Received August 11, 2020, accepted August 29, 2020, date of publication September 10, 2020, date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023346

Product Pre-Launch Prediction From Resilient Distributed e-WOM Data

SANDHYA NARAYANAN[1], PHILIP SAMUEL[2], AND MARIAMMA CHACKO[3]

[1]Information Technology Division, Cochin University of Science and Technology, Kochi 682022, India
[2]Department of Computer Science, Cochin University of Science and Technology, Kochi 682022, India
[3]Department of Ship Technology, Cochin University of Science and Technology, Kochi 682022, India

Corresponding author: Sandhya Narayanan (nairsands@gmail.com)

ABSTRACT Pre-launch success prediction of a product is a challenge in today's electronic world. Based on this prediction, industries can avoid huge losses by deciding on whether to launch or not to launch a product into the market. We have implemented a Multithreaded Hash join Resilient Distributed Dataset (MHRDD) with a prediction classifier for pre-launch prediction. MHRDD helps to remove the redundancy in the input dataset and improves the performance of the prediction model. Large volume of e-Word of Mouth (e-WOM) data like product reviews, comments and ratings available on internet about products can be used for pre-launch product prediction. In MHRDD, to identify features a distance similarity score is used. In order to remove duplicates, a hash key and join operations are used to create a hash table of significant features. With in-memory computations and hashing on the join operations, this model reduces redundancy of data. This model is scalable and can handle large datasets with good prediction accuracy. This paper presents a novel big data processing method that predicts product success before its launch in the market. Proposed method helps to identify features that are significant for the product to be successful. Based on the pre-launch prediction, companies can reduce cost, effort and time with improved product success.

INDEX TERMS Big data analytics, product pre-launch prediction, resilient distributed dataset, redundancy elimination.

I. INTRODUCTION

In social networking sites and other e-commerce applications, the product reviews are available in huge volumes. To a certain extent, product sale depends on customer reviews. Life of product depends on its quality where as online product reviews help to improve the product quality [1]. Online product reviews help in identifying the key changes needed for the product. Based on the customer suggestions manufacturer can design a good quality product. WOM is one of the information transfer methods which gives direct opinions about the product for the customers. Similarly, online customer reviews from the users are another source of information about a product [7]. Customer feedback, criticism, comments, suggestions, reviews, ratings, opinions etc which are available in the online mode as huge data are treated as Electronic Word of Mouth. Usually word of mouth refers to customer feedback of products or services in direct marketing. After online purchasing of the product, customers give their valuable pros and cons about the product to help others. No one-to-one physical communication with customers are there. E-WOM is a rich source of information for manufacturers to launch new products or improved versions of their products. Anyhow, techniques to extract useful decision-making information from this kind of data are rare. This is challenging as the data is unstructured, redundant and with huge volume [37], [44].

Success of a product depends on product reviews. Online shopping sites give several benefits to customers [7]. Online marketing methods can acquire a large number of customer suggestions in the form of product reviews and descriptive information about the product without any marginal cost [25]. Companies believe that almost all sites should provide effective content about the products to build loyalty [2], [5]. Poor quality products potentially affect the goodwill of industries. In the design stage itself, we can include quality features to improve the success of the product. The users should provide their valuable true reviews about the products which they use. Customers may give duplicate product reviews, which are redundant [3], [4]. These redundant reviews are handled in our proposed system. Usually, unauthorized users or biased interested users can give duplicate reviews which can devitalize the product sale [9]. Thus, to promote the manufacturing

The associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Dey.

quality, the proposed model for successful product launch prediction can be adapted.

In online marketing strategies, mainly three types of product reviews are examined: volume, strength, and disbandment [9]. A large volume of product reviews on shopping sites, blogs and forums create awareness among the users about the product. The rating on the products and the ratio of positive to negative opinions about the product are considered as the strength. Higher-strength implies better quality [34], [36]. Now studies are there for e-WOM product ratings as a revenue-forecasting tool for products such as television shows, movies, books and other products [8], [9]. The disbandment of communication measures how fast these customer suggestions extend over communities [10]. Thus, scalable infrastructural models are used to handle this type of big data [16], [19]. Compared to traditional database management systems, the challenges of big data [28] are more complex [20], [21], [22].

The online media has changed the way people express themselves and interact with others. Anyone can post reviews of products at online shopping sites (e.g., Flip kart, Amazon, Club factory, Snap deal) and they can express their views about the pros and cons of a product. It is identified that such user-generated contents, on online **s**ites provide useful information that can be exploited for different applications. Several works exist on extracting positive and negative reviews using natural language processing techniques and spam reviews recognition [11]–[13], [15]. These works do not provide any method on the feature extraction of these reviews, which helps to build quality products. Since quality control over reviews is not there, anyone can write on the web, which results in many duplicate reviews and spam reviews [23], [30].

In this paper, we focus on e-WOM customer suggestions about the product, which contain information of user views on the product and are useful to both prospective users and product manufacturers. The problem of data redundancy and significant feature extraction of the customer reviews dataset are handled to enhance the quality of dataset formation and cleaning. This can be helpful for a successful new product launch. The input raw datasets used for analysis contain duplicates reviews of products that are alternatives to the same product. This duplication is possible due to varied sizes, color and material for clothing items, blue-ray and DVD versions (for movies), color of the casing for mobile phones etc. The customers decide a specific alternative of the product, e.g., ''red case of iphone7S'', but their customer reviews and ratings also appear in the dataset review of the web pages of all the several variants of the product. Since the e-WOM data are acquired by crawling, all the same, product options have the same reviews and ratings, leading to redundant reviews [45].

In our proposed paper, problems such as data redundancy, scalability and prediction accuracy of huge datasets have been handled. With this, a scalable model to predict the success or failure of a new product prior to its launch in the market is implemented. Thus, it contributes to the manufacturing of the product with desired quality. Implementation of the model is achieved by Multithreaded Hash-join Resilient Distributed Dataset (MHRDD) method with machine learning prediction methods. This results in enhanced performance of the prediction model by removing the redundancy in the input dataset. The result analysis exhibits that the proposed model is more effective. Proposed product pre-launch prediction helps to design a good quality product, which can be helpful for the manufactures as well as consumers.

This paper is organized as follows. Section 2 reviews the literature. Section 3 describes our methodology. The result analysis is discussed in section 4 and the conclusion is provided in Section 5.

II. RELATED WORKS

The state-of-the-art techniques on duplicate data removal and data cleaning are mainly dependent based [51], [52], [54], and adaptive window-based [53]. The data cleaning method implemented by Bertossi *et al.* [51] describes the matching dependencies. This single matching dependency on tuples produces a collection of clean occurrences concerning a particular pair of bad tuples. But this method raises the computational complexity problem of query answering and also necessitates a space requirement problem.

Another adaptive approach is the Sorted Neighborhood Method [50], with respect to a key. This method sorts the data and then passes a window across the data matching unique documents that arrive within the same window. The disadvantage of this method is the fixed window size. For small window size, the redundant data is missed, and for large window size, unnecessary comparisons occur.

Raymond *et al.* [18] paper proposed the technique for untrusted review on spam detection and the same was done using text mining model concatenated with the semantic language model. Non-review spam detection is done by identifying different SVM classifiers [29] for different analysis methods. The results which obtained from this semantic language modelling and batch processing type text processing computational model are effective for the detection of untrusted customer reviews. This is also effective if spammers exercise confusion strategies in customer reviews [18], [32]. Gutierrez *et al.* [54] suggested an automated interpretation of complex text data for a crisis event. An unsupervised learning model called random forest method is trained for data pre-processing and feature extraction. Like the random forest, an ensemble approach is weak in interpretation and prediction. Also, the random forest method utilizes more memory and the application execution time is more considerable. Gutierrez *et al.* [31] illustrate a linear predictor model to preserve a subset of the essential features based on their association with the solicited output values. The model modifies the initial data set, exhibiting its various significant variable features. The authors considered nearby 10,000 features in the experiment. The limitations of the model is that it is sensitive to data redundancy and outliers.

The proposed MHRDD approach makes use of in-memory computation with distributed computing which makes the application execution faster. As the data increments or updation occur, proposed multithreaded hash join can optimize for multiple queries at the time of development with that massive data. Data redundancy can be eliminated to improve the prediction accuracy of the application. Significant features are identified which helps to improve the reliability of the predictive model.

Recently, Wang *et al.* [33] proposed the Online Group Feature Selection algorithm in which data instances sequentially added to the application. Another online feature selection method assumes that the total number of data elements is fixed while the number of features changes over time. Perkins *et al.* [42] proposed different stages of gradient descent approach with the grafting algorithm (Graft-GD). This grafting technique treats the feature selection methods as essential for predictor learning in a framework. The model works with the iterative programming method. The gradient descent model trains the predictor model and builds up the feature set.

Leung *et al.* [27] discussed an Alpha investing method, which appends features to a prediction model. The system builds with a dynamically generated candidate feature set. This Alpha investing technique requires the experience of the original input feature set, and this model never evaluates the duplicates among the selected features. The major disadvantage of the above feature selection methods is high computational cost and large dimensionality. Liu *et al.* [43] suggested a model for movie review summarization and rating. Latent Semantic Analysis (LSA-based) method is used to extract variable features of the product and summarize the product based on various features. The limitation of the LSA-based approach is that it cannot be realized efficiently; hence, it is hard to index based on specific dimensions. Thus decreases the prediction accuracy in unstructured, massive datasets.

Based on the grouping semantics, Balasundaram and Vengadeswaran [17] suggested an optimal data placement approach, which can reduce the time for query execution and query latency. This approach is implemented using the Hadoop framework with map-reduce processing. Hadoop's data placement approach designates the data chunks arbitrarily over the group of nodes without examining execution parameters. In the proposed MHRDD, multiple map tasks with in-memory computation are achieved by the resilient distributed dataset (RDD). The massive iterative processing applications in-memory computation makes the proposed approach faster than map-reduce data processing.

Pre-processing of the massive unstructured dataset plays a crucial role in predictive analytics applications. Manohara *et al.* [58] implemented feature extraction using dimensionality reduction for processing large financial dataset. Proposed MHRDD method with proper data preprocessing improves the prediction accuracy of the model. The proposed method eliminates the unwanted data; also, significant features are extracted in a distributed and reliable manner with fast in-memory computation. Chen *et al.* [57] described a Tight center loss function approach for iris image feature extraction from the large dataset using Tensor flow analysis. Time taken for processing the application is large compared to the proposed distributed in-memory computation. Proposed method over come the limitation of variable length sequence processing of symbolic looping approach.

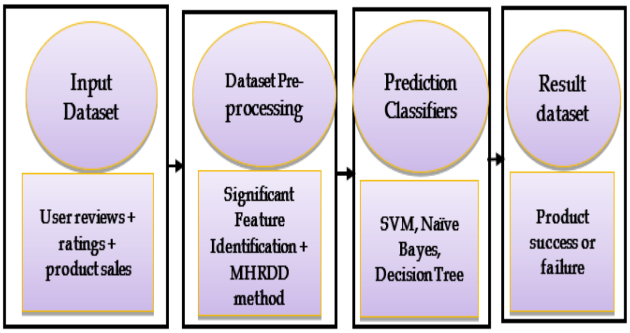

Input Dataset
User reviews + ratings + product sales
Dataset Pre-processing
Significant Feature Identification + MHRDD method
Prediction Classifiers
SVM, Naïve Bayes, Decision Tree
Result dataset
Product success or failure

FIGURE 1. Functional block diagram for the proposed product pre-launch prediction.

In our proposed paper, data pre-processing problems such as data redundancy, significant feature identification are handled properly. Also scalability and the prediction accuracy of huge datasets have been handled in a better manner. A scalable model to predict the success or failure of a new product before its launch in the industry is developed. The related works show that the proposed model is more effective compared to the state-of-the-art techniques.

III. PROPOSED METHODOLOGY

In this proposed work, the success or failure of a product is predicted before its launch with distance similarity score and Multithreaded Hash-join Resilient Distributed Dataset (MHRDD) method using appropriate prediction classifiers. Fig 1. shows the functional block diagram for the proposed product pre-launch prediction. The system consists of dataset aggregation, data pre-processing i.e., construction of duplicate data removal method and feature identification, building prediction classifier and testing. In this system dataset collection phase learns different product features from customer reviews, product ratings and product sales details.

Intended purpose is to acquire better knowledge of the input dataset for good prediction accuracy. One of the important steps consists of pre-processing of the dataset. In this phase, duplicate customer reviews are eliminated, best features are identified as well as missing and irrelevant data are handled. Resilient distribution of Spark framework is adopted to handle this large dataset in a scalable and fault tolerant manner. In the final phase, prediction accuracy is tested and compared using different classifiers.

A. DATASET AGGREGATION

E-commerce sites (like Amazon, Flipkart, Club-Factory, Snapdeal, etc.) provide customer reviews and ratings. Several datasets are available as public datasets [5], [6]; the proposed

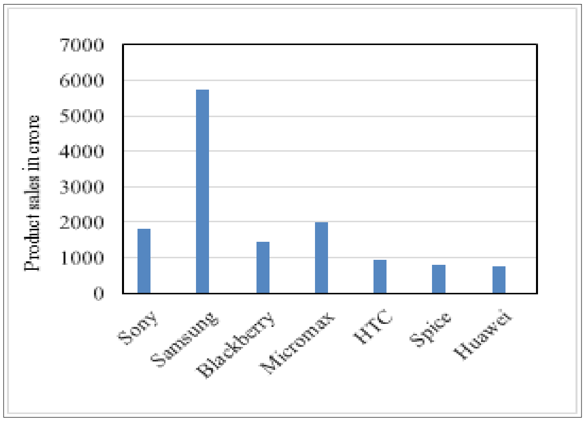


FIGURE 2. Sales average of 7 brands of mobile phones.

methodology can be used for different datasets. The product considered in this work is seven brands of mobile phones. The dataset we have considered contains reviews for a period of 3, 6, 12, 18 and 24 months, ratings and product sales details [5], [6]. Table 1 shows the sample customer review and ratings. Fig 2. shows the aggregate revenue generated in one year of respective mobile brand sales.

The e-WOM dataset consists of the pros and cons of the product. The software and hardware descriptions of the product is taken into consideration for the design of the product. Hence, hardware and software components of the product are considered as a product feature. In the raw input dataset, fifty five variable features exist. Identifying a feature is essential for this model to improve system performance. A distance similarity score approach has been implemented to identify the significant customer review features. Features having a distance similarity score value greater than 0.5 are considered for further processing.

Previous product sales in crore for different mobile brands in 12 months is shown in Fig 2. Product sale feature along with the customer reviews plays an important role for better prediction.

B. DATASET PRE-PROCESSING

Data Pre-processing uses resilient distributed dataset of Spark framework. Significant features of input dataset are identified using distance similarity score in the feature identification phase. Using significant features, duplicates and irrelevant data are removed by applying multithreaded hash join resilient distribution method.

1) DATASET PROCESSED IN RESILIENT DISTRIBUTION

Resilient distributed dataset (RDD) is collection of files with file partitioning in it. RDD's can be built using functions called transformations and actions. RDD is written through deterministic transformations. Hence, resilient distribution limits huge volume of writes but it permits fault tolerance. Lineage property of RDD helps to recover the information which avoids check point overhead. If failure occurs in any partitions of RDD, they can be recomputed parallel on different nodes without having to restart the complete program [26], [35].

The main actions of RDD are:

- Reduce (): RDD which aggregates the elements.
- Collect (): An array of all RDD's are returned to the result set.
- Count (): Total RDD element count is returned.
- saveAsTextFile(): Save the RDD elements into the text file..

The main Transformations of RDD are:

- map(x=>y): RDD in which the elements processed independently.
- filter(x=> TRUE): RDD in which the function returns true value as result.
- flatmap(x=>Range(x,y)): Each element is mapped to zero or more.
- ReduceBykey(x,y): RDD elements with similar key are grouped.
- Join(x[]): equi-join operation is performed on the key elements.

An RDD framework consists of a Master program, cluster manager and several slave nodes as workers. The workers consist of memory nodes and data nodes. The applications work as independent processes on each worker node, coordinated by the collect function in the main program. To program on a cluster, the collect function can connect to several types of partitions, which allocate resources across applications. Once connected, framework acquires memory nodes in the cluster, which are processes that run computations and store data for our application. Next, partition sends our application code to the data nodes. Finally, collect program sends data for the memory nodes to run [35].

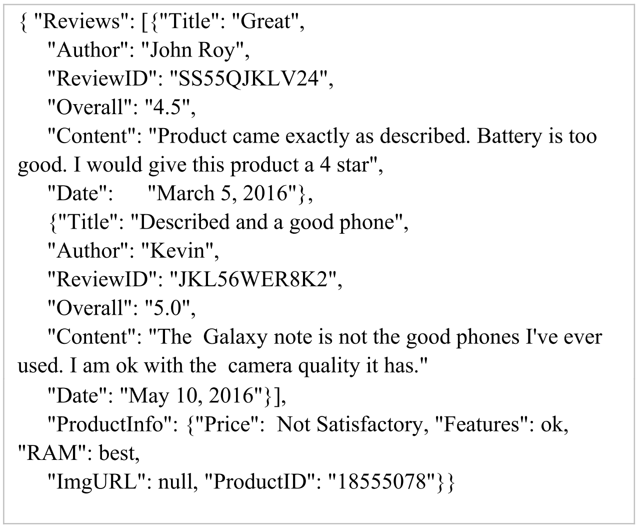
2) FEATURE IDENTIFICATION

In data cleaning, feature identification plays a major role. In the customer review dataset of mobile phone, a large number of features exist. Identifying significant feature is important for this model to improve the prediction accuracy. This stage is again subdivided into product feature identification, opinion identification of product and weighted feature based on opinion rank.

In product feature identification, significant features are identified using distance similarity measure. In opinion identification of the product, polarity of the customer review is measured. Based on the opinion rank feature weight is calculated. A distance measure similarity score ratio has been implemented to identify significant features. Features distance similarity score greater than 0.3 is considered as significant. In this work the threshold is kept as 0.3.

Let R_i and R_j be the reviews of customer Ci. For each customer Ci the feature count of review R_i is denoted as $f_{ci}(R_i)$, feature count of review $R_j$is denoted as $f_{ci}(R_j)$, and N be the total number of features. If the count of R_i is less than R_j, then the feature ratio δ for each customer Ci is

TABLE 1. Sample customer reviews and ratings dataset.



```
{ "Reviews": [{"Title": "Great",
    "Author": "John Roy",
    "ReviewID": "SS55QJKLV24",
    "Overall": "4.5",
    "Content": "Product came exactly as described. Battery is too
good. I would give this product a 4 star",
    "Date":     "March 5, 2016"},
    {"Title": "Described and a good phone",
    "Author": "Kevin",
    "ReviewID": "JKL56WER8K2",
    "Overall": "5.0",
    "Content": "The  Galaxy note is not the good phones I've ever
used. I am ok with the  camera quality it has."
    "Date": "May 10, 2016"}],
    "ProductInfo": {"Price":  Not Satisfactory, "Features": ok,
"RAM": best,
    "ImgURL": null, "ProductID": "18555078"}}
```

computed as

$$\delta = \frac{f_{c_i}(R_i)}{f_{c_i}(R_j)} \tag{1}$$

If feature count of R_j is less than R_i then,

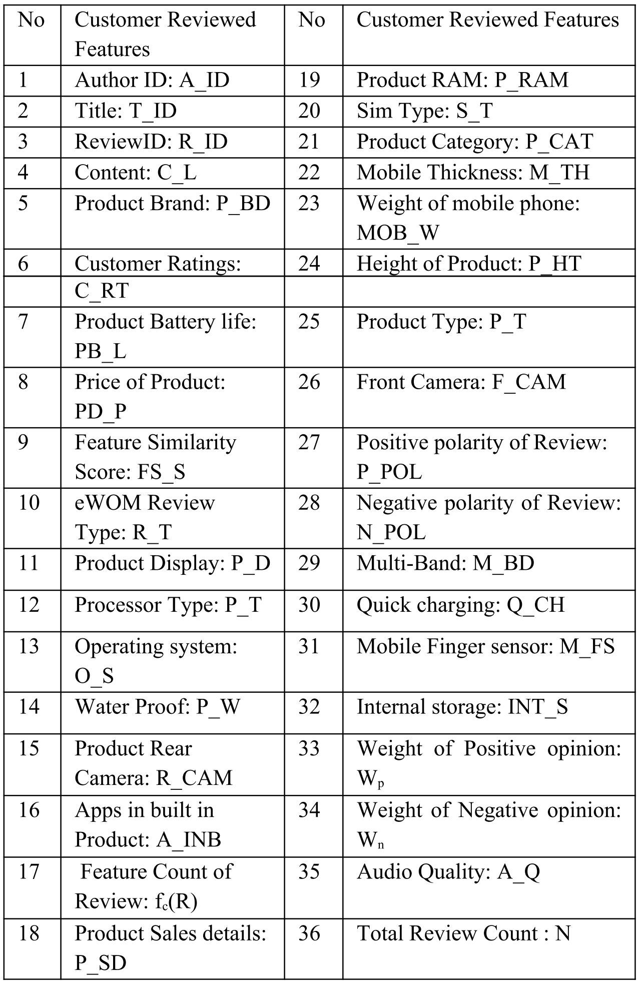
$$\delta = \frac{f_{c_i}(R_j)}{f_{c_i}(R_i)} \tag{2}$$

If feature count of R_i equals that of R_j then this ratio is considered as 1. Using the feature ratio for each customer Ci, distance similarity score is computed to identify the customer features. Customer review opinion has to be identified in-order to find the polarity of the review. Hence, to identify the review opinion we parse the sentence using MINIPAR [41]. Senti-WordNet [14] is used to classify the identified opinions from the polarity of individual reviews. For each review, the opinion sentences are examined and mapped into the positive or negative class based on the polarity value of the associated opinions. The overall weight of a feature is calculated by difference between the two polarity values of the opinion word multiplied with the number of sentences in which that opinion word repeats. Let W_p be the weight of the positive opinion of the feature i.e., positive polarity value of the opinion word multiplied with the number of sentences in which the opinion word repeats. and W_m be the weight of negative opinion of the feature, i.e., negative polarity value of the word multiplied with the number of sentences in which the opinion word repeats. Let z be the number of features in a review comment.

$$O = \sum_{i=1}^{z} \left(W_{pi} - W_{mi}\right) \tag{3}$$

Let's take an example for the feature battery life, W_p value is 0.871 and W_m value is 0.214 from the review content. Then the overall weight O_w is 0.657. The significance of the feature is identified using distance similarity score with overall weight of the feature and feature ratio. Distance similarity score is represented as ∂_f and it is calculated as,

TABLE 2. Significant features of e-WOM dataset.



No	Customer Reviewed Features	No	Customer Reviewed Features
1	Author ID: A_ID	19	Product RAM: P_RAM
2	Title: T_ID	20	Sim Type: S_T
3	ReviewID: R_ID	21	Product Category: P_CAT
4	Content: C_L	22	Mobile Thickness: M_TH
5	Product Brand: P_BD	23	Weight of mobile phone: MOB_W
6	Customer Ratings: C_RT	24	Height of Product: P_HT
7	Product Battery life: PB_L	25	Product Type: P_T
8	Price of Product: PD_P	26	Front Camera: F_CAM
9	Feature Similarity Score: FS_S	27	Positive polarity of Review: P_POL
10	eWOM Review Type: R_T	28	Negative polarity of Review: N_POL
11	Product Display: P_D	29	Multi-Band: M_BD
12	Processor Type: P_T	30	Quick charging: Q_CH
13	Operating system: O_S	31	Mobile Finger sensor: M_FS
14	Water Proof: P_W	32	Internal storage: INT_S
15	Product Rear Camera: R_CAM	33	Weight of Positive opinion: W_p
16	Apps in built in Product: A_INB	34	Weight of Negative opinion: W_n
17	Feature Count of Review: f_c(R)	35	Audio Quality: A_Q
18	Product Sales details: P_SD	36	Total Review Count : N

$$\partial_f = \frac{\sum_{i=1}^{N} O_i \delta_i}{N} \tag{4}$$

Let's take the example of the review content as shown in Table 1, *Battery is too good.* Consider 90 reviews with 55 reviews includes feature '*Battery*'. The distance similarity score with respect to the identified feature *battery* is $(0.657 \times 55)/90 = 0.401$.

The similarity score calculated will be between [0,1]. Features with score value less than 0.3 is neglected due to less significance. Table 2. shows the significant features identified from e-WOM dataset which is used for further prediction.

3) HASH JOIN RESILIENT DISTRIBUTION

Hash Join works with in-memory computation using RDD. Duplicates in the customer reviews are removed using a hash phase followed by join phase. We hash the two relations into partitions on disk using a hash function, and later join them to get the final result. Customers are partitioned into 2 divisions K_i and $\mathrm{P_i}$ based on the review features and opinions. Then, we will match K_i with P_i partition. In Join phase, we build hash table to perform the joining. The hash table is implemented using RDD. Hashing function works in different worker nodes in parallel and duplicate entry is recognized without writing to the same bucket.

Hash Key generation:

Let $i = 1,2,3, \ldots, k$ represents the customer indices, $j = 1,2,\ 3, \ldots, n$ represents review indices and $m = 1,2,3,\ \ldots, t$ represents feature indices

- N denotes the total number of customer reviews, X denotes the significant features.
- Dataset with features $X = \{x_1, x_2, x_3, \ldots, x_m\}$
- Subset of features Cx_i^δ are selected in the feature identification stage can be, $Cx_i^\delta \subset X$.
- Let O_i be the overall weight of opinion with respect to i^{th} review.
- The $\mathrm{j^{th}}$ value of a particular subset selected by the customer is denoted by $C_j x_m^\delta$.
- The hash value denoted by h_v is taken as the L1 norm of the feature vector where,

$$h_v = \left\| O_{w_i}.C_j x_m^\delta \right\|_1 \tag{5}$$

- Hash function key is defined as:

$$hash - fn = \frac{i}{N}\left[Z_i\% \left(h_v\right)\right] \tag{6}$$

As shown in Table 3, the hash function will map the features from each review with unique customer to the integers corresponding to the index of hash table. Using join attribute function one or more features in a review are combined. Insert the features in to the hash table (R_h) with respect to the index obtained from hash function (h_v). Customer reviews without significant features are not evaluated. Here hash function helps to map each record into a hash table. Once the table is evaluated completely, the redundant featured rows that end up in the hash bucket (HB) with same values twice are removed. Multiple instances of same review from same customer is eliminated. In the proposed method, equation (6) is used to find the key for hashing and put element into the suitable index of the table. MHRDD removes the duplicate entries faster as compared to other searching methods or deduplication [33] methods.

TABLE 3. Hash-join feature attribute algorithm.

```
for each feature f in review R do
   let $h(f)$ hash on join attributes $f(b)$
      place $f$ in hash table $R_h$ in bucket keyed
      by hash value $\mathrm{h_v}$
    for each tuple $k$ do
        let $h(f)$ hash on attach attributes $f(a)$
   if $h$ indicates a nonempty bucket ($HB$) of hash
table $R_h$
     if $h$ matches any $f$ in $HB$
    join $f(a)$ and $f(b)$
place relation in Queue
```

C. PRE-LAUNCH PREDICTION WITH DIFFERENT CLASSIFIERS

The next step in our approach is to build prediction classifiers. The classifiers used are Support Vector Machine (SVM), Naïve Bayes (NB), Decision tree (DT) and XGBoost

1) SUPPORT VECTOR MACHINE (SVM)

SVM is the supervised machine learning technique. The aim of SVM is to find out the better separating hyperplane between two class training datasets. This hyperplane should be far from the dataset elements of the other class. Support vectors are the dataset elements that lie close to the classifier margin. To find the optimal plane minimize the two decision boundaries distance. For the prelaunch prediction of the product the hyperplane which divides these classes have to be determined. For each product vector r_i, hyperplane is defined as

$$w.r_i + b \geq 1 \tag{7}$$

for r_i to be in class 1.

$$w.r_i + b \leq -1 \tag{8}$$

for r_i to be in class -1.

The product in the positive 1 class is considered as successful product, [from equation (7)] and if it is in the negative 1 class [from equation (8)] then the product is considered a failure.

2) NAÏVE BAYESIAN

Naive Bayes [27] is a classification technique which works on probabilistic model. This model requires that probabilistic features are independent and are not related to one another. Naïve Bayes probabilistic method for classification involves modeling the conditional probability distribution $P(S|\ R)$, where S ranges over classes and R over product reviews. During prelaunch product prediction it is denoted by success class value as '1' and failure class value as '0'.

$$P\left(s|R\right) = \frac{P\left(R|s\right)P\left(s\right)}{P\left(R\right)} \tag{9}$$

where 's' is the class instance, 'R' is the product feature vector of size 'k', where $\mathrm{R} = (\mathrm{r_1},\ \mathrm{r_2}, \mathrm{r_3}, \ldots, \mathrm{r_k})$.

The classifier model is

$$s = argmax_s P\left(s\right) i = \prod_{i=1}^{k} P\left(r_i|s\right) \tag{10}$$

TABLE 4. Decision tree algorithm.

```
1: Select the product as the root node $X$ and $n$ features
2: Allocate the highest weight product feature from $X_1$ to $X_n$
2: Allocate the decision class for the node $X$.
3: Descendant node is created for each node of $X$.
4: By sorting the training dataset to the appropriate suitable node
   leaf.
5: If classification of the input dataset is done, then STOP.
6:   else iteration of the new leaf nodes.
```

3) DECISION TREE

Decision Tree [39] is a machine learning algorithm, where prediction of target class is done based on decision rules. These rules are generated from past data obtained. We built a decision tree for predicting products status in the market before its launch. In each stage, decision tree selects each node by calculating the highest information gain of all the product feature attributes. Here decision tree is built on using the review featured dataset based on the Iterative Dichotomiser 3 (ID3) [39] method. Table 4. shows the decision tree algorithm for the pre-launch prediction.

Figure 3 shows a sample decision tree. The root attribute element is taken as the product. Tree is constructed based on the random input dataset. The product attribute is categorized into features of the customer reviews and the customer ratings. These feature and the ratings nodes are the descendent. Depending on the features and ratings, success or failure prediction of the product occurs. Characteristics of the product features can be grouped under into good, bad and average, depending on that success or failure prediction occurs. Rating scale is taken as 0-5 and a condition is set with rating scale of greater than 3 for the successful product and less than or equal to 3 in the case of failure.

4) XGBoost ALGORITHM

XGBoost algorithm [48] is a machine learning algorithm that utilizes a Gradient boosting framework. The training and testing dataset is used for the evaluation and also for the k-fold cross-validation model. The cross-validation model is one in which, rather than one training and testing set, '*t*' sets are built, called "folds," and then t-1 folds are used to train the dataset, as well as the tth fold, is used for testing purpose. This process is repeated until the test folds are divided. The mean of the individual folds is considered as the final result. We implemented the L2 regularization of the XGBoost algorithm to obtain a generalized model [46], [47]. Each level of the progression of the XGBoost algorithm can be observed as a version of the pre-launch product prediction process.

- Decision Tree: Every product has a set of features such as battery life, cost, etc. A decision tree is comparable to a product success prediction based on the features of the product.

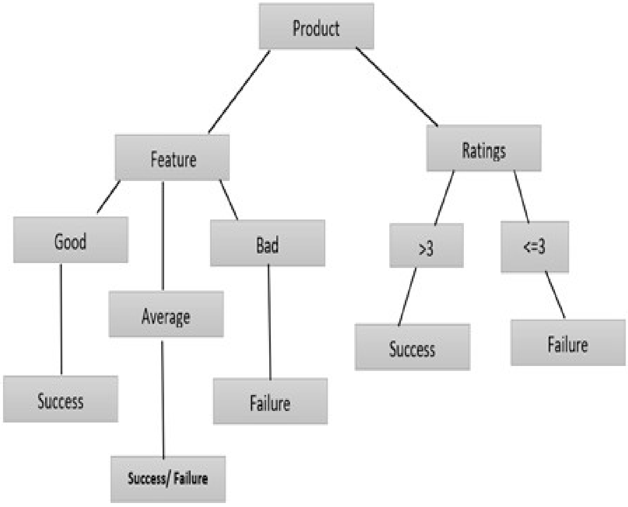


FIGURE 3. Decision tree for the concept product success or failure prediction.

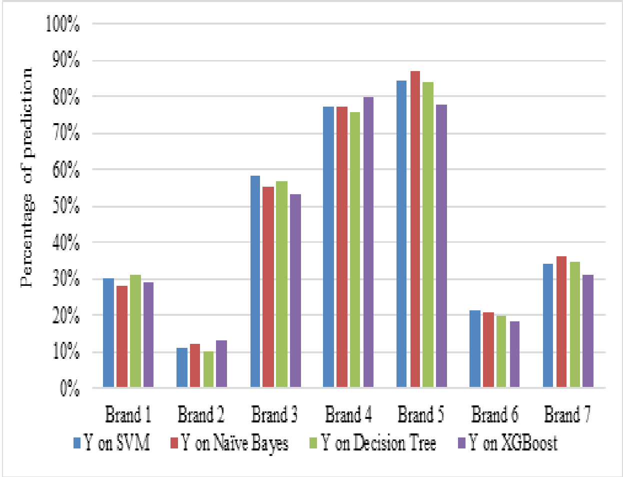


FIGURE 4. Failure prediction of product with different classifiers.

- **Bagging**: There are more than one product with a large number of review comments. Bagging collection involves combining reviews from different customers about all products for the final decision through significant feature extraction and data pre-processing and predictive analytics.
- **Random Forest**: This is a bagging algorithm with a subset of features that are picked at random.
- **Boosting**: It is an alternative method where each feature contributing to the success of the prediction modifies the evaluation measures based on features selected and prediction classifier. This 'boosts' the performance of the predictive analysis process.
- **Gradient Boosting**: This algorithm reduces the error rate, which is a special case of the gradient descent algorithm.
- **XGBoost**: XGBoost algorithm is an extreme variant of the gradient boosting method. It is an excellent aggre-

TABLE 5. Prediction of seven brands of product with different classifiers using proposed method.

X- Success Prediction Y- Failure Prediction	X on SVM	Y on SVM	X on Naive Bayes	Y on Naive Bayes	X on Decision Tree	Y on Decision Tree	X on XGBoost	Y on XGBoost
Brand 1	70%	30%	72%	28%	69%	31%	71%	29%
Brand 2	89.1%	27.9%	88%	12%	90.2%	9.8%	87%	13%
Brand 3	42%	69%	45%	55%	43.3%	56.7%	47%	53%
Brand 4	23%	77%	22.5%	77.5%	24.2%	75.8%	20%	80%
Brand 5	15.4%	75.6%	13%	87%	16%	84%	22%	78%
Brand 6	78.5%	21.5%	79%	21%	80.3%	19.7%	82%	18%
Brand 7	66%	34%	64%	36%	65.4%	34.6%	69%	31%

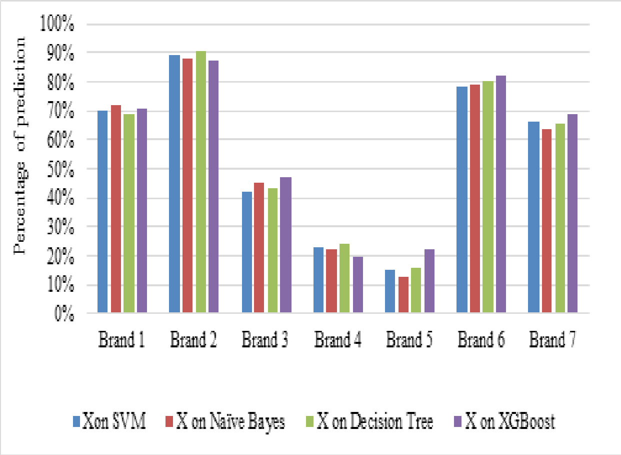


FIGURE 5. Success prediction of product with different classifiers.

gation of software and hardware optimization methods to generate better results utilizing fewer computing resources with minimum time.

IV. EXPERIMENTAL SETUP

The intended system was realized using Apache Spark framework. PySpark version 2.1.2. Amazon Web Services is used to run some components of the software system, having four Intel Xeon E5-2699V4 2.2G Hz processors with four cores and 16 GB of RAM on Spark cluster configurations. According to the scalability requirements, the software components can be configured and can run on separate servers.

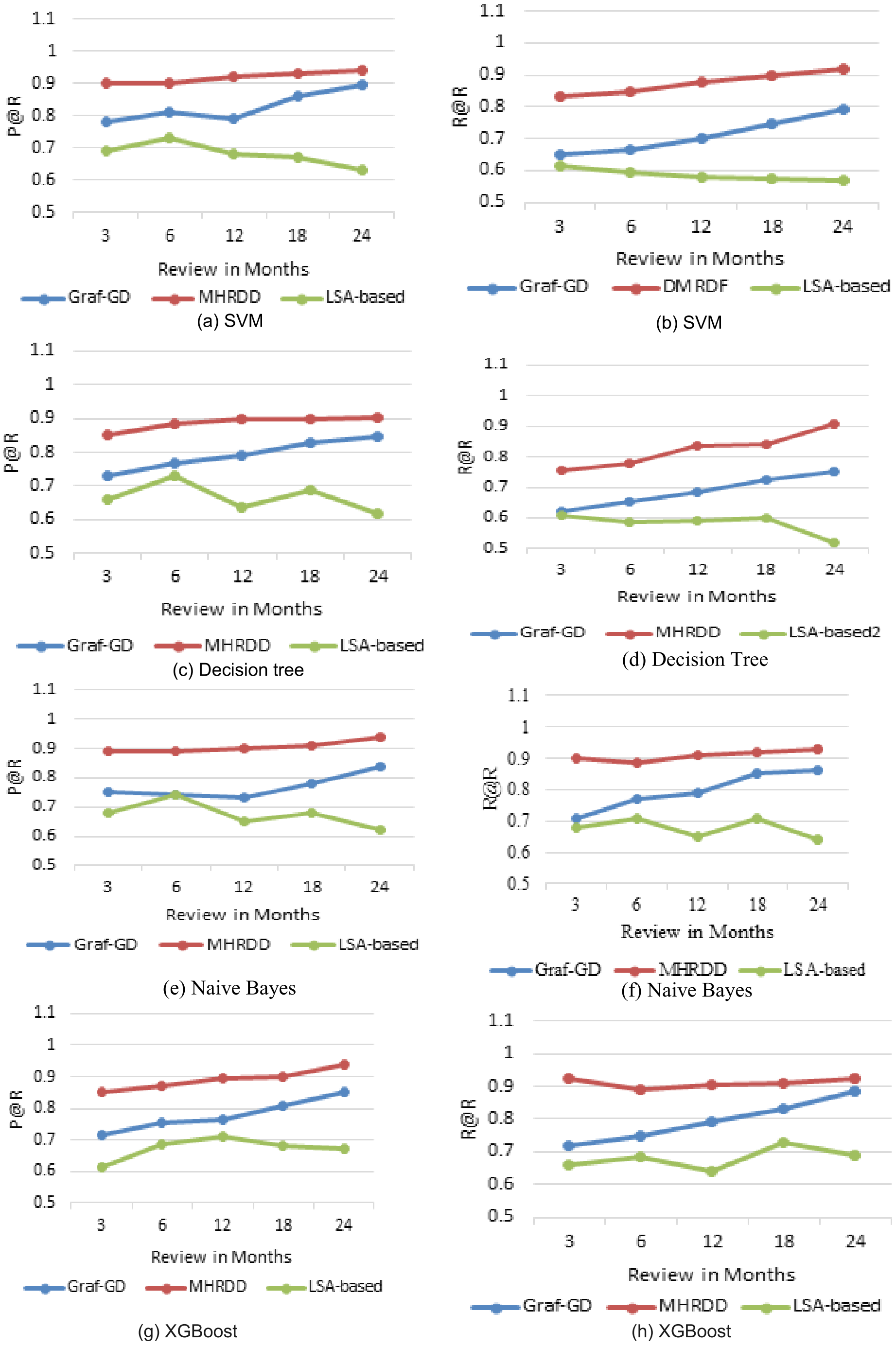
This model helps to predict the failure or success of a unique product in the market by analysing significant features from product customer reviews. A case study is conducted using customer reviews of 7 brands of mobile phones. Success or failure is the feature variable used for training and testing the dataset. For training purposes, 75 % of the dataset is used and for testing the model, the remaining 25% is used.

TABLE 6. Results of comparison of the proposed model with state-of-the-art techniques using 24 months e-WOM dataset.

Classifier	**Support Vector Machine**		
Method Used	**P@R (Precision)**	**R@R (Recall)**	**P-Accuracy% (Prediction Accuracy)**
MHRDD	0.941	0.92	97.4
LSA-based	0.63	0.567	83.2
Graf_GD	0.895	0.79	87.5
Classifier	**Naive Bayes**		
Method Used	**P@R (Precision)**	**R@R (Recall)**	**P-Accuracy% (Prediction Accuracy)**
MHRDD	0.936	0.909	96.5
LSA-based	0.62	0.52	79.8
Graf_GD	0.839	0.753	83
Classifier	**Decision Tree**		
Method Used	**P@R (Precision)**	**R@R (Recall)**	**P-Accuracy% (Prediction Accuracy)**
MHRDD	0.905	0.87	95.6
LSA-based	0.65	0.50	78.3
Graf_GD	0.849	0.763	85
Classifier	**XGBoost**		
Method Used	**P@R (Precision)**	**R@R (Recall)**	**P-Accuracy% (Prediction Accuracy)**
MHRDD	0.94	0.925	97.9
LSA-based	0.673	0.69	81
Graf_GD	0.852	0.885	89.2

V. RESULT ANALYSIS AND DISCUSSION

Prediction classifier is built and tested using SVM, Naïve Bayes, Decision tree and XGBoost algorithms. When dataset is preprocessed with MHRDD method we get good prediction



(a) SVM

(b) SVM

(c) Decision tree

(d) Decision Tree

(e) Naive Bayes

(f) Naive Bayes

(g) XGBoost

(h) XGBoost

FIGURE 6. **Precision and recall comparison of the MHRDD with state-of-the-art-methods using e-WOM dataset.**

results. Further the prediction percentage does not vary much with different classifiers. This shows that data pre-processing plays an important role in future prediction. Table 5 shows prediction of seven brands of products with different classifiers using proposed method.

Products pre-launch prediction of 7 brands of mobile phones are tested as shown in Table 5. This can be noted from Figure 4, where the failure prediction of product with Support vector machine, Naïve Bayes and Decision tree are shown. Figure 5, shows the effect of the success prediction of product with different classifiers with different number of customer reviews. Comparing with these classifiers XGBoost is having better prediction accuracy compared to other classifiers..

Figures 4 and 5 show that with different classifiers prediction do not vary much in the product pre-launch prediction using the e-WOM dataset. Brand 2 has the highest success

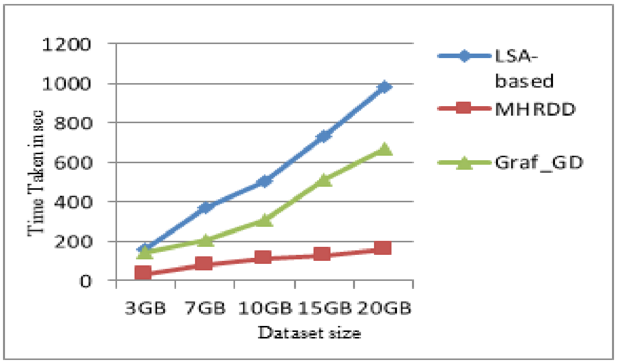


FIGURE 7. Execution time comparison of the proposed method with the state-of-the-art methods.

prediction percentage and Brand 5 has highest failure prediction percentage.

The reliability of the MHRDD method depends on precision, recall and performance accuracy [35], [38] measurement. Table 7 shows a comparison of precision, recall and accuracy measures of MHRDD, Grafting Gradient Descent and LSA-based methods with Support Vector Machine and Naïve Bayes, Decision tree classifier and XGBoost algorithm. The results shown in Table 6 are best proved using MHRDD with XGBoost classification with an accuracy of 97.9%. The MHRDD outperforms Grafting GradientDescent and LSA-based methods in P@R, R@R and P_Accuracy measures. Using proposed method, false negative (FN), true positive(TP), false positive (FP) and true negative (TN) are found out. The performance parameters such as Prediction accuracy (P-Accuracy), precision (P@R) and recall (R@R)are computed using equations (11), (12), and (13) respectively.

$$P - Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$P@R = \frac{TP}{TP + FP} \tag{12}$$

$$R@R = \frac{TP}{TP + FN} \tag{13}$$

As shown in Figure 6 (a), (b), (c), (d), (e), (f), (g) and (h) a scalability comparison of MHRDD with LSA-based and Graf_GD methods has conducted. As shown in figure 6, as the dataset size of the customer reviews increases precision and recall rate decreases for the Graf_GD and LSA-based methods. As the number of months increases the dataset size also increases, proposed MHRDD shows almost constant performance for large and small dataset. The result analysis shows that MHRDD approach outperforms the other two methods in big data analysis. MHRDD method is more scalable for different sizes of datasets compared to other methods.

Figure 7 shows the comparison of the time taken for execution of the MHRDD model with the state-of-the-art techniques. MHRDD method executes the application in lesser time when compared to Grafting gradient descent and latent semantic analysis method. Result analysis shows that the proposed model is scalable and fast.

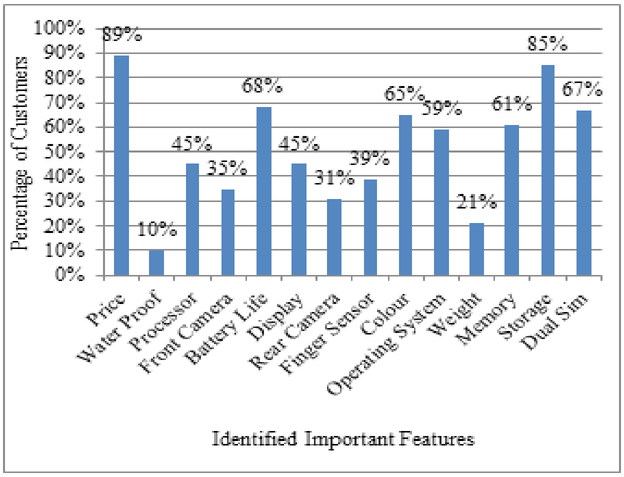


FIGURE 8. Identified important features for a successful product from e-WOM dataset.

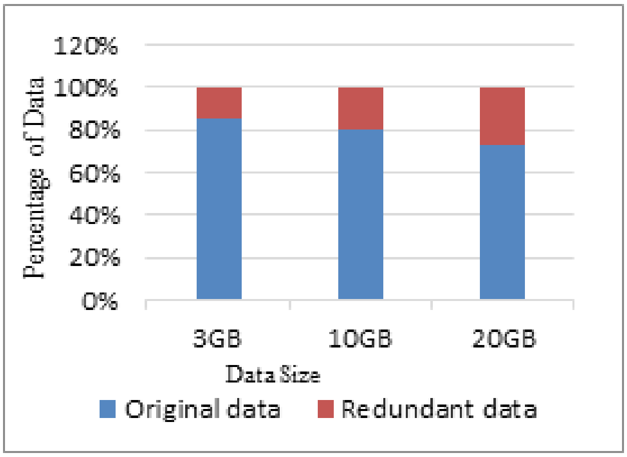


FIGURE 9. Dataset size versus percentage of redundant data eliminated.

Its performance is high, processing with large dataset. This shows the MHRDD applicability in big data analytics, whereas Graf_GD and LSA-based methods processing time is larger for large volume of dataset. Multithreaded programming using distributed computation as well as in-memory computation increases the model performance by reducing the execution time of the application. For big data analytics applications proposed system improves the execution performance.

Figure.8 illustrates important features that are required for the product to be successful. Customer feature from predictive analytics. Storage requirement has been identified by 85% of customers as significant feature and so on. With this evaluation customer requirements for a reviews, ratings and sales details of 7 brands of mobile phones are identified and evaluated with MHRDD using SVM, Naïve Bayes, Decision tree and XGBoost classifiers. The graph shows the significant features identified by the model against the percentage of

TABLE 7. Comparison of the proposed work with state-of-the-art techniques.

Paper	Method	Dataset Cleaning	Dataset Type	Distributed Data Processing model	Feature Extraction/ Application	Product Pre-Launch Prediction
Chen et al. (2020) [57]	Tight Center Loss function	Cosine similarity	Unstructured	No	Yes / Image feature extraction	No
Gutierrez et al. (2014) [31]	Linear Predictor Model	Not specified	Semi-structured	No	Yes/ Feature extraction from large dataset	No
Gutierrez et al. (2014) [54]	Random forest	Random forest	Unstructured	No	Yes/ Tweet Mining	No
Liu et al. (2012) [43]	Latent Semantic Analysis	Not Specified	Unstructured	No	No/ Movie review prediction	No
Perkins et al. (2003) [42]	Grafting Gradient Descent	Not Specified	Structured	No	Yes/ Feature Extraction	No
Manohara et al. (2017) [58]	Support Vector Machine and Logistic Regression [58]	Not Specified	Structured	Yes	No/ Predictive analytics on financial loan	No
Criminisi et al. (2012) [56]	Random decision forest model [56]	Not Specified	Semi-structured	No	No/ Prediction Framework	No
Our Proposed work MHRDD	MHRDD with XGBOOST, SVM, Naive Bayes and Decision tree	Distance similarity Measure with Multithreaded Hash Join Resilient Distribution	Unstructured	Yes	Yes/ Predictive analytics from Customer reviews	Yes

customers whose reviews are analyzed. As shown in Figure 8, larger number of customers identified product price as one the significant feature from predictive analytics. Storage requirement has been identified by 85% of customers as significant feature and so on. With this evaluation customer requirements for a product can be analyzed in a better manner, thus can improve the design of the product for better product quality and for product sustainability in the market. Significant features identification of the product plays an important role in predictive analytics.

Figure 9 shows, data size versus percentage of redundant data removed during pre-processing in three datasets by the proposed method. In the first 3GB dataset, 14% of redundant data has been removed. In the 10GB and 20GB datasets 20% and 27% of redundant data has been identified and removed. Removal of more redundant data increases the accuracy of the prediction model.

As shown in Table 7, a comparison of the proposed work with state-of-the-art techniques is detailed. The novelty of the proposed method with other approaches is shown here.

Table 7 compares the proposed MHRDD with the state-of-the-art techniques, dataset cleaning, method, type of the dataset, distributed data processing and application used for the implementation of the model.

VI. CONCLUSION

With fast technological developments, new products with innovative features are launched into the market. In this work, a novel big data processing has been implemented that predicts product success before its launch in the market. This helps the industrialists to launch and sustain a successful product in the market and also the consumers to get a good quality product. Customer product reviews, rating and product sales details are taken as the training dataset and for testing. A distance similarity scores along with a multithreaded hash-join method with a resilient distributed dataset for unwanted data removal and significant feature selection has been done. Along with this model, classification algorithms are implemented for prediction. We have given a priority weightage to product-based features on the opinion of the customer reviews. The model is fault-tolerant as it uses a resilient distributed dataset. The prediction accuracy, precision and recall of the MHRDD method outperforms the Grafting Gradient Descent and LSA-based methods.

Compared to the state-of-the-art techniques the prediction accuracy of the proposed method increases by 11% using significant feature identification and eliminating redundancy from dataset. Results show that the proposed MHRDD model performance is excellent as well as time taken for processing the application is less compared to the state of the art techniques. 27% of redundant unwanted customer reviews and ratings have been removed from the original raw dataset, which increases the model's prediction accuracy. Resilient dataset distribution property on Multithreaded Hash Join method has a long lineage; hence the aforementioned can achieve fault-tolerance. The MHRDD model is fast because of the distributed in-memory computation approach. The proposed approach can be extended to other product feature identification of big data predictive analytics. As future work, the model may be improved to make real-time streaming forecasts through a centralized API that explores customer suggestions, credentials, ratings and surveys from different reliable online sites.

REFERENCES

[1] S. K. Chauhan, A. Goel, P. Goel, A. Chauhanm, and M. K. Gurve, ''Research on product review analysis and spam review detection,'' in *Proc. Int. Conf. Signal Process. Integr. Networks*, Feb. 2017, pp. 390–393.

[2] S. Gopalani and R. Arora, ''Comparing apache spark and map reduce with performance analysis using K-Means,'' *Int. J. Comput. Appl.*, vol. 113, no. 1, pp. 8–11, Mar. 2015.

[3] S. Jeon, B. Hong, J. Kwon, Y. S. Kwak, and S. I. Song, ''Redundant data removal technique for efficient big data search processing,'' *Int. J. Softw. Eng. Appl*, vol. 7, no. 4, pp. 427–436, 2013.

[4] C. Dellarocas, X. Zhang, and N. F. Awad, ''Exploring the value of online product reviews in forecasting sales: The case of motion pictures,'' *J. Interact. Marketing*, vol. 21, no. 4, pp. 23–45, Jan. 2007.

[5] *Dataset*. Accessed: Jan. 18, 2020. [Online]. Available: https://www.kaggle.com/PromptCloudHQ/flipkart-products

[6] *Dataset*. Accessed: Feb. 6, 2020. [Online]. Available: https://www.kaggle.com/nehathakur28dec/amazon-mobile

[7] J. S. Lee and E. S. Lee, ''Exploring the usefulness of a decision tree in predicting people's locations,'' *Procedia-Social Behav. Sci.*, vol. 140, pp. 447–451, Aug. 2014, doi: 10.1016/j.sbspro.2014.04.451.

[8] G. Cui, H.-K. Lui, and X. Guo, ''The effect of online consumer reviews on new product sales,'' *Int. J. Electron. Commerce*, vol. 17, no. 1, pp. 39–58, Oct. 2012.

[9] P. Racherla, M. Mandviwalla, and D. J. Connolly, ''Factors affecting consumers' trust in online product reviews,'' *J. Consum. Behaviour*, vol. 11, no. 2, pp. 94–104, Mar. 2012.

[10] N. Jindal and B. Liu, ''Review spam detection,'' in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 1189–1190.

[11] T. Kolajo, O. Daramola, and A. Adebiyi, ''Big data stream analysis: A systematic literature review,'' *J. Big Data*, vol. 6, no. 1, p. 47, Dec. 2019, doi: 10.1186/s40537-019-0210-7.

[12] G. Liu, T. T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, and W. Chen, ''Repeat buyer prediction for E-Commerce,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 155–164.

[13] N. Jindal and B. Liu, ''Identifying comparative sentences in text documents,'' in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2006, pp. 244–251.

[14] A.-M. Popescu and O. Etzioni, ''Extracting product features and opinions from reviews,'' in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*, 2005, pp. 9–28.

[15] J. Wang, G. Qi, N. Sebe, and C. Aggarwal, ''Guest editorial: Big MEDIA DATA: Understanding, search, and mining,'' *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 82–83, Dec. 2017, doi: 10.1109/TBDATA.2017.2665161.

[16] Y. Demchenko, Z. Zhao, P. Grosso, A. Wibisono, and C. de Laat, ''Addressing big data challenges for scientific data infrastructure,'' in *Proc. 4th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2012, pp. 614–617.

[17] S. Vengadeswaran and S. R. Balasundaram, ''An optimal data placement strategy for improving system performance of massive data applications using graph clustering,'' *Int. J. Ambient Comput. Intell.*, vol. 9, no. 3, pp. 15–30, Jul. 2018, doi: 10.4018/IJACI.2018070102.

[18] R. Y. K. Lau, S. Y. Liao, R. C.-W. Kwok, K. Xu, Y. Xia, and Y. Li, ''Text mining and probabilistic language modeling for online review spam detection,'' *ACM Trans. Manage. Inf. Syst.*, vol. 2, no. 4, pp. 1–30, Dec. 2011.

[19] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, ''Survey of review spam detection using machine learning techniques,'' *J. Big Data*, vol. 2, no. 1, p. 23, Dec. 2015.

[20] A. B. Burmester, J. U. Becker, H. J. van Heerde, and M. Clement, ''The impact of pre- and post-launch publicity and advertising on new product sales,'' *Int. J. Res. Marketing*, vol. 32, no. 4, pp. 408–417, Dec. 2015, doi: 10.1016/j.ijresmar.2015.05.005.

[21] S. Singh and N. Singh, ''Big data analytics,'' in *Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT)*, 2012, pp. 1–4.

[22] K. Bakshi, ''Considerations for big data: Architecture and approach,'' in *Proc. IEEE Aerosp. Conf.*, Mar. 2012, pp. 1–7, doi: 10.1109/aero.2012.6187357.

[23] R. V. Bandakkanavar, M. Ramesh, and H. Geeta, ''A survey on detection of reviews using sentiment classification of methods,'' *IJRITCC*, vol. 2, no. 2, pp. 310–314, 2014.

[24] P. K. Chintagunta, S. Gopinath, and S. Venkataraman, ''The effects of online user reviews on movie box office performance: Accounting for sequential rollout and aggregation across local markets,'' *Marketing Sci.*, vol. 29, no. 5, pp. 944–957, Sep. 2010.

[25] H. Zhang, Z. Wang, S. Chen, and C. Guo, ''Product recommendation in online social networking communities: An empirical study of antecedents and a mediator,'' *Inf. Manage.*, vol. 56, no. 2, pp. 185–195, Mar. 2019.

[26] L. Gu and H. Li, ''Memory or time: Performance evaluation for iterative operation on Hadoop and spark,'' in *Proc. IEEE 10th Int. Conf. High Perform. Comput. Commun. IEEE Int. Conf. Embedded Ubiquitous Comput.*, Nov. 2013, pp. 721–727, doi: 10.1109/hpcc.and.euc.2013.106.

[27] K. M. Leung, ''Naive Bayesian classifier,'' Finance Risk Eng., Polytech. Univ. Dept. Comput. Sci., Tech. Rep., Nov. 2007.

[28] S. Hong and H. Kim, ''An analytical model for a GPU architecture with memory-level and thread-level parallelism awareness,'' *ACM SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 152–163, Jun. 2009, doi: 10.1145/1555815.1555775.

[29] S. Yu, X. Li, X. Zhao, Z. Zhang, F. Wu, J. Wang, Y. Zhuang, and X. Li, ''A bilinear ranking SVM for knowledge based relation prediction and classification,'' *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 588–600, Dec. 2019, doi: 10.1109/TBDATA.2018.2843766.

[30] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, ''Spotting opinion spammers using behavioral footprints,'' in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Chicago, IL, USA, 2013, pp. 632–640.

[31] C. E. Gutierrez, P. M. R. Alsharif, M. Khosravy, P. K. Yamashita, P. H. Miyagi, and R. Villa, ''Main large data set features detection by a linear predictor model,'' in *Proc. AIP Conf.*, vol. 1618, 2014, pp. 733–737, doi: 10.1063/1.4897836.

[32] S. Xie, G. Wang, S. Lin, and P. S. Yu, ''Review spam detection via temporal pattern discovery,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Beijing, China, 2012, pp. 823–831.

[33] G. Wang, S. Xie, B. Liu, and P. S. Yu, ''Identify online store review spammers via social review graph,'' *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 1–21, Sep. 2012.

[34] N. Jindal and B. Liu, "Review spam detection," in *Proc. 16th Int. Conf. World Wide Web*, Lyon, France, 2007, pp. 1189–1190.

[35] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," EECS Dept., UC Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2011-82, 2011.

[36] S. Shayaa, N. I. Jaafar, S. Bahri, A. Sulaiman, P. Seuk Wai, Y. Wai Chung, A. Z. Piprani, and M. A. Al-Garadi, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37807–37827, 2018, doi: 10.1109/ACCESS.2018.2851311.

[37] F. Mansur, V. Patel, and M. Patel, "A review on recommender systems," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Coimbatore, India, Mar. 2017, pp. 1–6, doi: 10.1109/ICIIECS.2017.8276182.

[38] N. Bharill, A. Tiwari, and A. Malviya, "Fuzzy based scalable clustering algorithms for handling big data using apache spark," *IEEE Trans. Big Data*, vol. 2, no. 4, pp. 339–352, Dec. 2016, doi: 10.1109/TBDATA.2016.2622288.

[39] C. Jin, L. De-lin, and M. Fen-xiang, "An improved ID3 decision tree algorithm," in *Proc. 4th Int. Conf. Comput. Sci. Edu.*, Jul. 2009, pp. 127–130.

[40] K. Yu, X. Wu, W. Ding, and J. Pei, "Scalable and accurate online feature selection for big data," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 2, pp. 1–39, Dec. 2016, doi: 10.1145/2976744.

[41] D. Lin, "Dependency-based evaluation of MINIPAR," in *Proc. Workshop Eval. Parsing Syst.*, Granada, Spain, 1998, pp. 317–329.

[42] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *J. Mach. Learn.*, vol. 3, pp. 1333–1356, Mar. 2003.

[43] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, G.-C. Lu, and E. Jou, "Movie rating and review summarization in mobile environment," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 42, no. 3, pp. 397–407, May 2012.

[44] X. Meng, L. Nie, and J. Song, "Big data-based prediction of terrorist attacks," *Comput. Electr. Eng.*, vol. 77, pp. 120–127, Jul. 2019, doi: 10.1016/j.compeleceng.2019.05.013.

[45] D. Basaran, E. Ntoutsi, and A. Zimek, "Redundancies in data and their effect on the evaluation of recommendation systems: A case study on the Amazon reviews datasets," in *Proc. 17th SIAM Int. Conf. Data Mining (SDM)*, 2017, pp. 390–398, doi: 10.1137/1.9781611974973.44.

[46] S. V. Murty and R. K. Kumar, "Accurate liver disease prediction with extreme gradient boosting," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 2288–2295, 2019, doi: 10.35940/ijeat.F8684.088619.

[47] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *Int. J. Control Theory Appl.*, vol. 9, pp. 651–662, Mar. 2017.

[48] M. Chen, Q. Liu, S. Chen, Y. Liu, C.-H. Zhang, and R. Liu, "XGBoost-based algorithm interpretation and application on post-fault transient stability status prediction of power system," *IEEE Access*, vol. 7, pp. 13149–13158, 2019, doi: 10.1109/ACCESS.2019.2893448.

[49] L. Bertossi, S. Kolahi, and L. V. S. Lakshmanan, "Data cleaning and query answering with matching dependencies and matching functions," in *Proc. 14th Int. Conf. Database Theory (ICDT)*, 2011, pp. 268–279, doi: 10.1145/1938551.1938585.

[50] G. V. Dhivyabharathi and S. Kumaresan, "A survey on duplicate record detection in real world data," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Jan. 2016, pp. 1–5, doi: 10.1109/ICACCS.2016.7586397.

[51] L. Bertossi, S. Kolahi, and L. V. Lakshmanan, "Data cleaning and query answering with matching dependencies and matching functions," in *Proc. ACM Int. Conf.*, 2011, pp. 268–279, doi: 10.1145/1938551.1938585.

[52] W. Fan, "Dependencies revisited for improving data quality," in *Proc. 27th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. (PODS)*, 2008, pp. 159–170, doi: 10.1145/1376916.1376940.

[53] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining Int. Conf. Knowl. Discovery Data Mining*, 2003, p. 39, doi: 10.1145/956755.956759.

[54] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, Dec. 2000, doi: 10.1145/1317331.1317341.

[55] C. E. Gutierrez, M. R. Alsharif, K. Yamashita, and M. Khosravy, "A tweets mining approach to detection of critical events characteristics using random forest," *Int. J. Next-Gener. Comput.*, vol. 5, no. 2, pp. 167–176, 2014.

[56] A. Criminisi, J. Shottonand, E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Found. Trends R Comput. Graph. Vis.*, vol. 2, no. 3, pp. 81–227, 2012.

[57] Y. Chen, C. Wu, and Y. Wang, "T-center: A novel feature extraction approach towards large-scale iris recognition," *IEEE Access*, vol. 8, pp. 32365–32375, 2020, doi: 10.1109/ACCESS.2020.2973433.

[58] G. Attigeri, M. M. Manohara Pai, and R. M. Pai, "Analysis of feature selection and extraction algorithm for loan data: A big data approach," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 2147–2151.

SANDHYA NARAYANAN received the M.Tech. degree in computational engineering and networking from Amrita Viswa Vidya Peetham, in 2011. She is a Researcher with the Information Technology Division, School of Engineering, Cochin University of Science and Technology (CUSAT). She has more than seven research publications in reputed journals and conferences. Her research interests include big data analytics, machine learning, and artificial intelligence.

PHILIP SAMUEL received the M.Tech. degree in computer and information science from the Cochin University of Science and Technology (CUSAT) and the Ph.D. degree in computer science and engineering from the IIT Kharagpur. He has more than 20 years of experience in teaching and research as a Faculty Member of the CUSAT, where he is currently a Professor with the Department of Computer Science. He has published more than 60 research papers in international conferences and journals. His research interests include big data analytics, distributed computing, automated software engineering, and artificial intelligence.

MARIAMMA CHACKO was born Changanacherry, India, in 1961. She received the bachelor's degree in electrical engineering from the University of Kerala, in 1985, and the master's degree in electronics and the Ph.D. degree in computer engineering from the Cochin University of Science and Technology (CUSAT), in 1987 and 2012, respectively. She has been working as a Faculty Member of the Department of Ship Technology, CUSAT, since 1990, where she is currently a Professor with the Department of Ship Technology. She has more than 25 research publications to her credit. Her research interests include the validation and optimization of embedded software, power quality in ships electrical systems, and the sensor less control of BLDC motors.