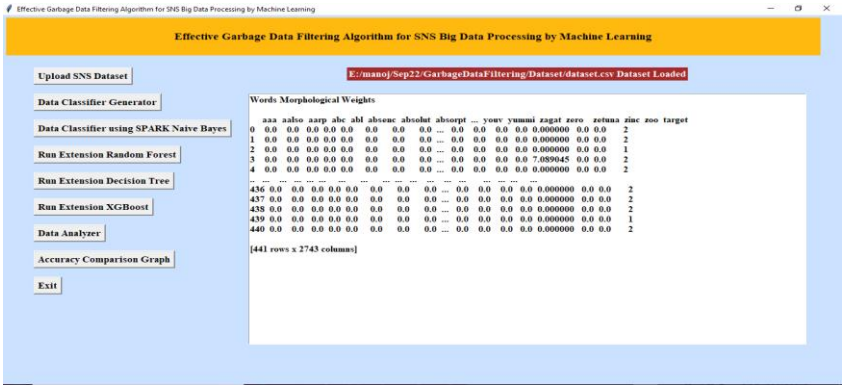

Fig. 5. Morphological weights

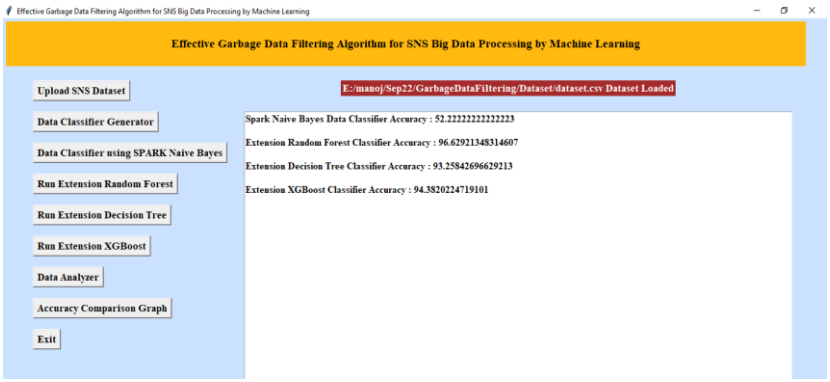

Fig. 6. Accuracy of all algorithms

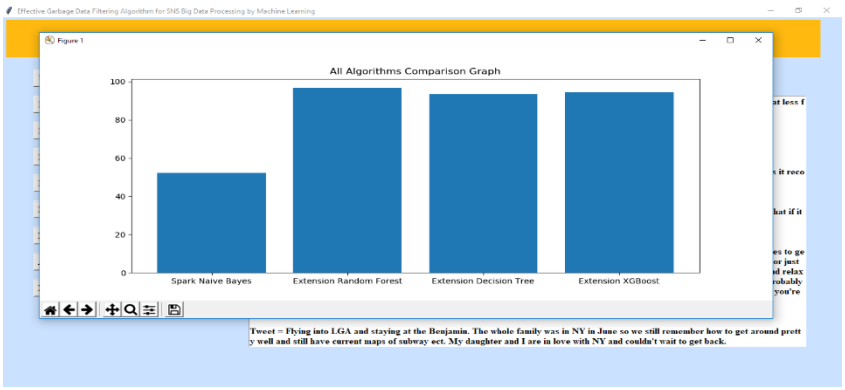

Fig. 7. Data analyze

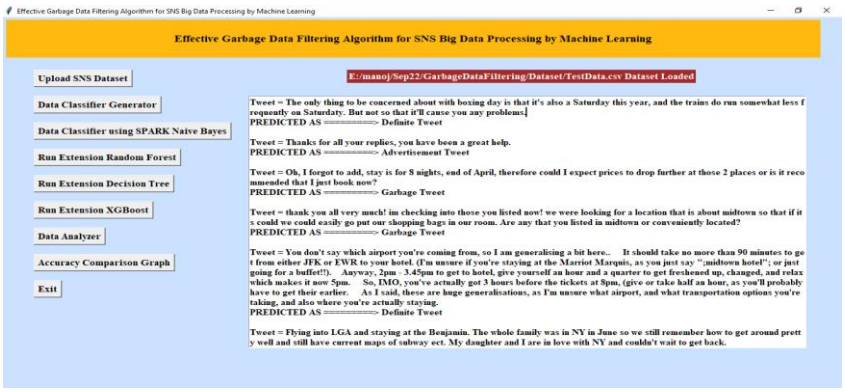

Fig. 8. Accuracy comparision graph

7 Conclusion

In this study, we created and evaluated a machine learning-based method for removing garbage data from SNS. The method uses machine learning to categorise unstructured data into three categories: garbage, advertising, and specific data. It will increase the accuracy of unstructured data analysis in SNS. When compared to the correct answer set, data filtering for the accuracy experiment showed an accuracy of 74.45%. It has been determined as a result that it is helpful in large data processing scenarios where a lot of data needs to be handled quickly. This leads to the following summary of the study's contribution: In the beginning, this research offered a useful garbage and advertisement data filtering technique that may be used in systems that process enormous amounts of data. By picking and processing only the most valuable data from a significant amount of data generated in daily life, such as SNS big data, it aims to increase the efficiency of big data processing. Second, we demonstrated a recursive machine learning data filtering method. Big data from SNS was utilised to create initial learning data, which was then used for data filtering. The filtered data was then used as learning data in the suggested system to increase the filtering's precision. Our research shows that SNS Big Data can be successfully analysed, and valuable data may be extracted from SNS Big Data in a variety of disciplines.

References

1. J. Qiu, Q. Wu, G. Ding, Y. Xu, S. Feng, “A survey of machine learning for big data processing,” EURASIP J. Adv. Signal Process. vol. **2016**, pp. 1-16, (2016)
2. S.Suthanharan, “Big data classification: Problems and challenges in network intrusion prediction with machine learning,” ACM SIGMETRICS Perf. Eval. Rev. vol. **41**, pp. 70-73 (2014)
3. O. Jarrah, P. Yoo, S. Muhaidat, G. Karagiannidis, K. Taha, “Efficient Machine Learning for Big Data: A Review,” Big Data Res. Vol**. 2**, pp. 87-93 (2015)
4. E. Xing, Q. Ho, W. Dai, J. Kim, Y. Yu, “Petuum: A New Platform for Distributed Machine Learning on Big Data,” IEEE Trans. Big Data, vol. **1**, pp. 49-67 (2015).
5. M. Chen, Y. Hao, K. Hwang, L. Wang, L. Wang, “Disease Prediction by Machine Learning Over Big Data From Healthcare Communities,” IEEE Access, vol. **5**, pp. 8869–8879, (2017)
6. M. Gunasekaran, V. Vijayakumar, R. Varatharajan, K. Priyan S. Revathi, H. Ching-Hsien, “Machine Learning Based Big Data Processing Framework for Cancer Diagnosis Using Hidden Markov Model and GM Clustering,” Wireless Personal Communications, vol. **102**, pp. 2099-2116 (2018)
7. W. Xiaofei, Z. Yuhua, L. Victor, G. Nadra, J. Tianpeng, “D2D Big Data: Content Deliveries over Wireless Device-to-Device Sharing in Large-Scale Mobile Networks,” IEEE Wireless Communications. vol. **25**, pp. 32-38 (2018)
8. Z. Zhenhua, H. Qing, G. Jing, N. Ming, “A deep learning approach for detecting traffic accidents from social media data,” Transportation Research Part C: Emerging Technologies, vol. **86**, pp. 580-596 (2017)
9. S. Ou; J. Lee, “Implementation of a Spam Message Filtering System using Sentence Similarity Measurements,” KIISE Trans. Comput. Pract. (KTCP), vol. **23**, pp. 57-64 (2017)

10. D. Cho; K. Lim; S. Cho; S. Han; Y. Hwang, "Classifying Windows Executables using API-based Information and Machine Learning," J. KIISE, vol. **43**, pp. 1325-1333 (2016)
11. H. Choi, J. Park, "Security tendency analysis techniques through machine learning algorithms applications in big data environments," J. Digit. Converg. vol. **13**, pp. 269-267 (2015)
12. M. Yang, M. Kiang, W. Shang, "Filtering big data from social media – Building an early warning system for adverse drug reactions," J. Biomed. Inf. vol. **54**, pp. 230-240, (2015)
13. R. Hu, W. Dou, J. Liu, "ClubCF: A Clustering- Based Collaborative Filtering Approach for Big Data Application," IEEE Trans. Emerg. Topics Comput. vol. **2,** pp. 302-313 (2013)