Received: May 24, 2023. Revised: June 29, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Progressive Sampling and RadeMacher Average for an Effective Frequent Pattern Mining in Big Data Environment

Yathish Aradhya Bandur Chandrashekariah[1]* Dinesha Hagare Annappaiah[2]

[1]*Department of Computer Science and Engineering, VTU-RRC, Belagavi, India*
[2]*Department of Computer Science and Engineering, Sridevi Institute of Technology, Tumkur, India*
* Corresponding author's Email: docs.kit@gmail.com

Abstract: Big data refers to the large amount of information that is collected from different areas and shared on the internet. However, this development has led to difficulties in using frequent itemset mining applications. To overcome the issue of frequent data mining, this research has introduced an empirical sampling algorithm using RadeMacher average (ESA-RMA). When considering the size of the initial sample and scheduling the samples, the ESA utilizes the RadeMacher average to bound the samples. Initially, the data is obtained from the dataset of the human activity recognition (HAR) and real time datasets from smartphone gyroscope and accelerometer, then obtained data is pre-processed using the data normalization technique. Then, the ESA is used to select the labelled data and RMA is used to bound the samples. This bounding process defines the upper limit of the input data which helps in the effective mining of frequent item sets. Thus, the data with redundant items are mined out using the proposed ESA-RMA method. The experimental results show that the proposed ESA-RMA has taken a minimum run time of 212 ms for data obtained from smartphone accelerometer which is comparatively lower than the existing Scalable Simple Random Sampling (ScaSRS) with processing time of 362 ms. Similarly, for HAR dataset, the proposed method took processing time of 5.43 s whereas the existing vertical frequent time interval–related pattern (VertTIRP) mining approach took processing time of 7.82 s.

Keywords: Big data, Empirical sampling algorithm, Frequent item set mining, Rademacher average, Human activity recognition.

1. Introduction

Database systems play an important role in storing big data with real-time applications. Moreover, the data varies from structured to unstructured data obtained from various applications such as system transactions, the world wide web, and so on [1,2]. Frequent item set mining (FIM) is known as one of the significant processes involved in mining big data which attracts more researchers to work on it. Mining out the frequent item set from the stream of data is one of the major issues involved in the process of FIM [3]. The process involved in mining the frequent item aims to detect the item set where the occurrence frequency exceeds the present frequency of massive databases like big data. The process involved in mining the redundant item offers different types of tasks regarding correlation analysis, local periodicity and plotting the fragments [4]. The detection of frequent item sets involves various types of resources which help in mining the frequent items and diminishes the burden of the process [5]. Mining the frequent dataset requires multiple passes through a database which helps in the progress of detecting frequent items in static or dynamic databases [6, 7]. However, complexities occurred during the evaluation of time in mining the data and this can be overwhelmed using the sampling technique.

The sampling technique can select the labelled data from the group of unlabelled data [8, 9]. Moreover, the techniques based on sampling consider the type of data and helps to minimize the response time [10]. However, the limitation occurs in the sampling technique in form of diminished accuracy value. To improve the efficiency and accuracy of frequent item set mining, a progressive sampling technique is employed, which leads to faster convergence and better results [11, 12]. Additionally, the progressive sampling technique determines the size of the entire database in a randomized manner [13]. The data gets segregated into two phases such as labeled data and pseudo-labeled data while mining the frequent data. Moreover, progressive sampling has the efficiency to figure out the data with reliable pseudo labels with their data indices [14, 15]. The existing works have problems related to execution time and error value while mining frequent data. To overcome the aforementioned problems, this research introduced a progressive sampling utilizing RadeMacher average value to evaluate the value of frequent item sets and mine it out.

The major contributions of this research are mentioned as follows:

1. This research introduced an empirical sampling algorithm using RadeMacher average to mine the frequent item set. Moreover, the empirical sampling algorithm is used to select the labeled data and the RadeMacher algorithm is used to bound the samples.
2. The effective RadeMacher average is computed by relaxing the bounded values of the sequential pattern using the empirical sampling algorithm.

The remaining the research paper is organized in the following way: Section 2 describes the related works based on sampling techniques and the proposed method is described in section 3. The section 4 describes the results and analysis and finally, the overall conclusion of this research is presented in section 5.

2. Related works

P.P. Jashma Suresh [16] introduced a hybrid switching framework to mine the frequent Itemset. The hybrid switching framework is the combination of NegNodesets combined with the list-based structure. The NegNodesets were comprised with bitmaps which creates concise pattern of Itemset and the list-based structure perform intersection operation to create list of frequent Itemset. Moreover, the transactional merging concept was used in hybrid approach to reduce the runtime by combining several transactions in a single Itemset. However, the operations performed based on listed operations consumes more memory.

Sacha Servan-Schreiber [17] have introduced Progressive Sequence mining with convergence guarantees (ProSecCo) algorithm for the progressive mining of frequent data from large datasets. The introduced method utilized VC-dimensions with strong probabilistic guarantees and deliver the intermediate results of frequent sequences with high-quality data. The ProSecCo provides a quality collection of sequences in a minimal time when compared with the non-progressive algorithms. At low-frequency thresholds, the explosion of pattern occurs which may affect the overall performance of ProSecCo.

Diego Santoro [18] have introduced a sampling-based algorithm to mine the frequent items from huge databases. The introduced sampling algorithm utilized the ideology of VC-dimension which helps in approximating the frequent sequential pattern. Moreover, the efficiency of the introduced sampling algorithm was evaluated for the small-level to large-level databases. The introduced sampling algorithm utilized upper bound maximum deviation to obtain better results at the time of mining massive data.

Kheyreddine Djouzi [19] have introduced an effective sampling methodology based on scalable simple random sampling (ScaSRS) and subsampled double bootstrap (SDB) to select smaller subsets during sampling. In the introduced sampling technique, the SDB method evaluates the variance and ScaSRS could scale up the entire process that aids in better accuracy of the introduced method. The introduced sampling method verifies the selected instance and makes use of minimum instances to attain better results. However, when the number of instances gets increased the computational time also gets increased.

Mingtao Lei [20] have introduced a transaction database graph model to mine out the top- k sequential patterns from the big data. Each path in the graph was comprised of multi-sequential transactions to find the sequential patterns with qualified guarantees. Initially, the length of the transaction path and was determined to collect the sequence of transactions. By using the collected transactional sequences, the proper top-k pattern is obtained. Moreover, the model utilized an unbiased estimator which helps to obtain provable guarantees with minimal error rate. However, the huge size of transaction data leads to time complexity and space cost.

N Yamuna Devi [21] have introduced a parallel direct vertical map reduce (PDVMR) programming approach to mine the frequent item from UCI machine learning repository dataset. The suggested approach is comprised with two stages such as mapping and reduction. The mapping was performed using the parallel direct vertical method and the execution of the suggested approach was based on the number of clusters. However, the scalability of the suggested approach was minimized when the number of nodes were increased.

Natalia Mordvanyuk [22] have introduced a vertical frequent time interval–related pattern (VertTIRP) mining approach to combine and mine the effective patterns. VertTIRP utilize temporal transitivity properties along with a pairing strategy to improve the speed of mining process. Moreover, temporal relations were used to remove the ambiguities based on epsilon approach. The suggested approach implicit on a transitive property which minimize the time and enhance the mining efficiency. However, VertTIRP approach was incapable to mine the closely related patterns.

The related works discussed in this section have the major drawback related to time complexities and error rates. By considering this, the proposed research introduced an empirical progressive sampling technique to overcome the fore-mentioned issues.

3. Preliminaries

This research mainly focused on gathering the top k by mining the sample from HAR and real time datasets from smartphone gyroscope and accelerometer. In the progressive sampling technique, I is considered as the set of items with arbitrary order A_o. The transaction is a subsection of I is comprised of transactional datasets and the item set is assigned as A which is contained in the transaction set T_s. The frequency of the item set in the dataset D is represented in Eq. (1),

$$f_D(A) = |T_D(A)|/|D| \tag{1}$$

Where f_D are the normal frequency value and the item sets with maximum frequency are denoted as $f_D^{(k)}$, the set of $Top\ k$ is denoted in Eq. (2) as follows:

$$Top\ k(D, I, k) = FI(D, I, f_D^{(k)}) \tag{2}$$

Where the maximum frequency of the dataset is represented as $f_D^{(k)}$.

3.1 Progressive sampling using RadeMacher averages

After collecting the top k samples from the dataset, The RadeMacher average is evaluated to minimize the runtime and the computational time. The basic process involved in the iterative progressive sampling process is represented as follows:

(i) In an iteration i, the random samples with pre-defined sizes are created based on randomized datasets

(ii) The stopping condition should be verified and extracted from the random sample S_i.

(iii) When the stopping condition gets satisfied, the approximates of the collected values are returned, otherwise improve i and return to first step.

This algorithm is designed to be bounded by RadeMacher averages and a criterion to choose the initial sample size thus expected to run on for fewer iterations when its predecessor methodologies are taken into consideration. The specified big data set is used to identify range spaces with less value for RadeMacher penalty. RadeMacher average is then computed for identified range space which happens to be the tight bound on the input to each subsequent iteration of the progressive sampling algorithm. The ε approximation is evaluated in a provided space range and the initial size of the sample is evaluated using the statistical divergence method. The best solution for sampling huge data, which is a non-polynomial time problem, can be produced by employing a progressive sampling technique with data bounds like the RadeMacher average. The progressive sampling algorithm is predicted to have an order polynomial runtime which is presented in Eq. (3) as follows:

$$\varepsilon = approximation + n * \\ computation\ time\ of\ learning\ algorithm \tag{3}$$

where the number of iterations is denoted as n and ε are known as the approximation value.

The RadeMacher approach utilized in the progressive sampling algorithm performs individual scanning for the datasets to detect the sampling. The algorithmic approach of the progressive sampling algorithm is mentioned as follows:

Input: HAR data and real time datasets from smartphone gyroscope and accelerometer

Output: Frequent data

1. Independent distribution of Big data was utilized in detecting the range of spaces
2. Evaluate RadeMacher average for range of space values where R ⊆ d which is represented in Eq. (4) as follows:

$$\text{Arg}\min\{P_r\{Sup_{h\in H}|\varepsilon_P(h) - \varepsilon_n(h)| \geq \varepsilon\} \leq \delta\} \tag{4}$$

Where R is the set of range of any domain d,the size of the sample is denoted as s, δ is represented as

Received: May 24, 2023. Revised: June 29, 2023.

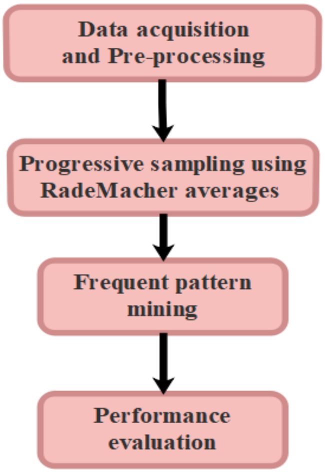
Data acquisition and Pre-processing
Progressive sampling using RadeMacher averages
Frequent pattern mining
Performance evaluation

Figure. 1 The overall process involved in mining the frequent pattern from big data

RadeMacher constant and the approximation value is represented as ε.

3. Evaluate the statistic optimal size based on divergence and it is represented as $\mathrm{S}_{\mathrm{optimal}}$.
4. Then, evaluate the sampling schedule which is presented in the form of geometric scheduled sampling shown in Eq. (5) as follows:

$$S_{schedule} = \{S_{optimal,}S_1, S_2, \dots, S_n\} \quad (5)$$

Where the sampled scheduling is denoted as $S_{schedule}$.

5. During the time of the first iteration, the optimal scheduling method $\mathrm{S}_{\mathrm{optimal}}$ is selected to evaluate and optimize the unstructured data.

These steps from 1 to 5 are evaluated for a new set of samples till the sample set achieved its progression or when the size of the sample set exceeds the bound of RadeMacher average.

6. The final step is based on training the set of samples based on their size for a selected set of ranges to obtain the set of frequent samples in the HAR and real time datasets from smartphone gyroscope and accelerometer.

3.2 Frequent pattern mining using RadeMacher model

Frequent pattern mining is a significant method to figure out the frequently used data/patterns from real-time databases. Moreover, this research addressed the problems related to time complexities and error rates. This research introduced a novel approach to mining out the redundant data using the proposed RadeMacher method for progressive sampling. Initially, the data is obtained from a series of real and synthetic databases and pre-processing is performed to filter out irrelevant information. The pre-processed data enters the phase of RadeMacher model which performs progressive sampling to mine out the frequent information with minimum convergence. The overall process involved in mining the frequent pattern is represented in Fig. 1 as follows:

3.3 Data acquisition and pre-processing

This subsection provides the datasets utilized in the proposed frequent pattern mining model and the data is obtained from both real and synthetic data. In this research, the data is obtained from human activity recognition (HAR) dataset [23] and the real time data obtained from accelerometer and gyroscope [24] is used to evaluate the proposed method. The HAR dataset is built from the records obtained from 30 subjects of daily activities performed by humans. Secondly, two real-time datasets from smartphone accelerometers and gyroscopes are used to evaluate the proposed method.

- *Smartphone accelerometer:* It contains the data obtained from the accelerometer of smartphones and the related activities of humans. This database is comprised of 11,762,265 samples.
- *Smartphone gyroscope:* This dataset is comprised of x, y, z coordinates which are captured from the gyroscope of the smartphones. It consists of 12,063,000 components after removing uncategorized components.

The obtained raw data is pre-processed using the data normalization technique which converts the values present in the dataset into a common scale of input features.

3.4 Progressive sampling

After computing the values of RadeMacher values, progressive sampling is employed in this research. Progressive sampling is defined as the method to choose the instance to build a training set of samples to attain a better convergence rate. At the initial stage, the method initiates with minimal instances based on the contexts and criteria. The value of $\mathrm{S}_{\mathrm{optimal}}$ is the sampling technique to train the progressive sampling algorithm. The samples with minimum training set will obtain better accuracy than $\mathrm{S}_{\mathrm{optimal}}$ so, Risk Minimization technique is employed in enhancing the efficiency of $\mathrm{S}_{\mathrm{optimal}}$ state to achieve better accuracy.

3.4.1. Empirical progressive sampling approach

The proposed technique presented in this research evaluates the RadeMacher average for the provided dataset and the RadeMacher average computes the upper bound size of the input data and confirms the presence of the model within the probably approximate correct framework (PAC). The RadeMacher average is highly capable to assess the large range of sample sets which is based on the convergence rate. When the probable convergence is not obtained, then it offers a tight bound and leads to sampling complexity. The size of the statistic optimal sample set diminished the useless iterative values and provides an effective sampling. The algorithmic approach to identify the statistic optimal size is provided in below mentioned pseudo code.

```
Input: The range space $R \in D$
Output: $(S_i, Q_i)$

1. Apply a random sampling technique to select the sample $S_i$ which is denoted in Eq. (6) as follows:

   $S_i = \{S_i | i = 1,2,3,4 \ldots\ldots N_i = size\ of\ (\varepsilon)\}$ (6)

   Where approximation value is defined as $\varepsilon$
2. For every $(S_i \in R)$, calculate r=random num (0.0.0.1) in every individual sample reading $S_i$.
Update the respective statistical value of $R$. If $[r < (S_i, N_i)]$ then update the respective statistic value of $S_i$.
3. Evaluate the value of $Q_i$ for every individual value of $S_i$ to obtain the output as $S_i, Q_i$
4. Plot the value of $(S_i, Q_i)$ and construct a curve, then use the linear regression technique to cover the probable outcome in the range space.
5. The mid-point value of the regression line is considered as $S_{optimal}$ that is considered the initial size of the sample in the progressive sampling algorithm. The proposed technique minimized the $Q_i$ values and significantly affects the overall runtime.
```

3.4.2. Map reduce

After the process of progressive sampling, the map-reduce technique is utilized which filters out the unrecognized samples which are present among the mapped samples. Reduction is made between each sample to eliminate the unstructured samples. The MapReduce algorithm is constructed based on two functions such as a map and reduce where the input is obtained from the key-value pairs which is denoted as (k, v). The map function obtains a pair of inputs in a single iteration and provides a multiset of pairs i.e. $\{(k_1, v_1), (k_2, v_2), \ldots, (k_n, v_n)\ \}$. The multiset union is denoted as U which is comprised of the multisets of map function when it is applied to every pair of inputs. The set U is partitioned into $U_{\bar{k}}$, where the specified key is denoted as $\bar{k}$ and the $U_{\bar{k}}$ is comprised of a pair of values denoted as $(\bar{k}, v)$. Similarly, the reduce function takes the input key as $\bar{k}$ and provides multiple sets of values. The outcome of the reduce function is utilized as the input for the mapping function to improve the MapReduce algorithms. The data provided by the mapping algorithm is segregated by key and transmitted to the reducing algorithm which is known as the shuffled step. By using this step, the grouped mapping is performed to filter out the unrecognized samples and provides the sampled set with better sample values.

3.4.3. RadeMacher average

The samples obtained from the map reduce technique are provided into the stage of RadeMacher average to improve the accuracy of the frequent data mining model. In provided hypothesis h, the generalization error is defined as the probability that the randomized sample gets uncategorized. The general aim of every learning algorithm is to detect a hypothesis with the least error rate, but it can't be suited while evaluating the sample classes because it is dependent on probability distribution. However, training the error samples has the probability to minimize the generalization error and the mathematical expression to minimize the error rate is presented in Eq. (7) as follows:

$$\varepsilon_n(h) = 1/n \sum_{i=1}^{n} L(h(X_i), Z_i) \tag{7}$$

Where L is known as the loss function and it is defined as $L = \{(1, if z \neq z'), (0, \text{otherwise})\}$

The empirical risk minimization technique aids in the occurrence of a minimal error rate and ensures the ERM with small error bound which lies in the range of $Sup_{h \in H} |\varepsilon_P(h) - \varepsilon_n(h)|$. The variation in the error of hypothesis and true generalization convergence tends to infinity. The RadeMacher value lies in the range of +1 and -1 with the individual probability of ½. Assume the RadeMacher variables as $r_{1,}r_2, \ldots, r_n$and the RadeMacher penalty is defined in Eq. (8) as follows:

$$R_n(H) = Sup_{h \in H} |1/n \sum_{i=1}^{n} r_i\, L(h(X_i), y_i)| \tag{8}$$

The symmetrical inequality of the empirical process is represented in Eq. (9) as follows:

$$E = \{Sup_{h\in H}|\varepsilon_P(h) - \varepsilon_n(h)|\} \leq 2E\{R_n(H)\} \quad (9)$$

Where the choices were taken among the RadeMacher random variables. The standard concertation of the inequalities with minimal probability value is denoted as following Eq. (10):

$$\varepsilon_p(h) \leq \varepsilon_n(h) + 2\, R_n(H) + \eta(\delta, n) \quad (10)$$

Where $\eta(\delta, n)$ is the smallest error term which regulates the randomization among the samples. The RadeMacher penalties are applied to provide an approximate solution for the progressive sampling algorithm. The minimum count of samples which is required to verify ERM with the distance ε cretes reduced generalization error for every $h \in H$ and the RadeMacher stopping time $V(\varepsilon, \delta)$ with the parameters (ε, δ), the value of $V(\varepsilon, \delta)$ is defined in Eq. (10) as follows:

$$V(\varepsilon, \delta) = min\{n_i = 2^i n_0(\varepsilon, \delta)|R_{ni}(H) < \varepsilon\} \quad (11)$$

3.4.4. RadeMacher average bounding

This section describes the steps involved in bounding the RadeMacher average. Initially, the item sets are sorted based on their frequencies. Assume $T_S(\{a\})$ is the set of sorted transaction items that comes under the set of broken arbitrary items. Consider the transaction $T_S(\{a\})$ which is in the order of set C_a. The following steps must be employed before transactional sample orders,

1. Before performing transactions with high amounts of C_a, the items set should be allocated in order of a.
2. The total count of transactions depends on the tie-breaking criterion where zero or more transaction item occurs in the order of a.

Let $C_{a,r}$ be the count of transactions which is contained in the set of sorted transactional items $T_S(\{a\})$ which comes in the order of set a.

The least number of items contained in the transactional set $T_S(\{a\})$ is denoted as $h_{a,r}$ which is presented in Eq. (12) as follows:

$$h_{a,r} = \sum_{j\geq r}^{X_a} g_{a,j} \quad (12)$$

Where X_a is the maximum value of r which exist in the transaction set and $h_{a,r}$ is the least number of items present in the set.

3.4.5. Mining the frequent item set

The probable conditions to stop scanning the sample are verified and the minimum index value $\widetilde{\omega}$ is evaluated and detected by computing s. To compute the value of $\widetilde{\omega}$, the sample values of $h_{a,r}$ and $g_{a,j}$ are considered. When the order of sample is obtained from the varying frequencies, then it is sufficient to look over every individual transaction sets based on the order of every item set. Slight modifications need to be performed to get the algorithm for evaluating the approximates to the set of top frequent item sets. The evaluation of algorithm is evaluated by stricter stopping conditions and performs an accurate mining algorithm to identify the high frequent item sets in the provided data samples. Thus, the RadeMacher average is used in progressive sampling technique to mine out the frequent item sets.

4. Results and analysis

This section provides results and analysis of the proposed empirical sampling algorithm using RadeMacher average. The efficiency of the proposed method is evaluated by means of run time, absolute estimation error and prediction error. The result section is sub sectioned into two categories such as performance analysis and comparative analysis. In performance analysis, the efficiency of the RadeMacher average used in the proposed sampling algorithm is evaluated with Vapnik-Chervonenkis (VC) dimension. In a comparative analysis, the efficiency of the proposed sampling algorithm using RadeMacher average is evaluated with the existing sampling techniques discussed in the related works of this paper.

4.1 Experimental setup

The proposed sampling algorithm using RadeMacher average is implemented using python programming language to evaluate the estimation error, prediction error and run time. The evaluation is carried out in a system with an Intel i5 2.7 GHz processor, 6GB random access memory (RAM) and windows 10 operating system.

4.2 Performance analysis

The performance of the RadeMacher average used in the proposed sampling algorithm is evaluated with the efficiency of VC dimension method. The performance among them is evaluated by means of runtime, absolute estimation error and selective prediction error. Table 1 shown below provides the

Received: May 24, 2023. Revised: June 29, 2023.

Table 1. Performance evaluation based on run time

Number of Samples	Run time of RadeMacher average (ms)	Run time of VC dimension (ms)
20	132	143
40	148	159
60	169	175
80	198	210
100	212	223

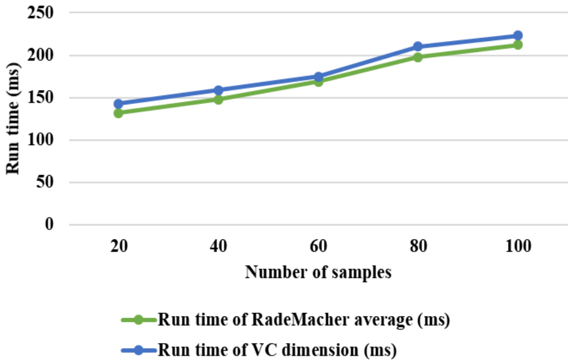

Figure. 2 Graphical representations for runtime

Table 2. Performance evaluation based on the absolute estimation error

Run time (ms)	Absolute estimation error for RadeMacher average (%)	Absolute estimation error for VC dimension (%)
120	0.21	0.28
140	0.33	0.40
160	0.46	0.53
180	0.62	0.78
200	0.89	0.97

results obtained by evaluating the run time, estimation error, the prediction error and the results are evaluated based on the number of samples. Table 1 provides the results obtained from the run time of RadeMacher and VC dimensions.

The results from Table 1 show that the RadeMacher average has taken minimum run time which ranges from 132ms-212ms whereas the VC dimension has taken 143ms -223ms for the same number of samples. This shows that the RadeMacher has taken minimum run time while comparing with the VC dimension. The better result of RadeMacher is due to the efficiency of RadeMacher bounds which aids in precise theoretical settings and distribution dependent. The graphical representation for the run time is shown in Fig. 2 as follows,

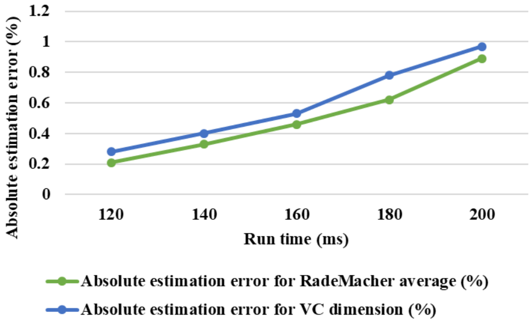

Figure. 3 Graphical representations for the results of absolute estimation error

Table 3. Performance evaluation based on prediction error

Number of Samples	Prediction error of RadeMacher average (%)	Prediction error of VC dimension (%)
20	0.18	0.22
40	0.24	0.37
60	0.29	0.43
80	0.41	0.51
100	0.54	0.67

Secondly, the performance of RadeMacher average is compared with VC dimension by means of absolute estimation error. The absolute estimation error is evaluated by measuring the variation between the inferred value and the actual value. Table 2 represented below shows the result of the error while evaluating different run times.

The results from Table 2 show that the RadeMacher average used in the proposed method obtained a minimum error rate which ranges from 0.21% to 0.89% whereas the error rate of VC dimension ranges from 0.28% to 0.97%. This result shows that RadeMacher has occurred a minimum error while comparing with VC dimension. The graphical representation for evaluation of absolute estimation error is shown in Fig. 3 as follows,

Finally, the performance is evaluated by means of prediction error. The prediction error is evaluated by the variation that occurred while predicting the number of frequent items set for the provided number of sample data. Table 3 shows the results obtained from prediction error while predicting the frequent item sets.

The results from Table 3 show that the RadeMacher average has obtained a minimum error rate of 0.18% to 0.54% which is comparatively lower than the prediction error of the VC dimension which is 0.22% to 0.37%. The better result is due to the bound used by RadeMacher which defines the upper limit of the input data to maintain it between the probably approximate correct (PAC) Framework. The graphical representation for the results of prediction error is represented in Fig. 4 as follows:

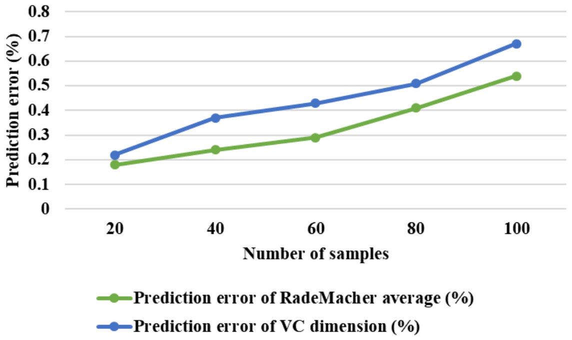

Figure. 4 Graphical representations for the results obtained from prediction error

Table 4. Comparative analysis for real time data obtained from smart phone accelerometer

Sampling methods	Dataset	Processing time (ms)
ScaSRS [19]	Smart phone accelerometer	362
	Smart phone gyroscope	454
ESA-RMA	Smart phone accelerometer	212
	Smart phone gyroscope	387

4.3 Comparative analysis

This section provides the comparison results of the proposed empirical sampling algorithm with sampling with existing approaches. The comparison is performed with two datasets such as HAR dataset and the real time dataset from accelerometer and gyroscope. The existing ScaSRS [19] is used to evaluate the proposed approach with real time data obtained from accelerometer and gyroscope which is represented in Table 4. In Table 5, the proposed approach is evaluated with VertTIRP [22] for HAR dataset.

The results obtained from Table 4 show that the proposed ESA-RMA has achieved better results by taking minimal processing time than ScaSRS for both the data obtained from smart phone accelerometer and smart phone gyroscope. For instance, the run time of the proposed ESA-RMA when evaluated for smart phone accelerometer is 212ms which is comparatively lower than the time taken by ScaSRS (362 ms). Similarly, for smart phone gyroscope dataset, the proposed approach has taken processing time 387 ms which is relatively minimum than the existing ScaSRS (454 ms). The better result of the proposed approach is due to the bounds of RadeMacher which helps in data distribution and its efficiency in defining the limit of the input data to maintain it between the probably approximate correct (PAC) framework.

Table 5. Comparative analysis for HAR dataset

Sampling methods	Dataset	Processing time (s)
VertTIRP [22]	HAR dataset	7.82
ESA-RMA		5.43

Secondly, the performance of the proposed approach is evaluated with HAR dataset. The Table 5 depicted below presents the result obtained while evaluating the proposed approach with VertTIRP. The comparison is performed by considering the run time as a common performance metric for 100 epsilons.

The results from Table 2 shows that the proposed ESA-RMA had took minimum time of 5.43 sec to mine the frequent pattern from HAR dataset whereas the existing VertTIRP took processing time of 7.82 sec. An effective data distribution performed by RadeMacher approach helps to describe the limit of the input data and helps to mine the frequent itemset effectively.

5. Conclusion

Sampling the big data is considered a challenging task due to the presence of frequent item sets present in it. This research introduced an empirical sampling algorithm for the application of frequent item set mining. The proposed empirical approach utilized RadeMacher average for bounding the samples and the bounds of RadeMacher which helps in data distribution and its efficiency in defining the limit of the input data. Furthermore, the experimental findings validate the efficiency of the proposed ESA-RMA technique to its processing time. The performance of ESA-RMA is evaluated with the existing sampling methods such as ScaSRS and VertTIRP. The obtained results show that the proposed ESA-RMA has taken a minimum run time of 212 ms for smart phone accelerometer dataset whereas ScaSRS have taken processing time of 387 ms. In the same way for HAR dataset, the proposed approach had taken the processing time of 5.43 ms whereas the existing VertTIRP had took 7.82 ms respectively. The future work will be based on implementing the proposed approach with high complex datasets.

Received: May 24, 2023. Revised: June 29, 2023.

Notation list

Parameter	Description
f_D	Normal frequency value of the Itemset
A	Arbitrary order
T_D	Transaction dataset
$f_D^{(k)}$	Maximum frequency value of the Itemset
I	Itemset
n	Number of iteration
ε	Approximation value
R	Range of Itemset
s	Size of the sample
$S_{optimal}$	Statistic optimal size based on divergence
$S_{schedule}$	Scheduled sample
S_i	Selected sample
k, v	Key value pairs
L	Loss function
$\eta(\delta, n)$	Error term
$V(\varepsilon, \delta)$	RadeMacher stopping time
$T_s(\{a\})$	Set of sorted transaction items
$h_{a,r}$	Least number of item in set of sorted transaction
$C_{a,r}$	Count of transactional item set
X_a	Maximum value in transaction set

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

For this research work all authors' have equally contributed in Conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing.

References

[1] M. R. A. Bana, M. S. Farhan, and N. A. Othman, "An Efficient Spark-Based Hybrid Frequent Itemset Mining Algorithm for Big Data", *Data*, Vol. 7, No. 1, p. 11, 2022.

[2] H. N. Dao, P. Ravikumar, P. Likhitha, U. K. Rage, Y. Watanobe, and I. Paik, "Finding Stable Periodic-Frequent Itemsets in Big Columnar Databases", *IEEE Access*, Vol. 11, pp. 12504-12524, 2023.

[3] W. Xiao, and J. Hu, "SWEclat: a frequent itemset mining algorithm over streaming data using Spark Streaming", *The Journal of Supercomputing*, Vol. 76, No. 10, pp. 7619-7634, 2020.

[4] Y. Zhang, W. Yu, X. Ma, H. Ogura, and D. Ye, "Multi-objective optimization for high-dimensional maximal frequent itemset mining", *Applied Sciences*, Vol. 11, No. 19, p. 8971, 2021.

[5] M. Yasir, M. A. Habib, M. Ashraf, S. Sarwar, M. U. Chaudhry, H. Shahwani, M. Ahmad, and C. H. M. N. Faisal, "D-GENE: deferring the GENEration of power sets for discovering frequent itemsets in sparse big data", *IEEE Access*, Vol. 8, pp. 27375-27392, 2020.

[6] S. Bagui, and P. Stanley, "Mining frequent itemsets from streaming transaction data using genetic algorithms", *Journal of Big Data*, Vol. 7, p. 54, 2020.

[7] R. Chen, S. Zhao, and M. Liu, "A fast approach for up-scaling frequent itemsets", *IEEE Access*, Vol. 8, pp. 97141-97151, 2020.

[8] D. T. Tran, M. Gabbouj, and A. Iosifidis, "Subset sampling for progressive neural network learning", In: *Proc. of 2020 IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, pp. 713-717, 2020.

[9] C. Wang and X. Zheng, "Application of improved time series Apriori algorithm by frequent itemsets in association rule data mining based on temporal constraint", *Evolutionary Intelligence*, Vol. 13, No. 1, pp. 39-49, 2020.

[10] P. Goyal, J. S. Challa, S. Shrivastava, and N. Goyal, "Anytime frequent itemset mining of transactional data streams", *Big Data Research*, Vol. 21, p. 100146, 2020.

[11] N. Bangera, and N. Kayarvizhy, "A Progressive Sampling based Approach to Reduce Sampling Time", In: *Proc. of 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, Bangalore, India, pp. 74-78, 2019.

[12] N. Aryabarzan and B. M. Bidgoli, "Neclatclosed: A vertical algorithm for mining frequent closed itemsets", *Expert Systems with Applications*, Vol. 174, p. 114738, 2021.

[13] N. D. C. García, A. L. M. Castaneda, D. E. Garcia, and M. V. Carriegos, "Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm", *Complexity*, Vol. 2019, p. 6278908, 2019.

[14] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example", *IEEE Transactions on Image Processing*, Vol. 28, No. 6, pp. 2872-2881, 2019.

[15] E. Higson, W. Handley, M. Hobson, and A. Lasenby, “Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation”, *Statistics and Computing*, Vol. 29, No. 5, pp. 891-913, 2019.

[16] P. P. J. Suresh, U. D. Acharya, and N. V. Reddy, “Mining frequent Itemsets from transaction databases using hybrid switching framework”, *Multimedia Tools and Applications*, pp. 1-21, 2023.

[17] S. S. Schreiber, M. Riondato, and E. Zgraggen, “ProSecCo: Progressive sequence mining with convergence guarantees”, *Knowledge and Information Systems*, Vol. 62, No. 4, pp. 1313-1340, 2020.

[18] D. Santoro, A. Tonon, and F. Vandin, “Mining sequential patterns with VC-dimension and Rademacher complexity”, *Algorithms*, Vol. 13, No. 5, p. 123, 2020.

[19] K. Djouzi, K. B. Bey, and A. Amamra, “A new adaptive sampling algorithm for big data classification”, *Journal of Computational Science*, Vol. 61, p. 101653, 2022.

[20] M. Lei, L. Chu, Z. Wang, J. Pei, C. He, X. Zhang, and B. Fang, “Mining top-k sequential patterns in transaction database graphs: A new challenging problem and a sampling-based approach”, *World Wide Web*, Vol. 23, No. 1, pp. 103-130, 2020.

[21] Yamuna Devi, N. “A Parallel Direct-Vertical Map Reduce Programming model for an effective frequent pattern mining in a dispersed environment”, *Concurrency and Computation: Practice and Experience*, Vol. 33, No. 24, p. e6470, 2021.

[22] N. Mordvanyuk, B. Lopez, and A. Bifet, “vertTIRP: Robust and efficient vertical frequent time interval-related pattern mining”, *Expert Systems with Applications*, Vol. 168, p. 114276, 2021.

[23] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. R. Ortiz, “A public domain dataset for human activity recognition using smartphones”, In: *Proc. of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Vol. 3, p. 3, 2013.

[24] Link for UCI repository dataset: https://archive.ics.uci.edu/ml/datasets/Intelligent+Media+Accelerometer+and+Gyroscope+%28IM-AccGyro%29+Dataset