

EMOTION DETECTION IN BIG DATA: DISCOVERY OF FREQUENT ACTIONABLE PATTERNS TO CULTIVATE JOY IN EDUCATION INNOVATION

Angelina Tzacheva¹, Sanchari Chatterjee², Rajia Shareen Shaik³ and Shiva Sai Praneeth Chakinala³

¹Computer Science and Information Technology College of Computing and Engineering, WestCliff University, Irvine, CA 92614,

²Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC, 28223

³Software Engineering Division, Walmart Inc, Bentonville, AR, 72713

ABSTRACT

Action Rules can be seen as a rule-based system designed to uncover actionable patterns hidden within large volumes of data. They are widely used across domains such as healthcare, education, business, finance, art, and social media to support informed decision-making. Action rules provide actionable suggestions for changing an object's state from an existing one to a desired outcome, supporting strategic decision-making and operational optimization. However, traditional action rule extraction models, which analyze data in a non-distributed fashion, suffer significant performance degradation when processing large datasets. To address this critical scalability challenge, we improved our proposed method called Modified Hybrid Action Rule Method by introducing a new frequency Threshold S (ς) along with the other two thresholds ρ and θ . The new threshold ensures that only stable, high-occurring action rules are retained. This hybrid approach combines existing frameworks to efficiently generate a set of action rules, thereby significantly improving computational performance and enabling robust action rule mining in big data environments.

KEYWORDS

Emotion Detection, Action Rules, Big Data, Scalability, Optimization, Sentiment Mining.

1. INTRODUCTION

Action rules can be seen as rule-based systems that extract actionable patterns hidden in large volumes of data. However, traditional extraction models suffer significant performance degradation when processing large datasets. To address this scalability challenge, this research improves the modified hybrid action rule method by introducing a new frequency threshold S (ς) along with existing thresholds ρ (ρ) and θ (θ). These thresholds ensure that only stable, high-occurring action rules are retained while significantly improving computational performance in big data environments.

Rule-based learning is a simple data mining method that identifies, learns, or develops 'rules' to store, operate, or apply [37]. In general, we constitute rules as given in Equation 1, where the antecedent is a conjunction of conditions, and the consequent is the resulting pattern in the provided data for the given conditions in the antecedent [39].

$$condition(s) \rightarrow result(s) \quad (1)$$

Action Rules are rule-based systems that extract actionable patterns hidden in large volumes of data generated daily across Education, Business, Finance, Medical, Transportation, Retail, and social media domains. Data mining across the domains facilitates the discovery of meaningful insights that enhance strategic decision-making, operational efficiency, customer experience, and support individual and organizational development.

Action Rules provide actionable suggestions on changing an object's state from an existing state to a desired outcome, aligning with market trends, optimizing strategies, and customer satisfaction. Data mining techniques such as rule extraction, classification, regression and clustering are involved to produce tremendous amount of patterns in form of results, regardless of whether they are of user's interest, or not. The main challenge for much of the knowledge discovery and machine learning algorithms is the lack of actionability [43].

For example, recommending the business to improve customer satisfaction [14] and sentiment analysis on Twitter [22]. Action rules follow the representation, like Equation 1, as given more precisely in Equation 2, where Ψ represents a conjunction of stable features, $(\alpha \rightarrow \beta)$ represents a conjunction of changes in values of flexible features and $(\theta \rightarrow \phi)$ represents desired change in decision action which is beneficial to the user.

$$[(\Psi) \wedge (\alpha \rightarrow \beta)] \rightarrow (\theta \rightarrow \phi) \quad (2)$$

Action rules recommending an actionable pattern are prone to incur a definite cost to the user [25], [30]. Cost for actions in action rules may cover time, energy, money, or human resources. Actions being recommended can cause both positive (benefits) and adverse (losses) effects for users [36]. Thus, the action rules-based recommendations system should take on below the cost which is low-cost to the users to help them to make plausible actions. The existing approaches [8,19,26,34] do not consider the cost-effectiveness of recommendations. In [25] [37], the concept of the cost of action rules is introduced and refined. Extracting low-cost action rules from a large dataset can be very time-consuming and requires a distributed and scalable approach to extracting them within a practicable timeframe. Distributed Processing frameworks like Hadoop [7] and Spark [42] have been introduced to make data mining and big data processing faster and easier. The data processing work is distributed among multiple nodes, with each node performing computations on its part of the data, using these frameworks. In this work, we utilize the Apache Spark framework to implement a scalable solution to the proposed Action Graph method, making it suitable for big data [42]. Spark provides APIs such as GraphX for productive parallel processing in large graphs [9].

In this paper, we propose an extension to our previous work on distributed actionable pattern mining with Spark [2]. We extract action rules from the business and student survey datasets, which help to obtain better, desirable outcomes for the future.

The extracted Actionable Patterns provide recommendations that can transform a user's sentiment or emotion into a more desirable state. In this work, we extract Action Rules from Student Survey Data using an enhanced Modified Hybrid Action Rule Mining framework [35]. The main contribution of this work is the introduction of a new frequency threshold $S(\zeta)$, which retains only frequently occurring and stable Action Rules across multiple random samples, complementing the partition threshold ρ for improved scalability. We also aim to suggest ways to improve teaching methods and student learning. Although our experiments focus on Student Survey Data, the proposed method is domain independent and can also be applied to customer

satisfaction, healthcare, and other decision-support applications. We implement and test our system in a scalable environment with big data using the Apache Spark platform.

2. RELATED WORK

Emotion detection has been widely studied in affective computing, sentiment analysis, educational data mining, and business intelligence [12,21,33]. The concept of action rules was introduced by Ras and Wieczorkowska in paper [27] and investigated further in papers [34,36]. The first implementation of action rule discovery called Lisp Miner was proposed by Jan Rauch (see <http://lispminer.vse.cz/>). The strategy basically follows the algorithm DEAR2 proposed in [34].

Action rules mining is applied in many domains including healthcare [5, 17, 32], education [28], art [8, 19], and business [10, 31]. Authors in this papers [2], [23] applied distributed computing frameworks like MapReduce [7] and Spark [42], and then efforts have been made to extract actionable recommendations completely in a clustered setup. Authors in this paper [15] proposed a new action rules discovery strategy based on random distribution of data to multiple sites, combining results from all sites and taking averages on parameters like Support and Confidence. Paper [6] is replacing vertical random distribution of data by vertical distribution based on correlation among attributes defined as their distance measure and also introduces a new parameter called Lightness (average number of action rules covering a testing object in a dataset). Bagavathi [2] handles the load balancing by uniformly distributing the data into partitions based on the decision attribute. Authors [23] introduced a new method of projecting the database into smaller chunks for handling data with many attributes and extracting action rules from them effectively.

The paper [21] navigates the potential implications of employing NLP for legal documentation, emphasizing its role in improving access to justice, bridging linguistic gaps, and fostering inclusivity within the legal system. AI and machine learning integration with AWS SageMaker: provides a comprehensive exploration of the current landscape and forward-looking possibilities in integrating artificial intelligence (AI) and machine learning (ML) using Amazon Web Services (AWS) SageMaker. The paper [41] adeptly navigates through the overview of AWS SageMaker, shedding light on its capabilities and features. Early lexicon-based and rule-driven methods provided interpretability but lacked scalability and robustness in large datasets [18].

Recent approaches employ machine learning and deep learning models such as SVMs, CNNs, RNNs, and transformer-based architectures to enhance classification accuracy [20]. While effective, these models often require large, labeled datasets and provide limited interpretability, particularly in real-world educational and organizational contexts [1].

In educational domains, emotion detection supports learning analytics, student engagement analysis, discussion forum mining, and adaptive learning systems [16,20]. Existing studies emphasize improving learning outcomes but frequently lack explicit mechanisms to control decision confidence and noise [4,13,23,29]. Business-focused research applies emotion analysis to customer feedback, brand perception, and decision support systems. Although scalable, many approaches overlook threshold and frequency constraints, leading to unstable predictions in high-volume data environments [3].

The paper [24] proposes Association Action Rules algorithm which discovers rules directly from raw datasets by extending the focus on identifying frequent patterns and suggesting attribute value changes that can lead to desirable decision outcomes. Unlike conventional action rule

mining, which often relies on pre-mined classification rules, their method integrates association rule mining principles with action rule logic. The proposed algorithm efficiently discovers rules where specific attribute changes, termed flexible, can influence a shift in a decision attribute while stable attributes remain unchanged.

The syntax and semantics of action rules have practical significance and have demonstrated their application in domains requiring decision support, such as customer retention and medical diagnosis. They emphasized the importance of influencing outcomes along with the data patterns. The work laid the theoretical groundwork for later developments in cost-sensitive, distributed, and hybrid action rule mining techniques and continues to be a key reference for researchers working on decision-driven data mining and intelligent systems. Threshold-based models and frequency constraints have demonstrated effectiveness in anomaly detection and pattern mining by improving robustness and interpretability. However, their integration into emotion detection remains limited [40].

Personalized Action Rules for Side Effects Object Grouping [33] introduced a customized approach to action rule discovery by addressing the problem of side effects during actionable knowledge extraction. The method groups objects with similar adverse side effects to generate personalized action rules tailored for each group [11]. Unlike traditional action rule mining, which often overlooks unintended consequences of actions, this work ensures that the extracted rules minimize undesired outcomes while maintaining their effectiveness.

The authors in paper [10] propose a method that extracts action rules from data using a Genetic Algorithm to efficiently explore the ample search space, achieving significantly higher support and confidence.

The traditional action rule mining DEAR methods rely on a two-stage process where classification rules are first mined and then transformed into action rules. These approaches are inherently limited by their dependence on classification models, which are primarily designed for prediction rather than for suggesting actionable changes. To overcome the drawbacks of irrelevance and ineffectiveness of action rules generated by such methods, GA2RM (Genetic Algorithm-Based Action Rule Mining) directly extracts action rules from raw datasets, eliminating the need for an intermediate classification step. GA2RM uses a Genetic Algorithm to explore the ample search space of possible attribute transitions. Encodes each candidate rule as a chromosome and evolves rule populations through selection, crossover, and mutation. The fitness of each rule is evaluated based on multiple-objective criteria, including support, confidence, and actionability. GA2RM consistently identifies a greater number of high-quality and interesting action rules than existing methods, thereby shifting the focus of action rule mining from predictive analytics toward more prescriptive, change-oriented insights.

In this work, we propose a new frequency threshold $S(\zeta)$ – that counts the number of times an action rule is occurring in an execution. We have combined random sampling method with our modified hybrid action rule mining approach with threshold Rho - for the number of partitions, which further improves the computational performance from our previous method that has only one threshold [21]. This generates action rules from 50 Random Sampling method and produces the high frequency rules which are above the threshold $S(\zeta)$. Our work differs by explicitly incorporating partition threshold (ρ), decision threshold (θ), and frequency threshold $S(\zeta)$ into a unified framework, improving stability, interpretability, and applicability across education and business domains.

Table 1. Example Decision System T

X	A	B	C	D
x_1	Y	N	N	D_1
x_2	Y	H	Y	D_2
x_3	Y	H	Y	D_1
x_4	N	N	N	D_2
x_5	N	H	N	D_1
x_6	N	N	Y	D_2
x_7	N	H	Y	D_2
x_8	N	H	N	D_1

We will apply our method to the Student Survey Data; however, this method can also be used for business and healthcare data. We are focusing on our work to suggest ways to improve teaching and student learning methods. We implement and test our system in a Scalable Environment with Big Data using the Apache Spark platform.

3. BACKGROUND

In this section, we provide a basic overview of decision systems, action rules, Spark and GraphX frameworks to help understand our methodology.

3.1. Decision System

Consider a decision system given in Table 1. An information System [23] can be represented as $T = (X, A, V)$ where,

X is a nonempty, finite set of objects: $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

A is a nonempty, finite set of attributes: $A = \{A, B, C, D\}$ and V_a is the domain of attribute a , which represents a set of values for attribute $\{a: a \in A\}$. For example, $V_B = \{N, H\}$.

An information system becomes a decision system if $A = A_{St} \cup A_{Fl} \cup \{d\}$, where d is a decision attribute. We choose attribute d if we need to extract a desired action changing values in $\{d_i: d_i \in V_d\}$. A_{St} is a set of stable attributes and A_{Fl} is a set of flexible attributes. For example, ZIPCODE is a stable attribute, and User Ratings can be a flexible attribute. Let us assume from Table 1 that $C \in A_{St}$. $A, B \in A_{Fl}$ and $D_1, D_2 \in V_d$, and the decision maker is looking for action rules that trigger the decision attribute change from D_1 to D_2 throughout this paper for example.

3.2. Information System

Consider the information system given in Table 2. An information system can be represented as $Z = (X, M, V)$ where, $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, M is non-empty finite set of attributes $\{A, B, C, E, F, G, D\}$, V is the domain of attributes in M , for instance the domain of attribute B in the system Z is $\{B_1, B_2, B_3\}$.

Table 2. Information System Z

X	A	B	C	E	F	G	D
x_1	A_1	B_1	C_1	E_1	F_2	G_1	D_1
x_2	A_2	B_1	C_2	E_2	F_2	G_2	D_3
x_3	A_3	B_1	C_1	E_2	F_2	G_3	D_2
x_4	A_1	B_1	C_2	E_2	F_2	G_1	D_2
x_5	A_1	B_2	C_1	E_3	F_2	G_1	D_2
x_6	A_2	B_1	C_1	E_2	F_3	G_1	D_2
x_7	A_2	B_3	C_2	E_2	F_2	G_2	D_2
x_8	A_2	B_1	C_1	E_3	F_2	G_3	D_2

The information system in Table 2 becomes a decision system if attributes M are classified into flexible attributes M_{Fl} , stable attributes M_{St} and decision attributes d. $M = (M_{St}, M_{Fl}, d)$. From table 2, $M_{St} = \{A, B, C\}$, $M_{Fl} = \{E, F, G\}$, and $d = \{D\}$.

3.3. Action Rules

In this subsection, we give definitions of action terms, action rules, and the properties of action rules [38]

Let $T = (X, A \cup \{d\}, V)$ be a decision system, where d is a decision attribute and $V = \cup \{V_i : i \in A\}$. Atomic action terms are expressions having the form $(m, m_1 \rightarrow m_2)$, where $m \in A$ and $m_1, m_2 \in V_m$. Clearly, $m_1 = m_2$ if $m \in A_{St}$. In that case, we can simplify the expression as (m, m_1) or $(m = m_1)$.

Action Rules can take the form of $t_1 \wedge t_2 \wedge \dots \wedge t_n \rightarrow t_d$, where t_i is an atomic action term for each $i \in \{1, 2, \dots, n\}$ and D is the decision value. Example of action rule: $(a, a_1 \rightarrow a_2) \wedge (b, b_1 \rightarrow b_2) \rightarrow (D, D_1 \rightarrow D_2)$

Properties of Action Rules: Action Rules are considered interesting based on the metrics such as Support, Confidence, Coverage, and Utility. Higher these values, more interesting they are to the end user.

Consider an action rule R of the form:

$(Y_1 \rightarrow Y_2) \rightarrow (Z_1 \rightarrow Z_2)$ where,

Y is the condition part of R; Z is the decision part of R

Y_1 is a set of all left side action terms in the condition part of R

Y_2 is a set of all right-side action terms in the condition part of R

Z_1 is the decision attribute value on left side

Z_2 is the decision attribute value on right side

In [45], the support and confidence of an action rule R is given as

$$Support(\mathcal{R}) = \min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}$$

$$Confidence(\mathcal{R}) = \left[\frac{card(Y_1 \cap Z_1)}{card(Y_1)} \right] \cdot \left[\frac{card(Y_2 \cap Z_2)}{card(Y_2)} \right]$$

Later, Tzacheva et al. [37] proposed new set of formulas for the calculation of Support and Confidence of action rules. Their idea is to reduce the complexities in searching data several times for Support and Confidence of an action rule. The new formula is given below

$$Support(\mathcal{R}) = \{card(Y_2 \cap Z_2)\}$$

$$Confidence(\mathcal{R}) = \left[\frac{card(Y_2 \cap Z_2)}{card(Y_2)} \right]$$

Tzacheva et al. [37] also introduced the concept of utility for action rules. The utility of action rules takes the following form. In most cases, the Utility of Action Rules equals the Old Confidence of the same action rule.

$$Utility(\mathcal{R}) = \left[\frac{card(Y_1 \cap Z_1)}{card(Y_1)} \right]$$

Coverage of an action rule means how many decisions from values, from the entire decision system S, are being fully covered by all extracted action rules. In other words, using the extracted action rules, Coverage defines how many data records in the decision system can successfully transfer from Z1 to Z2 .

3.4. Spark

Spark [42] is a framework that is quite similar to MapReduce [7] for processing large quantities of data in a parallel fashion. Spark introduces a distributed memory abstraction strategy called Resilient Distributed Datasets (RDD) that can perform in-memory computations on nodes distributed in a cluster. The results of each operation are then stored in memory, which can be accessed for future processes and analyses, creating another RDD in turn. Thus, Spark reduces the number of disk accesses for storing intermediate outputs which is similar to Hadoop MapReduce. Spark functions in two stages: 1. Transformation, 2. Action. During the Transformation stage, computations are performed on data splits, and the results are stored in the worker nodes memory as RDD. While the Action stage on an RDD collects results from all the workers and sends them to the driver node or saves the results to a storage unit. With RDDs, Spark helps machine learning algorithms skip innumerable disk accesses during iterations.

4. DATASET DESCRIPTION

To test our methods, we use two datasets: Student Survey Data [35]. Student survey data aim to evaluate students' emotions and overall satisfaction with course teaching methods and group work experiences. The survey is designed to gather meaningful insights into students' feelings towards Active Learning methods and other factors that can aid students in their learning process. The data is collected in the courses that implement the Active Learning methods and teaching style. This survey dataset contains 50 attributes. The original data includes 549 instances and 59 attributes. Data is collected in classes employing Active Learning methods to assess students' opinions about their learning experience in the years 2019 and 2020. The data size on disk is 59 Kilobytes. For scalability purposes to test the performance of our proposed method with Big Data, we replicate the original Student Survey Data 100 times. The replicated dataset has a total of 54900 instances. Size on disk is 5.815 Megabytes.

5. METHODOLOGY

In our paper, we propose a Hybrid Method with Reduced Threshold-S to generate action rules with the highest frequency of occurrence. We have combined random sampling method with our Modified Hybrid Action Rule mining approach with Threshold ρ - for the Number of Partitions, which further improves the computational performance from our previous method that has only one threshold [35]. This generates the Action Rule from 50 Random Sampling method and produces the high frequency rules which are above the Threshold S. We also have compared this method by replicating the original data to create 1 million records and 10980 records. We will apply our method to the Student Survey Data; however, this method can also be used for healthcare and many other datasets. We are focusing on our work to suggest ways to improve teaching and Student Learning methods. We implement and test our system in a Scalable Environment with Big Data using the Apache Spark platform. In this section, the algorithms for Action Rules are described in detail.

5.1. Modified Hybrid Action Rule Mining with Partition Threshold ρ

The Modified Hybrid Action Rule Mining Algorithm incorporates Partition Threshold ρ (ρ) to provide scalability for big data. Threshold ρ enables users to control the size of the intermediate table, preventing performance degradation when table sizes become huge. It horizontally and vertically divides large datasets to reduce computational complexity. Our proposed new method addresses this problem, as the Threshold ρ enables the user to control the size of the table, thereby increases computational speed.

Our proposed method, Modified Hybrid Action Rule Mining with Partition Threshold ρ , is presented in Fig. 2 and the proposed methodology is depicted in Fig. 1.

5.2. Hybrid Method with Reduce Threshold - S (ζ)

We introduce an enhancement to the Modified Hybrid Action Rule Mining approach with a partition threshold ρ by incorporating additional threshold, S, to further refine the rule selection process. Specifically, introduce an additional Monte Carlo-based method to estimate the frequency of occurrence of each generated action rule. The new threshold parameter, S (ζ), serves as a refined filter for extracting reliable action rules from the set of frequent action rules generated.

The Modified Hybrid Action Rule Mining approach that incorporates a partition threshold ρ approach improves scalability by controlling the number of partitions through a partition threshold ρ . Our enhancement further strengthens the reliability of the extracted action rules, especially in high-dimensional or noisy big data environments. The Monte Carlo method validates the stability of the rule across samples, while the threshold S (ζ) filter eliminates low-frequency patterns. The proposed modified hybrid method with reduce threshold S (ζ) ensures that only stable, high-occurring Action Rules are retained, enhancing the effectiveness of the final rule set.

In this study, the Random Sampling technique is utilized to generate 50 independent data samples from the complete student survey dataset. We apply a modified hybrid method to each of these samples to extract Action Rules, using a predefined threshold parameter denoted as ρ (ρ) to control rule generation. As a result, a wide range of comprehensive action rules were derived from multiple randomized data subsets. To evaluate the reliability and frequency of these rules, we implement the Monte Carlo Method, which calculates how often each Action Rule appears

across the 50 generated data samples. 50 independent random samples were selected because they provide a practical balance between computational overhead and statistical stability. Preliminary experimentation showed that increasing the number of samples beyond fifty produced only marginal changes in the frequency distribution of the extracted Action Rules while significantly increasing execution time. Therefore, fifty samples were adopted as an effective compromise between computational efficiency and result reproducibility. This frequency-based analysis enables us to identify which rules are consistently recognized and are, therefore, likely to represent meaningful patterns in the data. To further refine the selection, we introduce a reduced threshold value, $S(\zeta)$. Action Rules that occur more frequently than this threshold is considered substantial rules. For example, if an action rule such as AR4 appears 108 times across all generated samples, it meets the criteria for a valid rule. In contrast, rules that occur less frequently than threshold S are discarded and considered not a potential rule. This approach ensures that high-occurring Action Rules are retained in the final rule set.

Our proposed method, Modified Hybrid Action Rule Mining with Partition Threshold Rho, is presented in Fig. 4 and the proposed methodology is depicted in Fig. 3.

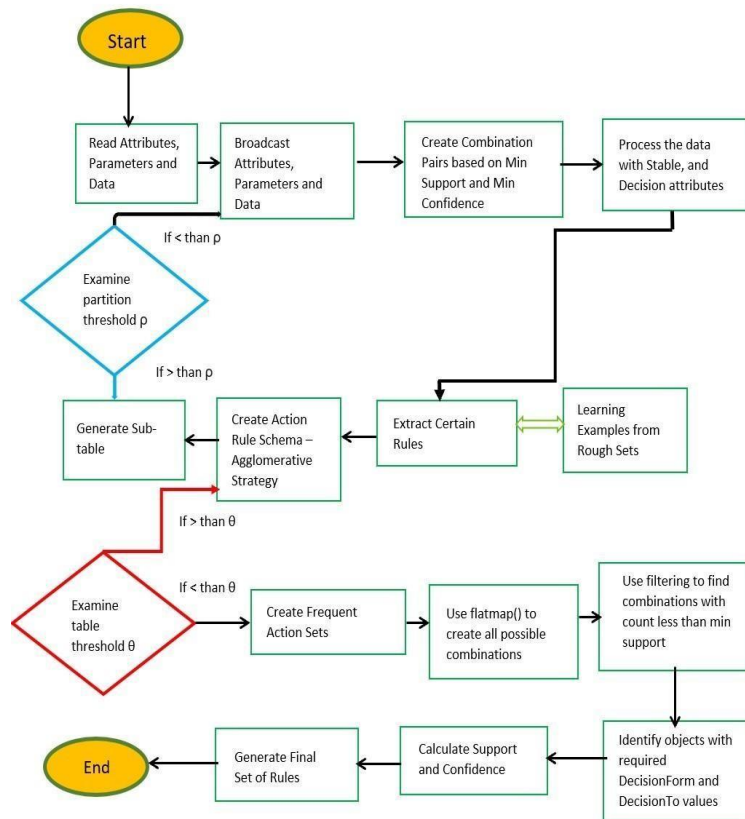


Fig.1. Hybrid Action Rule Mining Algorithm Partition Threshold Rho - Flowchart.

```

1. Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
2.   (where certainRules are provided by algorithm LERS)
3.   for each rule r in certainRules
4.     if consequent (r) equals decisionTo
5.       Form ActionRuleSchema (r)
6.       ARS <- ActionRuleSchema (r)
7.     end if
8.   end for
9.   for each schema in ARS
10.    Identify objects satisfying schema
11.    Form partition
12.    While partition size > Rho p
13.      Form subtable
14.      While subtable size > Theta θ
15.        Divide subtable until subtable < Theta θ
16.      Generate frequent action sets using Apriori
17.      Combine frequent action set to form Action Rules
18.      (such that the frequent action sets satisfy the
19.       decisionFrom -> decisionTo)
20.      Output <- Action Rules
21.    end for

```

Fig.2. Modified Hybrid Action Rule Mining with Partition Threshold Rho - Algorithm.

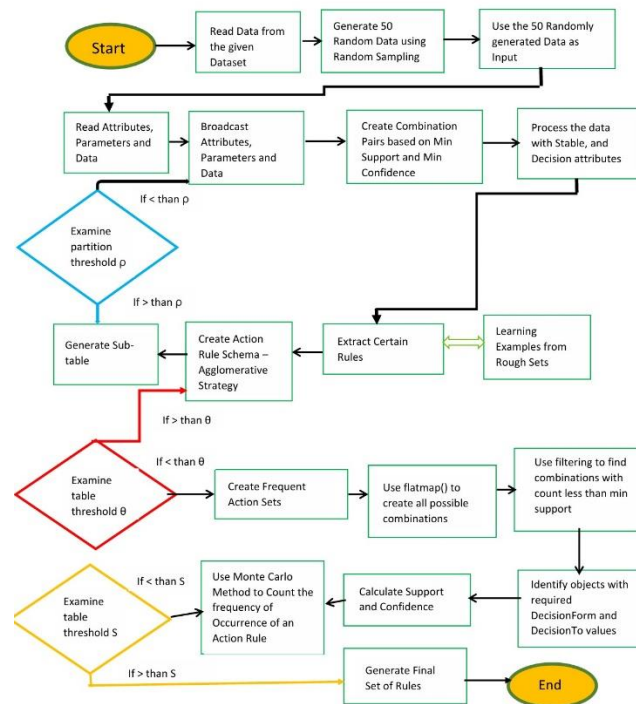


Fig.3. Hybrid Method with Frequency Threshold S Flowchart.

```

1.   Generate 50 Random Sampling
2.   Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
3.   (where certainRules are provided by algorithm LERS)
4.   for each rule r in certainRules
5.       if consequent (r) equals decisionTo
6.           Form ActionRuleSchema (r)
7.           ARS <- ActionRuleSchema (r)
8.       end if
9.   end for
10.  for each schema in ARS
11.      Identify objects satisfying schema
12.      Form partition
13.      While partition size > Rho p
14.          Form subtable
15.          While subtable size > Theta Θ
16.              Divide subtable until subtable < Theta Θ
17.          Generate frequent action sets using Apriori
18.          Combine frequent action set to form Action Rules
19.          (such that the frequent action sets satisfy the
20.              decisionFrom -> decisionTo)
21.      Action Rule generated
22.      Use Monte Carlo Method
23.      Count the Frequency of
24.      Occurrence of an Action Rule
25.      From Partition Table
26.      Generate frequency
27.      While frequency size > Threshold S
28.          Output <- Action Rules
29.  end for

```

Fig.4. Hybrid Method with Frequency Threshold S Algorithm.

6. EXPERIMENTS AND RESULTS

In this work, we use student survey data that focuses on student emotions. We applied the data with three different thresholds and compared the Computational time. See section IV. Dataset Description. We achieve faster computational time through our newly proposed method for the Student Survey.

6.1. Experiment 1 - Modified Hybrid Action Rule Mining with Partition Threshold Rho Implementation in Spark AWS Cluster

We perform this experiment on the Student Survey Data using our proposed Modified Hybrid Action Rule Mining Method, which utilizes Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GB memory along with EBS storage of 64GB. Our proposed method takes 3900 seconds to complete the computation on the replicated Student Survey Data. We experiment with 3 different Threshold values of p : 5, 10, and 15 and θ : 5, 10 and 15. The runtime comparison for different Threshold values for two different thresholds θ and p implemented on Student Survey data is shown in the Table 3 below. For Student Survey data, the threshold value of $p = 5$ and $\theta = 15$ provide optimum performance.

Selected Action Rules generated by this method are shown in Table 14. Action Rule 1 in Table 14 states that when Team Member Responsibility changes from Responsible Members to Helpful

Members and Team Sense of Belonging increases from 3 to 4 and if the Team Work Helped Diversity decreases with support of 31250 and a confidence 100% then the Student Emotion changes from Sadness to Joy. This action rule suggests that when team members become more supportive and students feel a stronger sense of belonging within their team, their emotions are likely to shift from sadness to joy.

Table 4 and 5 shows the runtime comparison of threshold theta and threshold rho each 5,10 and 15 in local machine. Table 10 shows the experiments in local with 10980 rows which we created by replicating the original Student Survey dataset 286 times. Now the size of the dataset is 1.1MB. Table 11 shows the Experiments with 1 million rows which is 26042 times replicated with the original Student Survey in local with size 102MB. Table 12 is the experiments in AWS with 10980 rows. We created this dataset by replicating it with 286 times with the original Student Survey, so the total size of the dataset is 1.1MB. Table 13 shows the experiments in AWS with 1 million rows which we created by replicating the original Student Survey dataset 26042 times. Now the size of the dataset is 102MB.

Table 14 shows the Action Rules generated from Student Survey Data with Threshold Rho 15 and Theta 5 for 10980 rows in AWS, as it shows the optimum runtime. Table 15 the Action Rules generated from Student Survey Data with Threshold Rho 15 and Theta 15 for 1Million rows in AWS, as it shows the optimum runtime. Table 16 shows the Action Rules generated from Student Survey Data with Threshold Rho 5 and Theta 5 for 10980 rows in local, as it shows the optimum runtime. Table 17 shows the Action Rules generated from Student Survey Data with Threshold Rho 10 and Theta 10 for 1Million rows in local, as it shows the optimum runtime.

6.2. Experiment 2 – Modified Hybrid Method with Reduce Threshold – $S(\zeta)$

The new threshold parameter, $S(\zeta)$, serves as a refined filter for extracting reliable action rules from the set of frequent action rules generated. The Modified Hybrid Action Rule Mining approach that incorporates a partition threshold (ρ) approach improves scalability by controlling the number of partitions through a partition threshold (ρ). Our enhancement further strengthens the reliability of the extracted action rules, especially in high-dimensional or noisy big data environments. The Monte Carlo method validates the stability of the rule across samples, while the threshold $S(\zeta)$ filter eliminates low-frequency patterns. The proposed modified hybrid method with reduce threshold $S(\zeta)$ ensures that only stable, high-occurring action rules are retained, enhancing the effectiveness of the final rule set. Table 4 and Table 5 shows the runtime comparison for entire Student Survey dataset and 50 random sampling for the Student Survey dataset using threshold Rho and Theta in the local and AWS- Cluster respectively. Table 6 represents the action rules generated from 50 Random student survey data with threshold 80 in local. The first action rule shows that out of entire records it occurred 128 times which is above the threshold, and it has high occurrence frequency. AR1 in Table 6 describes: if student attitude toward teamwork changes from strong dislike to very positive, then student emotion shifts from Sadness to Joy. The Meta-Actions for the same could be like providing regular feedback checkpoints to identify teamwork issues early or like assigning defined roles within teams (leader, researcher, designer, and presenter).

The experimental results consistently demonstrate that introducing the frequency threshold S substantially reduces the number of unstable Action Rules while preserving highly recurring recommendations. Furthermore, the proposed framework achieves faster execution on AWS than local execution due to distributed computation. These observations indicate that the combination of partition threshold ρ , decision threshold θ , and frequency threshold $S(\zeta)$ effectively balances computational efficiency with rule stability. Increasing the value for frequency threshold $S(\zeta)$

produces fewer but more stable Action Rules, whereas lower values retain a larger number of potentially noisy recommendations. These observations indicate that careful threshold selection is necessary to balance runtime and rule quality. Thus resulting in near-linear time and space complexity while keeping the additional overhead small.

We perform this experiment on the same subset of Student Survey Data with less attributes and lower parameters using our method, which utilizes Amazon Web Services (AWS) cluster 5.36.2 with two primary nodes and 1 task, 4 vCore and 16GB memory along with EBS storage of 64GB. Our proposed method takes 10000 milliseconds and 3000 milliseconds to complete the computation on the 50 Random Sampling Student Survey Data.

Table 10 and Table 11 specify the runtime comparison for 50 random sampling for the 1.1mb Student Survey dataset and 102mb Student Survey dataset respectively and both were run in local. Table 12 and Table 13 specifies the runtime comparison for 50 random sampling for the 1.1mb Student Survey dataset and 102mb Student Survey dataset respectively and both were run in AWS-Cluster. Table 14, 15, 16 and 17 describes the Action Rules derived from Student survey data with threshold Rho in AWS 5.36.0 cluster with 5, 10 and 15 partitions respectively.

Table 3. Threshold values - ρ and θ Run Time for Student Survey Data

Threshold	Threshold	Time in Second
5	5	447
5	10	355
5	15	352
10	5	452
10	10	643
10	15	455
15	5	354
15	10	354
15	15	353

Table 4. Time of Execution Student Survey Dataset (Local Machine)

Threshold Rho	Threshold Theta	Time Execution Entire Dataset(sec)	Time Execution 50 Random Samples(sec)
5	5	68	10
5	10	81	10
5	15	75	10
10	5	65	10
10	10	84	10
10	15	82	10
15	5	69	10
15	10	78	10
15	5	136	10

Table 5. Time of Execution Student Survey Dataset (AWS-EMR – Rho, Theta)

Threshold Rho	Threshold Theta	Time Execution Entire Dataset(s)	Time Execution 50 Random Samples(S)
---------------	-----------------	----------------------------------	-------------------------------------

5	5	23	3
5	10	27	3
5	15	24	3
10	5	21	3
10	10	27	3
10	15	27	3
15	5	23	3
15	10	24	3
15	5	44	3

Table 6. Action Rules generated with Student Survey Data with Threshold S(ζ)

Action Rules for Random Sampling With Threshold S (ζ) = 80
(LikeTeamWork, 1Don't -> 5VeryMuch) => (StudentEmotion, Sadness -> Joy)[Support = 2, Confidence = 100] = 128
(LikeTeamWork, 1Don't -> 4Quiteabit) => (StudentEmotion, Sadness -> Joy)[Support = 2, Confidence = 100] = 300
(TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) Support = 2, Confidence = 100] = 91
(GroupAssignmentBenefit, None -> SocialLearning) => StudentEmotion, Sadness -> Joy) Support = 2, Confidence = 100] = 82
(GroupAssignmentBenefit, None -> SharedKnowledge) => StudentEmotion, Sadness -> Joy) Support = 2, Confidence = 100] = 154

Table 7. Student Survey Data – In AWS – With Threshold Rho 5

Enhance Emotion- Sadness To Joy
(GroupAssignmentBenefit, None -> AllofThem) ^ (LikeTeamWork, 2Verylittle -> 4Quiteabit) ^ (TeamFormation, 2BelowAverage -> 3Average) ^ (TeamMemberResponsibility, IrresponsibleMembers -> HelpfulMembers) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]
(LikeTeamWork, 4Quiteabit -> 5VeryMuch) ^ (NumberofTeamMembers, 10orMore -> 5to7) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]
NumberofTeamMembers, 8to10 -> 10orMore) ^ (TeamFormation, 4Perfect -> 3Average) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 72, Old Confidence: 100%, New Confidence: 100%]
(NumberofTeamMembers, 5to7 -> 8to10) ^ (TeamSenseofBelonging, 4CompleteSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]
(NumberofTeamMembers, 10orMore -> 5to7) ^ (TeamFormation, 3Average -> 4Perfect) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]

Table 8. Student Survey Data – In AWS – With Threshold Rho 10

Enhance Emotion- Sadness To Joy
(NumberofTeamMembers, 4orLess -> 10orMore) ^ (TeamFormation, 3Average -> 4Perfect) ^ (TeamWorkHelpedDiversity, 2Occasionally -> 4VeryOften) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]

(TeamFormation, 4Perfect -> 3Average) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 96, Old Confidence: 100%, New Confidence: 100%]
(GroupAssignmentBenefit, None -> SocialLearning) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]
(NumberOfTeamMembers, 8to10 -> 10orMore) ^ (TeamSenseofBelonging, 3AverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 72, Old Confidence: 100%, New Confidence: 100%]
(NumberOfTeamMembers, 5to7 -> 8to10) ^ (TeamFormation, 2BelowAverage -> 3Average) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]

Table 9. Student Survey Data – In AWS – With Threshold Rho 15

Enhance Emotion- Sadness To Joy
(TeamMemberResponsibility, ResponsibleMembers -> HelpfulMembers) ^ (TeamSenseofBelonging, 3AverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 3Often -> 2Occasionally) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]
(NumberOfTeamMembers, 8to10 -> 5to7) ^ (TeamMemberResponsibility, TechnicallyEffectiveMembers -> ResponsibleMembers) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 2Occasionally) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]
(NumberOfTeamMembers, 5to7 -> 10orMore) ^ (TeamMemberResponsibility, IrresponsibleMembers -> TechnicallyEffectiveMembers) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]
(NumberOfTeamMembers, 10orMore -> 5to7) ^ (TeamMemberResponsibility, FriendlyMembers -> HelpfulMembers) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 48, Old Confidence: 100%, New Confidence: 100%]
(GroupAssignmentBenefit, None -> SharedKnowledge) ^ (TeamMemberResponsibility, IrresponsibleMembers -> ResponsibleMembers) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]

Table 10. Student Survey Data – In Local – With Threshold Rho and Theta

Rho	Theta	No. of Rows	Time of Execution (Local)	Replication factor	Size
5	5	10980	2771 milliseconds	286 times	1.1mb
5	10	10980	3658 milliseconds	286 times	1.1mb
5	15	10980	3993 milliseconds	286 times	1.1mb
10	5	10980	4353 milliseconds	286 times	1.1mb

10	10	10980	5648 milliseconds	286 times	1.1mb
10	15	10980	6697 milliseconds	286 times	1.1mb
15	5	10980	3233 milliseconds	286 times	1.1mb
15	10	10980	3923 milliseconds	286 times	1.1mb
15	15	10980	4341 milliseconds	286 times	1.1mb

Table 11. Student Survey Data – In Local– With Threshold Rho and Theta

Rho	Theta	No. of Rows	Time Execution of (Local)	Replication factor	Size
5	5	1 Million	5935 milliseconds	26042 times	102mb
5	10	1 Million	5816 milliseconds	26042 times	102mb
5	15	1 Million	5850 milliseconds	26042 times	102mb
10	5	1 Million	5689 milliseconds	26042 times	102mb
10	10	1 Million	5679 milliseconds	26042 times	102mb
10	15	1 Million	5763 milliseconds	26042 times	102mb
15	5	1 Million	5926 milliseconds	26042 times	102mb
15	10	1 Million	6245 milliseconds	26042 times	102mb
15	15	1 Million	6022 milliseconds	26042 times	102mb

Table 12. Student Survey Data – In AWS – With Threshold Rho and Theta

Rho	Theta	No. of Rows	Time of Execution (AWS)	Replication factor	Size
5	5	10980	19915 milliseconds	286 times	1.1mb
5	10	10980	21503 milliseconds	286 times	1.1mb
5	15	10980	21944 milliseconds	286 times	1.1mb
10	5	10980	27427 milliseconds	286 times	1.1mb
10	10	10980	30225 milliseconds	286 times	1.1mb
10	15	10980	30701 milliseconds	286 times	1.1mb
15	5	10980	19629 milliseconds	286 times	1.1mb

15	10	10980	20300 milliseconds	286 times	1.1mb
15	15	10980	23215 milliseconds	286 times	1.1mb

Table 13. Student Survey Data – In AWS – With Threshold Rho and Theta

Rho	Theta	No. of Rows	Time of Execution (AWS)	Replication factor	Size
5	5	1 million	39908 milliseconds	26048 times	102mb
5	10	1 million	36013 milliseconds	26048 times	102mb
5	15	1 million	35242 milliseconds	26048 times	102mb
10	5	1 million	35153 milliseconds	26048 times	102mb
10	10	1 million	35421 milliseconds	26048 times	102mb
10	15	1 million	35439 milliseconds	26048 times	102mb
15	5	1 million	37637 milliseconds	26048 times	102mb
15	10	1 million	35344 milliseconds	26048 times	102mb
15	15	1 million	34304 milliseconds	26048 times	102mb

Table 14. Student Survey Data – In AWS – With Threshold Rho 15 and Theta 5 for 10980 rows

AR1: (GroupAssignmentBenefit, None -> SharedKnowledge) ^ (TeamMemberResponsibility, UnHelpfulMembers -> FriendlyMembers) ^ (TeamWorkHelpedDiversity, 3Often -> 4VeryOften) => (StudentEmotion, Sadness -> Joy) [Support: 16, Old Confidence: 100%, New Confidence: 100%]
AR2: (GroupAssignmentBenefit, None -> SocialLearning) ^ (TeamFormation, 1Imperfect -> 4Perfect) ^ (TeamMemberResponsibility, FriendlyMembers -> HelpfulMembers) => (StudentEmotion, Sadness -> Joy) [Support: 8, Old Confidence: 100%, New Confidence: 100%]
AR3: (GroupAssignmentBenefit, SharedKnowledge -> SocialLearning) ^ (LikeTeamWork, 1Don't -> 5VeryMuch) ^ (NumberOfTeamMembers, 4orLess -> 5to7) => (StudentEmotion, Sadness -> Joy) [Support: 16, Old Confidence: 100%, New Confidence: 100%]
AR4: (GroupAssignmentBenefit, None -> AllofThem) ^ (NumberOfTeamMembers, 8to10 -> 10orMore) ^ (TeamSenseofBelonging, 1FeelAlienated -> 3AverageSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 40, Old Confidence: 100%, New Confidence: 100%]
AR5: (LikeTeamWork, 5VeryMuch -> 4Quiteabit) ^ (NumberOfTeamMembers, 5to7 -> 8to10) ^ (TeamMemberResponsibility, UnHelpfulMembers -> FriendlyMembers) => (StudentEmotion, Sadness -> Joy) [Support: 24, Old Confidence: 100%, New Confidence: 100%]

Table 15. Student Survey Data – In AWS – With Threshold Rho 15 and Theta 15 for 1million rows

AR1: (TeamMemberResponsibility, ResponsibleMembers -> HelpfulMembers) ^ (TeamSenseofBelonging, 3AverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 3Often -> 2Occasionally) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]
AR2: (NumberOfTeamMembers, 8to10 -> 10orMore) ^ (TeamMemberResponsibility, HelpfulMembers -> TechnicallyEffectiveMembers) ^ (TeamWorkHelpedDiversity, 3Often -> 4VeryOften) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]
AR3: (GroupAssignmentBenefit, None -> SocialLearning) ^ (NumberOfTeamMembers, 5to7 -> 8to10) => (StudentEmotion, Sadness -> Joy) [Support: 62500, Old Confidence: 100%, New Confidence: 100%]
AR4: (LikeTeamWork, 2Verylittle -> 4Quiteabit) ^ (TeamFormation, 2BelowAverage -> 3Average) ^ (TeamMemberResponsibility, IrresponsibleMembers -> HelpfulMembers) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 62500, Old Confidence: 100%, New Confidence: 100%]
AR5: (TeamMemberResponsibility, FriendlyMembers -> HelpfulMembers) ^ (TeamSenseofBelonging, 4CompleteSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]

Table 16. Student Survey Data – In Local– With Threshold Rho 5 and Theta 5 for 10980 rows

AR1: GroupAssignmentBenefit, EliminateStress -> SocialLearning) ^ (LikeTeamWork, 3Somewhat -> 2Verylittle) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 4, Old Confidence: 100%, New Confidence: 100%]
AR2: (GroupAssignmentBenefit, SharedKnowledge -> SocialLearning) ^ (LikeTeamWork, 1Don't -> 5VeryMuch) ^ (TeamWorkHelpedDiversity, 3Often -> 2Occasionally) => (StudentEmotion, Sadness -> Joy) [Support: 12, Old Confidence: 100%, New Confidence: 100%]
AR3: (GroupAssignmentBenefit, None -> EliminateStress) ^ (TeamMemberResponsibility, TechnicallyIneffectiveMembers -> ResponsibleMembers) ^ (TeamWorkHelpedDiversity, 4VeryOften -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 20, Old Confidence: 100%, New Confidence: 100%]
AR4: (LikeTeamWork, 2Verylittle -> 5VeryMuch) ^ (NumberOfTeamMembers, 5to7 -> 10orMore) ^ (TeamMemberResponsibility, IrresponsibleMembers -> HelpfulMembers) ^ (TeamWorkHelpedDiversity, 2Occasionally -> 3Often) => (StudentEmotion, Sadness -> Joy) [Support: 4, Old Confidence: 100%, New Confidence: 100%]
AR5: (LikeTeamWork, 4Quiteabit -> 5VeryMuch) ^ (NumberOfTeamMembers, 4orLess -> 8to10) ^ (TeamFormation, 1Imperfect -> 2BelowAverage) => (StudentEmotion, Sadness -> Joy) [Support: 20, Old Confidence: 100%, New Confidence: 100%]

Table 17. Student Survey Data – In Local– With Threshold Rho 10 and Theta 10 for 1Million rows

AR41: (TeamMemberResponsibility, ResponsibleMembers -> HelpfulMembers) ^ (TeamSenseofBelonging, 3AverageSenseofBelongingtotheTeam -> 4CompleteSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 3Often -> 2Occasionally) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]
--

AR42: (NumberOfTeamMembers, 5to7 -> 10orMore) ^ (TeamMemberResponsibility, IrresponsibleMembers -> FriendlyMembers) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) => (StudentEmotion, Sadness -> Joy) [Support: 62500, Old Confidence: 100%, New Confidence: 100%]
AR43: (LikeTeamWork, 2Verylittle -> 4Quiteabit) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 2Occasionally -> 4VeryOften) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]
AR44: (TeamMemberResponsibility, IrresponsibleMembers -> FriendlyMembers) ^ (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam -> 3AverageSenseofBelongingtotheTeam) ^ (TeamWorkHelpedDiversity, 2Occasionally -> 4VeryOften) => (StudentEmotion, Sadness -> Joy) [Support: 31250, Old Confidence: 100%, New Confidence: 100%]
AR45: (GroupAssignmentBenefit, SharedKnowledge -> AllofThem) ^ (TeamFormation, 3Average -> 4Perfect) ^ (TeamMemberResponsibility, HelpfulMembers -> ResponsibleMembers) => (StudentEmotion, Sadness -> Joy) [Support: 105861, Old Confidence: 83%, New Confidence: 83%]

6. CONCLUSION

This paper introduced a novel and intelligent approach to data partitioning that significantly enhances both efficiency and the discovery of actionable knowledge. Our methods, which utilizes a frequency threshold S (ς) along with partition threshold Rho (ρ) and threshold θ (θ), to horizontally and vertically divide large datasets, effectively reduces computational complexity and expense. A key benefit of our approach is its superior processing time when compared to existing random distribution methods, making it a more practical solution for large-scale data analysis. Since we performed our experiments over replicated datasets, the replication effectively increases data volume for evaluating distributed computation, it does not introduce additional diversity or complexity that naturally exists in real-world big data. Furthermore, the analysis of the cost and feasibility of each meta-action empowers users to make informed, goal-oriented decisions, where we got an idea about the expensiveness of an action rule. Using the Monte Carlo method on our action rules generated by 50 randomly generated data, we were able to determine the frequency of the most common rules, identifying them as the most effective. While we acknowledge that there may be a slight decrease in the quality of the rules generated, the overall efficiency and practical application of our method make it a powerful tool for data analysis and decision-making. Besides execution time, the extracted Action Rules consistently achieved high Support and 100% Confidence across the presented experiments. Furthermore, the frequency threshold S (ς) improves rule stability by selecting Action Rules that repeatedly appear across independent random samples. The application of this method across diverse domains, including the Student Survey Dataset, highlights its adaptability and potential to provide tangible insights for decision-making in education.

7. FUTURE WORK

Future work will focus on improving the algorithm to reduce the trade-off between rule quality and computational efficiency. We will explore advanced heuristics, such as an adaptive ρ value that adjusts based on data characteristics, to ensure more relevant and robust results. We also plan to evaluate the method on large, real-world datasets to test scalability and performance across industries to further demonstrate scalability under realistic heterogeneous data distributions rather than replicated datasets. Additionally, we will integrate our approach with other action rule techniques, such as GA-based Action Rule mining, and machine learning methods (e.g., Random

Forest or Gradient Boosting) to automate meta-action selection. This will strengthen its impact in data analytics.

REFERENCES

- [1] Acheampong, F., Wenyu, C., Nunoo-Mensah, H. ("Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, 2020. <https://doi.org/10.1002/eng2.12189>
- [2] Bagavathi, A., Mummoju, P., Tarnowska, K., Tzacheva, A.A., Ras, Z.W., "SARGS method for distributed actionable pattern mining using spark," in *IEEE International Conference on Big Data (Big Data)*, vol.12, pp. 4272–4281, 2017. <https://doi.org/10.1109/BigData.2017.8258481>
- [3] Bagavathi, A., Rao, V., Tzacheva, A.A., "Data distribution method for scalable actionable pattern mining," in *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*. ACM, p. 3, 2018. <https://doi.org/10.1145/3178158.3178162>
- [4] Baker, R.S., Inventado, P.S. "Educational data mining and learning analytics," in *Learning analytics*. Springer, 2014, pp. 61–75. https://doi.org/10.1007/978-1-4614-3305-7_4
- [5] Benedict, A.C., Ras, Z.W., Cylulko, P., Gladyszewska-Cylulko, J., Enhancing typhlo music therapy with personalized action rules: A data-driven approach", *Information*, MDPI, Vol. 16, Issue 8, 2025 <https://doi.org/10.3390/info16080666>
- [6] Benedict, A., Ras, Z.W. Towards scalable action rule discovery: a structured vertical partitioning method. *Journal of Intelligent Information Systems* 64, 5–35 (2026). <https://doi.org/10.1007/s10844-025-00961-5>
- [7] Dean, J., Ghemawat, S. (2008). "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008. <https://doi.org/10.1145/1327452.1327492>
- [8] Dobbs, T., Ras, Z.W., On art authentication and the Rijksmuseum challenge: A residual neural network approach, *Expert Systems with Applications*, Vol. 200, Elsevier, 2022. <https://doi.org/10.1016/j.eswa.2022.116933>
- [9] Gonzalez, J.E., Xin, R.S., Dave, A., Crankshaw, D., Franklin, M.J., Stoica, I., "Graphx: Graph processing in a distributed dataflow framework." in the *Proceedings of 11th USENIX Symposium on Operating System Design and Implementation*, vol. 14, pp. 599–613, 08 2014.
- [10] Hashemi, S., Shamsinejad, P., "GA2RM: A GA-based action rule mining method." *International Journal of Computational Intelligence and Applications*, 20(2), 2150012, 2021. <https://doi.org/10.1142/S146902682150012X>
- [11] Im, S., Ras, Z.W., "Action rule extraction from a decision table: Ared," in *International Symposium on Methodologies for Intelligent Systems*. Springer, 2008, pp. 160–168. https://doi.org/10.1007/978-3-540-68125-0_18
- [12] Khder, M., Abu-AlSondos, I., Bahar, Y., "The impact of implementing data mining in business intelligence." *International Journal of Entrepreneurship*, vol. 25, pp. 1–9, 2021. <https://www.abacademies.org/articles/the-impact-of-implementing-data-mining-in-business-intelligence-11473.html>
- [13] Koh, H.C., Tan, G., "Data mining applications in healthcare," *Journal of healthcare information management*, vol. 19. <https://pubmed.ncbi.nlm.nih.gov/15869215/>
- [14] Kuang, J., Daniel, A., Johnston, J., Ras, Z.W. (2014). "Hierarchically structured recommender system for improving nps of a company," in *International Conference on Rough Sets and Current Trends in Computing*. Springer, 2014, pp. 347–35. https://doi.org/10.1007/978-3-319-08644-6_36
- [15] Kuang, J., Ras, Z.W., Daniel, A., "Personalized meta-action mining for NPS improvement." In *Foundations of Intelligent Systems*. Springer, pp. 79–87. https://doi.org/10.1007/978-3-319-08338-4_9
- [16] Machova, K., Szaboova, M., Paralic, J., Micko, J., "Detection of emotion by text analysis using machine learning," *Frontiers in Psychology*, 2023. <https://doi.org/10.3389/fpsyg.2023.1190326>
- [17] Mardini, M.T., Ras, Z.W., Extraction of actionable knowledge to reduce hospital readmissions through patients' personalization, *Information Sciences Journal*, Elsevier, Vol. 485, 2019, 1-17 <https://doi.org/10.1016/j.ins.2019.02.006>
- [18] Nandwani, P., Verma, R. "A review on sentiment analysis and emotion detection from text," *Social Network Analysis and Mining*, 2021 . <https://doi.org/10.1007/s13278-021-00776-6>

- [19] Powell, L., Gelich, A., Ras, Z.W., How to raise artwork prices using action rules, personalization, and artwork visual features, *Journal of Intelligent Information Systems*, Vol 57, No. 3, Springer, 2021, 583-599 <https://doi.org/10.1007/s10844-021-00660-x>
- [20] Ragunathan, S. "Emotional Detection from Text Using NLP (Natural Language Processing) With Prediction Probability," 2024. https://www.researchgate.net/publication/379190401_Emotional_Detection_from_Text_Using_NLP_Natural_Language_Processing_With_Prediction_Probability
- [21] Vinay, S.B., "Natural language processing for legal documentation in indian languages," *International Journal of Business and Data Analytics (IJBDA)*, vol. 1, pp. 1–15, 01 2024. https://iaeme.com/MasterAdmin/Journal_uploads/IJNLP/VOLUME_2_ISSUE_1/IJNLP_02_01_001.pdf
- [22] Ranganathan, J., Irudayaraj, A.S., Bagavathi, A., Tzacheva, A.A. "Actionable pattern discovery for sentiment analysis on twitter data in clustered environment," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–15, 2018. <https://doi.org/10.3233/JIFS-169472>
- [23] Ras, Z.W., Dardzinska, A., "Action rules discovery: A new simplified strategy." pp. 445–453, 2006. https://doi.org/10.1007/11875604_51
- [24] Ras, Z.W., Dardzinska, A., Tsay, L.S., Wasyluk, H., "Association action rules," in *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*. IEEE, 2008, pp. 283–290. <https://doi.org/10.1109/ICDMW.2008.45>
- [25] Ras, Z.W., Tzacheva, A.A., "In search for action rules of the lowest cost," in *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer, 2005, pp. 261–272. https://doi.org/10.1007/0-387-28391-9_18
- [26] Ras, Z.W., Wyrzykowska, E., Wasyluk, H. "Aras: Action rules discovery based on agglomerative strategy," in *International Workshop on Mining Complex Data*. Springer, 2007, pp. 196–208. https://doi.org/10.1007/978-3-540-68416-9_16
- [27] Ras, Z.W., Wieczorkowska, A., "Action-Rules: How to increase profits of a company," in *Principles of Data Mining and Knowledge Discovery (PKDD 2000)*. LNCS, vol. 1910. Springer., 2000. https://doi.org/10.1007/3-540-45372-5_70
- [28] Shmueli, G., Bruce, P.C., Yahav, I., Patel, N.R., Lichtendahl Jr, K.C., "Data mining for business analytics: concepts, techniques, and applications." In R. John Wiley & Sons, 2017. <https://www.wiley.com/engb/Data+Mining+for+Business+Analytics%3A+Concepts%2C+Technique+s%2C+and+Applications+in+R-p-9781118879337>
- [29] Strack, B., DeShazo, J.P., Gennings, C., Olmo, J.L., Ventura, S., Cios, K.J., Clore, J.N., "Impact of hospital readmission rates: analysis of 70,000 clinical database patient records," *BioMed research international*, vol. 2014. <https://doi.org/10.1155/2014/781670>
- [30] Su, P., Li, D., Su, K., "An expected utility-based approach for mining action rules," in *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, ser. ISI-KDD '12. New York, NY, USA: ACM, 2012, pp. 9:1–9:4. [Online]. <http://doi.acm.org/10.1145/2331791.2331800>
- [31] Tarnowska, K., Ras, Z.W., Daniel, L., Recommender system for improving customer loyalty, *Studies in Big Data*, Vol. 55, Springer, 2020. <https://doi.org/10.1007/978-3-030-13438-9>
- [32] Tarnowska, K., Ras, Z.W., Jastreboff, P.J., A data-driven approach to clinical decision support in tinnitus retraining therapy, *Frontiers in Neuroinformatics*, Switzerland, September 28, 2022 <https://doi.org/10.3389/fninf.2022.934433>
- [33] Touati, H., Kuang, J., Hajja, A., Ras, Z.W. "Personalized Action Rules for Side Effects Object Grouping", in *International Journal of Intelligent Systems*, 2013. <https://doi.org/10.4236/ijis.2013.31A004>
- [34] Tsay, L.S., Ras, Z.W. "Action rules discovery: system dear2, method and experiments," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 119–128, 2005. <https://doi.org/10.1080/09528130412331314586>
- [35] Tzacheva, A.A., Easwaran, A. "Modified hybrid scalable action rule mining for emotion detection in student survey data," in *International Conference on Data Mining, Big Data, Database and Data Technologies*, pp. 13–19, 01 2022.
- [36] Tzacheva, A.A., Ras, Z.W., "Action rules mining," in *International Journal of Intelligent System*, vol. 20, pp. 719–736, 01 2005. <https://doi.org/10.1002/int.20043>
- [37] Tzacheva, A.A., Sankar, C.C., Shankar, R., "Support confidence and utility of action rules triggered by meta-actions," in *proceedings of 2016 IEEE International Conference on Knowledge Engineering*

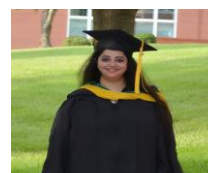
- and Applications, ser. ICKEA 2016. IEEE Computer Society, 2016. <https://ijcionline.com/paper/13/13624ijci07.pdf>
- [38] Tzacheva, A.A., Ranganathan, J., “pattern discovery from student feedback: Identifying factors to improve student emotions in learning”, vol. 14, no. 12. World Academy of Science, Engineering and Technology, 2020, pp. 1260 – 1265. [Online]. <https://publications.waset.org/vol/168>
- [39] Tzacheva, A.A., Chatterjee, S., “ Actionable Pattern Discovery for Emotion Detection in Big Data in Education and Business” in International Conference on AI, Machine Learning and Data Science (AIMDS), Vol.13, No 6, pp. 93 - 117 , 2024,Doi: 10.5121/ijci.2024.130607 <https://ijcionline.com/paper/13/13624ijci07.pdf>
- [40] Vinay, S.B., “Identifying research trends using text mining techniques: A systematic review,” *International Journal of Data Mining and Knowledge Discovery (IJDMKD)*, vol. 1, pp. 1–11, 01 2024.https://iaeme.com/MasterAdmin/Journal_uploads/IJDMKD/VOLUME_1_ISSUE_1/IJDMKD_01_01_001.pdf
- [41] Vinay, S.B. , “AI and machine learning integration with AWS Sage Maker: current trends and future prospects,” *International Journal of Artificial Intelligence Tools (IJAIT)*, vol. 1, pp. 1–24, 01 2024. [https://iaeme.com/MasterAdmin/Journal_uploads/IJAIT/VOLUME_1_ISSUE_1/IJAIT_01_01_001.p df](https://iaeme.com/MasterAdmin/Journal_uploads/IJAIT/VOLUME_1_ISSUE_1/IJAIT_01_01_001.pdf)
- [42] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., Stoica, I., “Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing,” in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI’12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2.
- [43] Z. Cui, W. Chen, Y. He, and Y. Chen, 2015. “Optimal Action Extraction for Random Forests and Boosted Trees”, In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. Association for Computing Machinery, New York, NY, USA, 179–188, 2015,Doi: <https://doi.org/10.1145/2783258.2783281>

AUTHORS

Angelina A. Tzacheva is currently a professor, and Ph.D. Program Chair, Department of Computer Science and Information Technology, Westcliff University, Irvine, CA 92612. Her research interests include Data Mining, Knowledge Discovery in databases distributed knowledge systems, medical imaging, multimedia databases, and bioinformatics.



Sanchari Chatterjee received MS degree from University of North Carolina, Charlotte, in Computer Science. Currently, she is pursuing her PhD in Computer Science from the University of North Carolina, Charlotte. Her research interests include Data Mining, Knowledge discovery in databases, Machine Learning, Artificial Intelligence.



Rajia Shareen Shaik received an MS degree from University of North Carolina, Charlotte, in Computer Science. Currently, she is working as a Software Engineer at Walmart, specializing in event-driven systems in the supply chain domain. Her research interests include Data Mining, Knowledge discovery in databases, Machine Learning, Artificial Intelligence.



Shiva Sai Praneeth Chakinala is a Software Engineer at Walmart, specializing in building scalable and efficient data-driven applications. His work focuses on software architecture, automation, and integrating advanced analytics into enterprise systems. His research interests lie in Data Science and Machine Learning.

