Grenze International Journal of Engineering and Technology, June Issue

Feature Extraction and Reduction by using Modified Apriori algorithm

Dr. Asha T[1], Pratyush Vatsa[2], Khushwant Kumar[3], Raj Kumar Karmakar[4], Sugam Chand M[5] and Dr. Maya B. S[6]
[1-6]Bangalore Institute of Technology/Computer Science & Engineering, Bengaluru, India.
Email: sugamgowda284d@gmail.com, ashat@bit-bangalore.edu.in, sspratyush7@gmail.com, khushwantkumar30062001@gmail.com, rajkumarkarmakar335@gmail.com, mayabs@bit-bangalore.edu.in

***Abstract*— Feature selection is of great importance in the world of data mining. The existing algorithms for doing so lead to extremely high dimensionality when interactions among features are considered. So, an algorithm is needed to extract only useful features. An Improvised approach for association rule mining is devised, which does not require the users to provide support and confidence values and we are considering the interaction among features and relative confidence to get the better result. The user can specify the number of rules and features that have to be generated. This method has an advantage over Standard Apriori, as it gives more control over the efficiency and quality of the results. It is also capable to handle large data sets and high dimensions more effectively. We calculated mean, median, variance and standard deviation on the lift value of rules generated and found that variance of improvised apriori is 30 – 40% lower than that of standard apriori, which indicates that modified Apriori performs better than standard apriori. The mean and median value of the lift is also 20 – 50% higher in modified apriori than in standard apriori, which implies that the features are more tightly coupled in modified apriori than in standard apriori and hence have a higher probability of getting the perfect result.**

***Index Terms*— Data mining, Itemset, Rule, Support, Feature Selection, Association rule mining, Apriori algorithm, Confidence, Relative Confidence, Lift, ARAF.**

I. INTRODUCTION

Data mining is a process of extracting knowledge from large datasets using advanced computational methods, including statistics, machine learning, and artificial intelligence. It involves the discovery of hidden patterns, relationships, and trends within the data, which can be used to make informed decisions. Data mining involves several steps: data collection and preparation, then data modeling, evaluation, and interpretation.

One of the most widely used techniques in data mining is association rule mining. It is a process of discovering interesting relationships or patterns among variables in large datasets. It involves identifying frequently occurring itemsets, and then generating rules based on those itemsets. These rules can be used to make predictions or to gain insights into the behavior of the underlying system.

Association rules are typically represented in the form of "if-then" statements, where the "if" part represents the antecedent, and the "then" part represents the consequent. The strength of the association is measured using statistical measures such as support, confidence, and lift.

The Apriori algorithm is one of the most widely used algorithms for mining frequent itemsets in a dataset. It was proposed by Agrawal and Srikant in 1994[1] and is considered a seminal algorithm in the field of data mining.

Grenze ID: 01.GIJET.9.2.180

The algorithm uses a breadth-first search strategy to generate all possible itemsets in a dataset and uses a downward closure property to eliminate candidate itemsets that are infrequent. The downward closure property states that any subset of a frequent itemset must also be a frequent itemset. The Apriori algorithm is based on the concept of support. The algorithm scans the dataset to determine the support of each individual item in the dataset. It then uses these support values to generate candidate itemsets of size two by joining frequent itemsets of size one. The algorithm continues this process to generate candidate itemsets of increasing size until no more frequent itemsets can be generated. Once all frequent itemsets have been identified, association rules can be generated by applying a minimum confidence threshold. Association rules are statements of the form $X \rightarrow Y$, where X and Y are itemsets and the arrow denotes implication.

The Apriori algorithm is closely related to association rule mining because it is used to identify frequent itemsets, which are the basis for generating association rules. Although the Apriori algorithm is a widely used algorithm for association rule mining, it has some drawbacks, including High computational complexity, Large memory requirements, Generating a large number of candidate itemsets, and Limited ability to handle noisy data. Also, we have to specify the support and confidence that might omit some of the important features required for the prediction analysis. Using the above knowledge obtained from these methods, we can apply the same to datasets such as market basket analysis, medical diagnosis, financial fraud detection, and so on for prediction analysis.

II. Related Works

[3] The authors propose a simple method for sure screening interactions (SSI) in this paper. They propose a fast algorithm, named "BOLT-SSI". The statistical theory has been established for SSI and BOLT-SSI, guaranteeing their sure screening property. The performance of SSI and BOLT-SSI are evaluated by comprehensive simulation and real case studies. However, the above method runs only for certain classified datasets. If there is an unclassified dataset, the screening method does not work, and another method must be implemented.

[4] The authors introduce a highly efficient algorithm for generating meaningful association rules from a database. This algorithm incorporates innovative estimation and pruning techniques, as well as efficient buffer management. Unlike traditional methods that rely on minimum support and confidence thresholds, this algorithm maximizes the utilization of generated rules by avoiding the omission of significant rules that may not meet those thresholds.

[5] The paper introduces efficient algorithms for discovering frequent itemsets, which is a computationally demanding task. These algorithms leverage the structural properties of frequent itemsets to enable rapid discovery. The database items are grouped into clusters, which represent potential maximal frequent itemsets. Each cluster creates a subset of the itemset lattice, forming a sub-lattice.

[6] In this paper, the authors explain the fundamentals of association rule mining and moreover derive a general framework. Based on this they describe today's approaches in context by pointing out common aspects and differences.

[7] This paper aims to improve such an exhaustive search-based classification system CBA (Classification Based on Associations). We use this method for classifying the association rules.

[8] The paper introduces CMAR, a novel method for associative classification called Classification based on Multiple Association Rules. CMAR extends the efficient frequent pattern mining technique, FP-growth, by constructing a class distribution-associated FP-tree and efficiently mining a large database. The classification process is based on weighted analysis using multiple robust association rules. However, the complexity increases due to the need to scan the tree, which can vary depending on the order in which it is formed.

[9] In this paper, the authors introduce a new method called Backtracking. It can be incorporated into many existing high-dimensional methods based on penalty functions and works by building increasing sets of candidate interactions iteratively. Models fitted on the main effects and interactions selected early on in this process guide the selection of future interactions. However, the number of times a function calls itself can vary largely due to which the complexity of the algorithm may vary indefinitely.

[10] In this paper, the author discusses the methods one can use for pattern mining on various datasets and across different dimensionality.

III. Motivation

Association rule mining is a valuable tool for uncovering meaningful patterns in large datasets across a range of domains. However, the existing Apriori algorithm used for association rule mining has certain limitations, including an unpredictable runtime and the need for users to set two difficult-to-determine parameters. Recent works have proposed innovative approaches to mining frequent itemsets and association rules, including dynamic

parameters, pruning techniques, and ranking mechanisms. By combining and adapting these methods, we aim to develop an enhanced version of the Apriori algorithm that addresses the limitations of the existing algorithm and provides more efficient, scalable, and high-quality results for association rule mining.

IV. Problem Domain

Data mining is the process of extracting knowledge from large datasets using advanced computational methods. It involves discovering hidden patterns, relationships, and trends within the data to drive business insights and informed decision-making. Association rule mining is a widely used technique in data mining that involves identifying frequently occurring itemsets and generating rules based on those itemsets to gain insights and make predictions. However, association rule mining faces several challenges and limitations, such as high computational complexity, large memory requirements, difficulty in setting minimum support and confidence values, generating a large number of candidate itemsets and rules, limited ability to handle noisy data, and ranking rules based on multiple criteria. To address these challenges and limitations, there is a need for developing efficient and effective methods for association rule mining that can provide better results.

V. Problem Definition

The existing Apriori algorithm requires the user to specify two parameters: the minimum support and the minimum confidence. These parameters determine which itemsets and rules are considered frequent and interesting, respectively. However, setting these parameters can be challenging and subjective, as different values may lead to different results. Moreover, these parameters may not capture all the relevant or useful patterns in the data, as they only consider the frequency and accuracy of the rules, but not other factors such as novelty, diversity, or utility. Furthermore, the existing Apriori algorithm may generate a large number of candidate itemsets and rules, many of which can be redundant or irrelevant and may not contribute enough to produce the desired output. Therefore, there is a need for a modified version of the Apriori algorithm that can provide more precise and meaningful results by allowing the user to specify the number of itemsets and confident rules that they want to consider, instead of the minimum support and minimum confidence values. This would make our algorithm more user-friendly and applicable to people who are not domain-specific.

VI. Problem Statement

Propose an algorithm for feature selection, interaction and reduction inspired by association rule mining methods and verify the effectiveness of the proposed algorithm by conducting a series of experiments.

VII. Innovative Content

We propose an algorithm that uses association rule mining to select feature sets with high relative confidence, which indicates a strong association with the class label. We also reduce redundancy by removing subsumed features. We compare our algorithm with existing algorithms on benchmark datasets and show that it can select a small subset of relevant and non-redundant features with comparable or better classification performance.

VIII. Problem Formulation

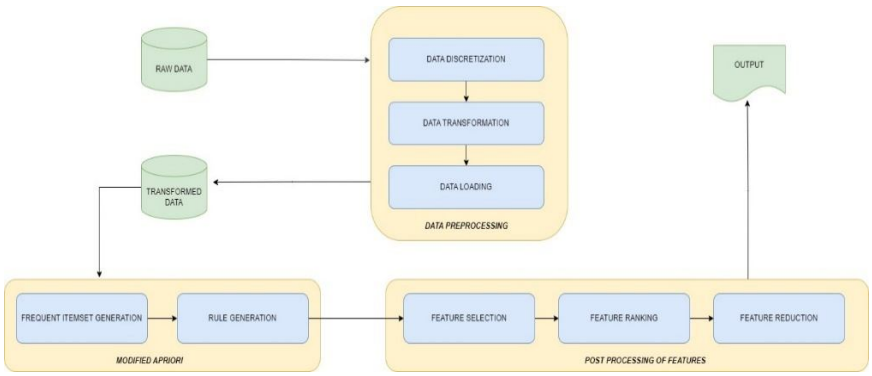

Fig. 1: Representation Design.

In this section, we provide the design that we have used for capturing the problem statement. The problem is captured in four steps as given in Fig. 1. First, the data is fed into the algorithm after preprocessing. Now, we perform feature selection on the provided dataset by using the association rule mining algorithm that we propose in later sections. Following that, we extract feasible interactions among the generated features and remove the redundant features. Finally, we evaluate the effectiveness of the proposed algorithm and record the results.

IX. METHODOLOGY

The Apriori Algorithm is a popular algorithm used for finding frequent itemsets in a dataset. It uses a bottom-up approach where frequent subsets of items are iteratively generated and combined to form larger itemsets. The algorithm reduces the search space by eliminating infrequent itemsets that cannot be part of any frequent itemset. The algorithm was first proposed by Rakesh Agrawal and Ramakrishnan Srikant in their 1994 paper titled "Fast Algorithms for Mining Association Rules"[1]. The main idea of the Apriori algorithm is that only the itemset containing no infrequent itemsets needs examining whether it is frequent. Therefore, the number of candidate itemsets is reduced rapidly.

Algorithm 1 is an algorithm to extract features from a set of association rules. The input to the algorithm is a set of association rules denoted as AR. The output of the algorithm is a set of features represented by F. For each association rule, the algorithm checks if its right-hand side has only a class label. If the rule satisfies this condition, then the antecedent of the rule, which is the left-hand side of the arrow, is added to the set of features. Finally, once all the association rules have been processed, the algorithm returns the set of features as the output.

Algorithm 1 Generate Features

Input: AR(a set of association rules);
Output: F(a set of features);
1: $F = \emptyset$
2: for all R in AR do
3: if R has the form of "$(X_i = x_i, X_j = x_j, \ldots, X_m = x_m) \rightarrow (Y = c)$" then
4: $F \leftarrow F \cup \{(X_i = x_i, X_j = x_j, \ldots, X_m = x_m)\}$
5: end if
6: end for

Algorithm 2 Apriori Algorithm [2]

Input: D(database);
minsupp(minimum support);
minconf(minimum confidence);
Output: R_t (all association rules)
1: L1= {frequent 1-itemsets}
2: for k = 2; L_{k-1} != ∅; k++ do
3:C_k = apriori-gen (L_{k-1});
4: for all transactions T ∈ D do
5: C_t = subset (C_k, T);
6: for all candidates C ∈ C_t do
7: Count(C)←Count(C)+1;
8: end for
9:end for
10: $L_k = \{C \in C_t | \text{Count}(C) \geq \text{minsupp} \cdot |D|\}$
11: end for
12: $L_f = U_k L_k$
13: Return R_t=GenerateRules(L_f , minconf)

Standard Apriori Algorithm with association rules as input (Algorithm 3) -
Our method applies to association rules as features (ARAF) [2]. The association rules here mean class association rules specifically. The feature is a main effect if the antecedent has only one item, otherwise, it's an interaction.

ARAF has two advantages. First, it can handle any antecedent size in theory. Second, it is easy to interpret. Rules are very understandable for humans, and the features from rules inherit this property.

Algorithm 3 Standard Apriori

Input: D(database);
minsupp(minimum support);
minconf(minimum confidence);
Output: F (generated features);
1: Run the Apriori algorithm to get Lk
2: Generate the confident rules from Lk
for i from 1 to k:
CR_i ={R="$(X_1 = x_1, X_2 = x_2, \ldots X_i = x_i) \rightarrow Y = c$" |conf(R)≥minconf,
$(X_1 = x_1, X_2 = x_2, \ldots X_i = x_i, Y=c) \in L_i$}
3: Return F = GenerateFeatures($U_i\, CR_i$)

The parameters of minimum support (minsupp) and minimum confidence (minconf) are not easy to determine and have some limitations. The values of these parameters affect the number and quality of the association rules that are produced. If the values are too high, the algorithm may not find any rules or only a few that are not very useful. On the other hand, if the values are too low, the algorithm may generate many rules that are not reliable or relevant. This can also cause problems with the space and time efficiency of the algorithm, as it has to store and process a large number of rules.
To avoid the above drawbacks caused by the Standard Apriori algorithm we propose a new algorithm/method called Modified Apriori with feature reduction, in which we use number of frequent itemsets and rules instead of support and confidence thresholds.
Unbalanced data is a situation where the number of observations in one class is significantly higher or lower than the number of observations in the other classes. This can lead to biased or inaccurate model performance, as the model tends to learn more about the majority class, rather than the minority class.
Modified Apriori –

Algorithm 4 Modified Apriori

Input: D(database);
d_{freq}(the number of frequent itemsets);
d_{conf} (the number of confident rules);
Output: F(generated features);
1: Count the support of itemsets in IS(c)1, where
$IS_1^{(c)}$={$(X_i=x_i, Y=c)$, for $i \in \{1,2,...,m\}$, for $c \in C$}
2: Generate $FS_1^{(c)}$={d_{freq}/|C| most frequent itemsets in $IS_1^{(c)}$}
3: k = 2
4: while $IS_k^{(c)}$!= ∅ and k < number of columns:
Count the support of itemsets in IS_k, where
$IS_2^{(c)}$ ={ $(X_i =x_i, X_j = X_j , \ldots\ldots, X_k=x_k, Y=c)$}
All k-1 length subsets of $(X_i = x_i, X_j = X_j, \ldots\ldots, X_k = x_k)$ are in $FS_{k-1}^{(c)}$
Generate $FS_k(c)$={d_{freq}/|C| most frequent itemsets in $IS_{k-1}^{(c)} \cup IS_k^{(c)}$}
5: Calculate the relative confidence of rules in AR, where
AR = {class association rules generated from $FS_k^{(c)}$}
6: Generate CR={d_{conf} most relatively confident rules in AR}
7: Return F=GenerateFeatures(CR)

The method that we are currently proposing is for the Modified version of the Standard Apriori algorithm where we are reducing the complexity [and dimensionality]. We have created certain functions which help us in reducing the complexity. The functions which we have implemented are namely, generating frequent itemsets, generating candidate itemsets, and generating confident association rules from frequent itemsets.
Consider a rule A -> B with a confidence of 0.8 and another rule C -> B with a confidence of 0.6. Although the confidence of the first rule is higher, the relative confidence of the second rule may be higher because it may have

fewer antecedents or a more significant antecedent than the first rule. Therefore, relative confidence is useful in identifying the most important rules for a specific class.
By using relative confidence, we can select the most relevant rules for each class based on their strength of association. This can help in identifying the most significant features that contribute to the prediction of a specific class.
Relative confidence(rconf) is a measure of the strength of association between two sets of items in a transactional dataset. It is calculated as the confidence of a rule (i.e., the conditional probability of the consequent given the antecedent) divided by the support of the antecedent. The formula for relative confidence is:
rconf(X->Y) = (supp(X U Y)/(supp(X) – supp(X U Y))/(supp(Y)/(n-supp(Y)).
Relative confidence provides a way to evaluate the importance of a rule while taking into account the frequency of the antecedent in the dataset. Rules with high relative confidence indicate a strong association between the items in the antecedent and consequent and can be used to identify patterns in the data.
Modified Apriori with feature reduction –

Algorithm 5 Modified Apriori with feature reduction

1: All the other steps are the same as Algorithm 4, except
2: generating AR(in Step 5 there) by following steps.
3: For i from 2 to k
4: For all c ∈ C do
5: For all I in $FS_i^{(c)}$ do
6: If I ∈ $IS_1^{(c)}$ then
7: Suppose I=(Xi = xi; Y = c)
8: Denote rule="(Xi = xi) → (Y = c)"
9: AR ← AR U rule
10: else
11:Suppose I=(Xi = xi, Xj = xj,…. Xk = xk; Y = c)
12: Denote rule=(Xi = xi, Xj = xj,…. Xk = xk) → (Y = c)
13: add_rule = true
14: For all item ∈ I do
15: if rconf(item) > rconf(I)
16: add_rule = false
17: end if
18: end for
19: if add_rule == true then
20:AR ← AR U rule
21: end if
22: end for
23: end for
24: end for
25: return AR

Considering we have 3 feature sets, A, A, B, and A, C we will consider only that feature which has the highest relative confidence among 3 and a subset is present in all the feature sets which we are considering.

Algorithm for reduce_redundancy –

Inputs:
Rules
Transactions
Output:
1:Verdict
2: rules = ∅
3: for rules_out in confident_rules:
4:out_rconf = rules_out.rconf
5:rule_considered = rules_out

6:for rules_in in confident_rules:
7:in_rconf = rules_in.rconf
8:if in_rconf > out_rconf:
9:out_rconf = in_rconf
10:rule_considered = rules_in
11:rules = rule_considered
12: return rules

X. Results & Data Model

TABLE I: COMPARISON TABLE BETWEEN STANDARD APRIORI AND MODIFIED APRIORI WITH FEATURE REDUCTION.

	Standard Apriori				Modified Apriori with Feature Reduction			
Dataset	**Mean**	**Median**	**Variance**	**Standard deviation**	**Mean**	**Median**	**Variance**	**Standard deviation**
diabetes	56.51561	40.6639	3529.923	59.41315537	105.415	90.00985	2786.863	52.79074267
country_wise	41.67521	26.71429	474.9322	21.79293983	59.26848	53.42857	354.9647	18.84050565
heart	242.4515	227.5085	2617.436	51.16088417	234.9043	246.5898	2923.957	54.07362205
heart_2020_cleaned	333.4553	312.9264	4671.252	68.34655555	543.141	529.0844	1370.84	37.02485587

The above table provides the mean, median, variance, and standard deviation for two different versions of the Apriori algorithm applied to four different datasets: diabetes, country_wise, heart, and heart_2020_cleaned.
In the above table, the mean, median, standard deviation, and variance of lifts have been compared for the features generated by the standard and Modified Apriori algorithms respectively applied to four different datasets: diabetes, country_wise, heart, and heart_2020_cleaned. Lift is a measure of how good a rule or a model is at predicting something. A high lift means the rule is very accurate and useful, while a low lift means the rule is not much better than random guessing.
We can observe that for diabetes, country_wise and heart_2020_cleaned datasets, mean and median of lifts are higher in Modified Apriori than Standard Apriori. Also, standard deviation and variance are lower in Modified Apriori than Standard Apriori. For the heart dataset, mean and median of lifts are lower in Modified Apriori than Standard Apriori. Also, standard deviation and variance are higher in Modified Apriori than Standard Apriori.
Based on the above observations, we can infer that in general, the variance of Standard Apriori is higher than that of Modified Apriori , which indicates that Modified Apriori performs better than Standard Apriori . Low variance implies that lifts of the features are closer to the mean value. And since the mean value is higher in Modified Apriori algorithm, the Modified Apriori algorithm gives more accurate features. The mean and median value of the lift is also higher in Modified Apriori than in Standard Apriori , which implies that the features are more tightly coupled in Modified Apriori than in Standard Apriori and hence have a higher probability of getting the perfect result.

XI. Comparision of Results

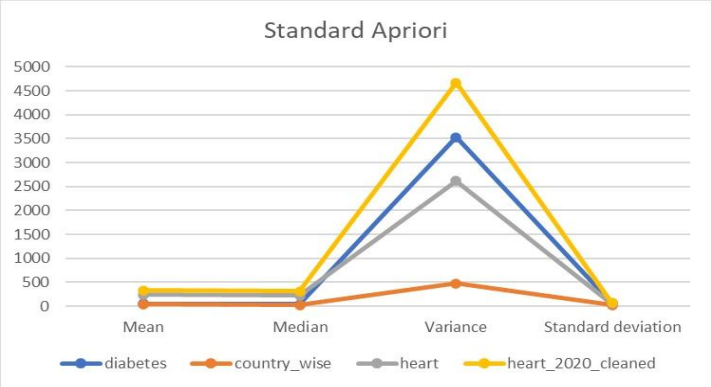

Fig. 2: Output of standard Apriori feature selection.

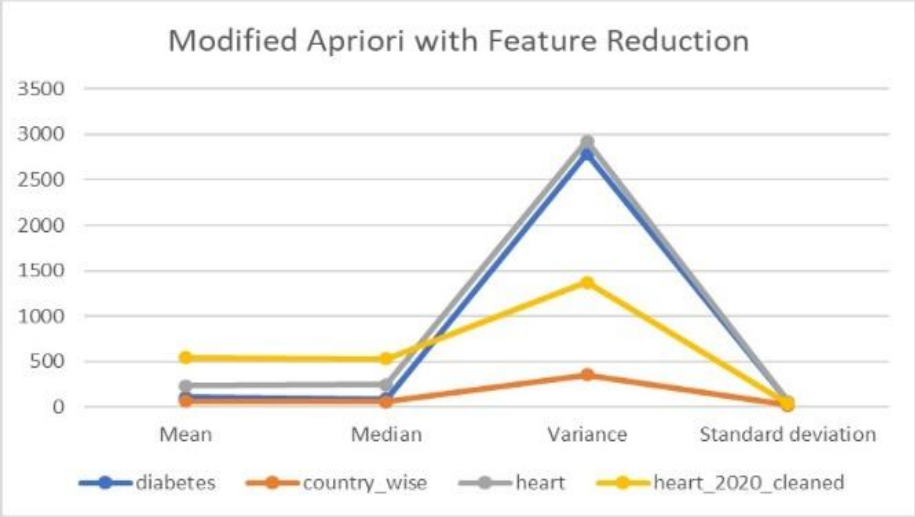

Fig. 3: Output of proposed method for feature selection.

To support the above results we have plotted a line graph denoting the Dataset [X-axis] vs Numerical intervals [Y-axis] for both the algorithms i.e., Standard Apriori and Modified Apriori with feature reduction.
As we can see in the above line chart, we can conclude that Modified Apriori with feature reduction works better when compared to standard Apriori both in terms of Time and Space complexity.

XII. Justification of Result

In standard apriori, we are considering minimum support and confidence which leads to either of the two cases i.e., few or no association rules are generated or the generated association rules may be unreliable. Hence, a large number of association rules are generated and considered, and then the prediction analysis is done.
In Modified Apriori, instead of considering support and confidence, we are considering the itemset count and rule count and computing the relative confidence for each rule used. The higher the relative confidence, the better the rule. Hence, fewer rules are generated and when we consider those rules for prediction analysis, the results generated are better and more efficient.

XIII. Conclusion

Association rule mining is really a fast-growing area nowadays. Researchers aim at finding the best association rules which can be used for providing better analysis. This paper basically gives a more efficient and optimized version of the Standard Apriori algorithm that can be used in various fields for prediction analysis. Also, various evaluation metrics are used to prove this, and statistical data are provided to support this. This research can be further enhanced by considering more factors helpful for reducing the complexity and also various fields in which it can be optimized further.

References

[1] R.Agrawal and R.Srikant,"Fast algorithms for mining association rules", *Proc. 20th Int. Conf. on VLDB, Volume 30*, Pages 487–499, 1994.
[2] Qiuqiang Lin and Chuanhou Gao, "Discovering Categorical Main and Interaction Effects Based on Association Rule Mining", *Transaction on Knowledge and Data Engineering*, Volume 1, Pages 1-1, 2021.
[3] M. Zhou, M. Dai, Y. Yao, J. Liu, C. Yang, and H. Peng, "BOLT-SSI: A statistical approach to screening interaction effects for ultra-high dimensional data," *Arvix*, Volume 3, Pages 30-40, 2019.
[4] R. Agrawal, T. Imieliundefinedski, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, Volume 22, Pages 207–216, 1993.
[5] M. Zaki, S. Parthasarathy, M. Ogihara, and W. li, "New algorithms for fast discovery of association rules," *KDD'99*, Volume 10, Pages 100-111, 1999.
[6] J. Hipp, U. Guntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining — a general survey and comparison," *SIGKDD Explor. Newsl.*, Volume 2, no. 1, Pages 58– 64, 2000.
[7] B. Liu, Y. Ma, and C. K. Wong, "Improving an association rule-based classifier," *PKDD'00*, Volume 32, Pages 504– 509, 2000.

[8] W. Li, J. Han, and J. Pei, “Cmar: Accurate and efficient classification based on multiple class-association rules,”, *ICDM’01. USA: IEEE Computer Society*, Volume 25, Pages 369– 376, 2001.
[9] R. D. Shah, “Modelling interactions in high-dimensional data with backtracking,” *J. Mach. Learn. Res.*, Volume 17, No. 1, Pages 7225–7255, 2016.
[10] S. Ventura and J. M. Luna, “Pattern Mining with Evolutionary Algorithms”, *Springer*, Volume 16, Pages 134-145, 2016