Modified Hybrid Scalable Action Rule Mining for Emotion Detection in Student Survey Data

Angelina Tzacheva
Department of Computer science
University of North Carolina at Charlotte, Charlotte, North Carolina, USA, 28223
Email: aatzache@uncc.edu
Akshaya Easwaran
Department of Information Technology
University of North Carolina at Charlotte, Charlotte, North Carolina, USA, 28223
Email: aeaswar1@uncc.edu

Abstract—Everyday several tons of data are generated by Social Media, Education sectors. Mining this data can provide a lot of meaningful insights on how to improve user experience in social media and how to improve student learning experience. Action Rule Mining is a method that can extract such actionable patterns from various datasets. Action Rules provide actionable suggestions on how to change the state of an object from an existing state to a desired state for the benefit of the user. There are two major frameworks in the literature of Action Rule mining namely Rule-Based method where the extraction of Action Rules is dependent on the pre-processing step of classification rule discovery and Object-Based method where it extracts the Action Rules directly from the database without the use of classification rules. Hybrid Action rule mining approach combines both these frameworks and generates complete set of Action Rules. The hybrid approach shows significant improvement in terms computational performance over the Rule-Based and Object-Based approach. In this work we propose a Modified Hybrid Action rule approach which further improves the computational performance with big datasets.

Keywords—Actionable Patterns, Action Rules, Emotion Detection, Data Mining, Rule-Based, Object-Based.

I. Introduction

THE spectacular growth and development of social media platforms like Twitter, Facebook, Instagram and other similar ones generate very large amount of unstructured data. Product and Service companies collect customer satisfaction data through surveys. Educational institutions collect student data on their learning experience in the courses. Analysis of this data may provide meaningful perception into the opinion and feelings of people towards certain subjects, products or services.

Opinion mining is the process of extracting valuable data from opinions of people to assess their feelings and thoughts on various topics.Understanding what people like or dislike is critical for making informed business and political decisions.

Action Rule mining method helps to extract Action Rules. These Action Rules extracted from a decision system suggest possible transition of data from an existing state to a desired state [1]. The formal definition of Action Rule [1] is defined as in “1”.

$$[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\phi \rightarrow \psi) \tag{1}$$

Where, ω represents the conjunction of fixed condition features that are shared by both groups, $(\alpha \rightarrow \beta)$ represents changes in flexible attributes, and $(\phi \rightarrow \psi)$ represents the desired change in the decision attribute such that it benefits the user.

In this work, we focus on Opinion Mining from Text to suggest Actionable Recommendations. The Actionable Patterns may suggest ways to alter the user’s sentiment or emotion to a more positive or desirable state. Action Rule Mining literature mainly consists of two major frameworks namely, Rule-Based approach and Object-Based approach. The third approach known as Hybrid Action Rule mining combines the above two frameworks to work well with large datasets. In this work we propose a Modified Hybrid Action Rule mining approach that further improves the computational performance. We apply our method to Student Survey Education data. We aim to suggest ways to improve the Teaching Methods and Student Learning. We implement and test our system in Scalable Environment with BigData using the Apache Spark platform.

II. Related Work

There are two major Action Rule mining frameworks in Data Mining literature. They are the Rule-Based Action Rule mining and Object-Based Action Rule mining frameworks. Hybrid Action Rule mining is a new approach that combines the above Rule-Based and Object-Based approaches to generate the Action Rules much quickly.

A. Rule-Based Action Rule Mining

Rule-Based Action Rule mining approach extracts intermediate classification rules using Learning Based on Rough Sets (LERS) approach [2] and then uses it to extract classification rules from complete information system or using the Extracting Rules from Incomplete Decision (ERID) approach [3]. Rule-Based approach is further sub-divided into methods that generate Action Rules from certain pairs of classification rules like Discovering Extended Action Rules (DEAR) method [4], [5] and methods that generate Action Rules from single classification rule like the Action Rules Based on Agglomerative Strategy method [6].

B. Object-Based Action Rule Mining

Object-Based Action Rule mining approach extracts Action Rules directly from database using the Action Rule Extraction from Decision Table (ARED) method [7] or using the Association Action Rules [1] without extracting intermediate classification rules.

III. Background

The following section provides background information related to Action Rule mining.

Information system given in table I is a system $\mathbb{Z} = (\mathbb{X}, \mathbb{M}, \mathbb{V})$, where $\mathbb{X}$ is set of objects $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ in the system; $\mathbb{M}$ is non-empty finite set of attributes $\{A, B, C, E, F, G, D\}$; $\mathbb{V}$ is the domain of attributes in $\mathbb{M}$, for instance the domain of attribute B in the system $\mathbb{Z}$ is $\{B_1, B_2, B_3\}$.

TABLE I
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 I becomes a Decision System if the attributes $\mathbb{M}$ are classified into flexible attributes M_{fl} , stable attributes M_{st} and decision attributes d, $\mathbb{M}$ = $(M_{st}, M_{fl}, \{d\})$. From table I M_{st} = $\{A, B, C\}$, M_{fl} = $\{E, F, G\}$, and d = D.

A. Action Rule

The expression $\mathrm{r} = [\mathrm{t}_1 \rightarrow \mathrm{t}_2]$ is an Action Rule where, t_1 is an action term and t_2 is an atomic action term. The following is an example Action Rule from table I. $[\mathrm{B}_1 \wedge \mathrm{C}_1 \wedge (\mathrm{F}, \mathrm{F}_3 \rightarrow \mathrm{F}_1) \wedge (\mathrm{G}, \rightarrow \mathrm{G}_1) \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1)$.

B. Support and Confidence

Support and Confidence of rule r is given as below:

* $sup(r) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}$.
* $conf(r) = \frac{card(Y_1 \cap Z_1)}{card(Y_1)} \cdot \frac{card(Y_2 \cap Z_2)}{card(Y_2)}$.
* $card(Y_1) \neq 0$, $card(Y_2) \neq 0$, $card(Y_1 \cap Z_1) \neq 0$, $card(Y_2 \cap Z_2) \neq 0$.
* $conf(r) = 0$ otherwise.

C. Learning from Rough Sets (LERS)

LERS [2] method constructs rules with a conditional part of the length $k + 1$ after all rules with a conditional part of length k have been constructed. This method finds the certain rules and possible rules representing the decision attribute in terms of other attributes in the system. Let us consider table I as a Decision System with the following attributes $\mathbb{M}$ = $(M_{st}, M_{fl}, \{d\})$, where M_{st} = $\{A, B, C\}$, M_{fl} = $\{E, F, G\}$, and d = D.

The list of certain rules and possible rules that LERS method generates from the table I is as follows.

* Certain Rules
  - $E_1 \rightarrow D_1$
  - $G_3 \rightarrow D_2$
  - $F_3 \rightarrow D_2$
  - $E_3 \rightarrow D_2$
  - $B_2 \rightarrow D_2$
  - $B_3 \rightarrow D_2$
  - $A_3 \rightarrow D_2$

  - $A_2 \wedge G_1 \rightarrow D_2$
  - $A_1 \wedge E_2 \rightarrow D_2$
  - $A_2 \wedge C_1 \rightarrow D_2$
  - $A_1 \wedge C_2 \rightarrow D_2$
  - $E_2 \wedge C_1 \rightarrow D_2$
  - $G_2 \wedge B_1 \rightarrow D_3$
  - $G_1 \wedge E_2 \rightarrow D_2$
  - $G_1 \wedge C_2 \rightarrow D_2$
  - $A_1 \wedge C_1 \wedge B_1 \rightarrow D_1$
  - $A_2 \wedge B_1 \wedge C_2 \rightarrow D_3$

  - $A_2 \wedge E_2 \wedge B_1 \wedge C_2 \rightarrow D_3$
  - $A_2 \wedge F_2 \wedge E_2 \wedge B_1 \rightarrow D_3$
  - $A_1 \wedge G_1 \wedge C_1 \wedge B_1 \rightarrow D_1$
  - $A_2 \wedge F_2 \wedge B_1 \wedge C_2 \rightarrow D_3$
  - $A_1 \wedge F_2 \wedge C_1 \wedge B_1 \rightarrow D_1$
  - $G_1 \wedge F_2 \wedge C_1 \wedge B_1 \rightarrow D_1$
* Possible Rules
  - $A_2 \wedge G_2 \wedge F_2 \wedge E_2 \wedge C_2 \rightarrow D_1$
  - $A_2 \wedge G_2 \wedge F_2 \wedge E_2 \wedge C_2 \rightarrow D_2$
  - $A_2 \wedge G_2 \wedge F_2 \wedge E_2 \wedge C_2 \rightarrow D_1$

D. Action Rules Based on Agglomerative Strategy (ARoGs)

ARoGs [6] uses LERS method [2] to extract Action Rules. By using LERS as the intermediate step the overall complexity is reduced when compared to DEAR [4], [5] method.

The sample Action Rules that ARoGs method generates from table I where the decision value is changed from $D_2 \rightarrow D_1$ is shown below.

ARoGs method uses the certain rules extracted by employing the LERS strategy and generates the Action Rule schema. For each of the Action Rule schema the Action Rules are constructed. Let us consider the following classification rule "2".

$$G_1 \wedge F_2 \wedge C_1 \wedge B_1 \rightarrow D_1 \tag{2}$$

The Action Rule schema associated with the above equation "(2)" for reclassification $D_2 \rightarrow D_1$ is given in equation "(3)" and the corresponding Action Rule is given in equation "4".

$$[\mathrm{B}_1 \wedge \mathrm{C}_1 \wedge (\mathrm{F}, \rightarrow \mathrm{F}_1) \wedge (\mathrm{G}, \rightarrow \mathrm{G}_1)] \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1). \tag{3}$$

$$[\mathrm{B}_1 \wedge \mathrm{C}_1 \wedge (\mathrm{F}, \rightarrow \mathrm{F}_1) \wedge (\mathrm{G}, \mathrm{G}_3 \rightarrow \mathrm{G}_1)] \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1). \tag{4}$$

E. Apriori Based Association Action Rule Mining(AAR)

The Association Action Rules by Ras et al. [1] generates the Action Rules using frequent action sets in Apriori-like fashion. The frequent action set generation is further divided in two steps: the merging step and the pruning step.

- **Merging step**: The algorithm merges the previous two frequent action sets into a new action set.
- **Pruning step**: The algorithm discards the newly formed action set if it does not contain the desired action.

For our example, using the data from table I, the primary action sets generated by AAR are shown in table II. The frequent action sets generated by AAR are shown in table III.

TABLE II
PRIMARY ACTION SETS

Attribute	Primary Action Set
A	(A, A_1), (A, A_2), (A, A_3)
B	(B, B_1), (B, B_2), (B, B_3)
C	(C, C_1), (C, C_2)
E	(E, E_1), (E, E_2), (E, E_3), (E, $E_1 \rightarrow E_2$), (E, $E_1 \rightarrow E_3$), (E, $E_2 \rightarrow E_1$), (E, $E_2 \rightarrow E_3$), (E, $E_3 \rightarrow E_1$), (E, $E_3 \rightarrow E_2$)
F	(F, F_2), (F, F_3), (F, $F_2 \rightarrow F_1$), (F, $F_2 \rightarrow F_3$), (F, $F_3 \rightarrow F_1$), (F, $F_3 \rightarrow F_2$)
G	(G, G_1), (G, G_2), (G, G_3), (G, $G_1 \rightarrow G_2$), (G, $G_1 \rightarrow G_3$), (G, $G_2 \rightarrow G_1$), (G, $G_2 \rightarrow G_3$), (G, $G_3 \rightarrow G_1$), (G, $G_3 \rightarrow G_2$)
D	(D, D_1), (D, D_2), (D, D_3), (D, $D_1 \rightarrow D_2$), (D, $D_1 \rightarrow D_3$), (D, $D_2 \rightarrow D_1$), (D, $D_2 \rightarrow D_3$), (D, $D_3 \rightarrow D_1$), (D, $D_3 \rightarrow D_2$)

TABLE III
FREQUENT ACTION SETS

Iteration	Frequent Action Set
Iteration 1	(A, A_1) $\wedge$ (D, $D_2 \rightarrow D_1$) (A, A_2) $\wedge$ (D, $D_2 \rightarrow D_1$) (A, A_3) $\wedge$ (D, $D_2 \rightarrow D_1$) (B, B_1) $\wedge$ (D, $D_2 \rightarrow D_1$) (B, B_2) $\wedge$ (D, $D_2 \rightarrow D_1$) (B, B_3) $\wedge$ (D, $D_2 \rightarrow D_1$) ....
Iteration 2	(A, A_1) $\wedge$ (B, B_1) $\wedge$ (D, $D_2 \rightarrow D_1$) (A, A_1) $\wedge$ (B, B_2) $\wedge$ (D, $D_2 \rightarrow D_1$) (A, A_1) $\wedge$ (B, B_3) $\wedge$ (D, $D_2 \rightarrow D_1$) ....
Iteration n	....

In our example, the action set is discarded if the desired result $(D, 2 \rightarrow 1)$ is not present in it. From each frequent action set, the association Action Rules are created. The algorithm thus generates frequent action sets and forms the association Action Rules from these action sets. For our example, using the data from the Information system in table I, the algorithm generates Association Action Rules, an example is shown below:

$$(B, B_1 \rightarrow B_1) \wedge (C, C_1 \rightarrow C_1) \wedge (E, E_3 \rightarrow E_1) \rightarrow (D, D_2 \rightarrow D_1)$$

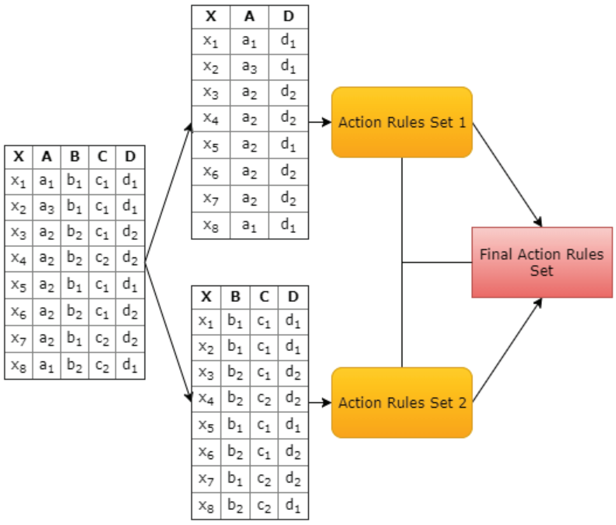

Fig. 1. Example Data Distribution using Vertical Data Distribution.

IV. METHODOLOGY

A. Vertical Split - Data Distribution for Scalable Association Action Rules

Authors Bagavathi et al. [8] propose extracting Action Rules by splitting the data vertically, in contrast to the classical horizontal split, which is performed by parallel processing systems. This method utilizes Association Action Rules [1] - an iterative method to extract all possible action rules. In order to overcome the computational complexity and expense, authors Bagavathi et al. [8] propose vertical data split method for faster computation and parallel processing. In this method, the data is split vertically into 2 or more partitions, with each partition having only a small subset of attributes. The example Data Distribution using Vertical Data Distribution is shown in the Fig.1 The algorithm runs separately on each partition, does transformations like map(), flatmap() functions and combine results with join() and groupBy() operations.

B. Hybrid Action Rule Mining

The Rule-Based Action Rule Mining method using LERS [2] has the disadvantage of calculating the preexisting decision rules in order to generate the Action Rule. To do that it requires complete set of attributes which makes it difficult to implement it in a distributed cloud environment.

The Object-Based Action Rule Mining method can be implemented in distributed cloud environment by using vertical data split method [8], in which only subsets of the attributes are taken for scalability purpose. However, since this method is iterative in nature it takes longer time to process huge datasets.

Hybrid Action Rule mining approach [9] generates complete set of Action Rules by combining the Rule-Based and Object-Based methods. It provides scalability for big datasets, and allows for improved performance compared to the Iterative

```
Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
    (where certainRules are provided by algorithm LERS)
    for each rule r in certainRules
        if consequent(r) equals decisionTo
            Form ActionRuleSchema(r)
            ARS ← ActionRuleSchema(r)
        end if
    end for
    for each schema in ARS
        Identify objects satisfying schema
        Form subtable
        Generate frequent action sets using Apriori
        Combine frequent action set to form Action Rules
        (Such that the frequent action sets satisfy the decisionFrom → decisionTo)
        Output ← Action Rules
    end for
```

Fig. 2. Hybrid Action Rule Mining Algorithm.

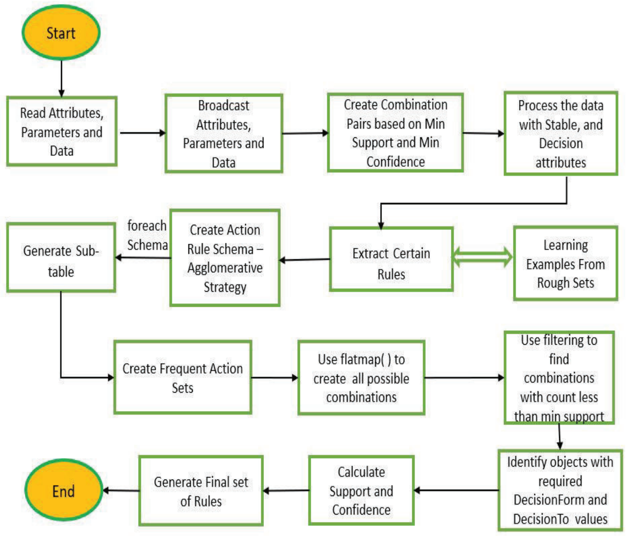

Fig. 3. Hybrid Action Rule Mining Algorithm - Flowchart.

Association Action Rule approach. The pseudocode of the algorithm is given in Fig.2.

The Hybrid Action Rule Mining Algorithm works with the Information System as follows. The information system in table I has the following attributes: stable P_{st}, flexible P_{fl} and decision d, $\mathbb{P} = (P_{st}, P_{fl}, \{d\})$. From table I $P_{st} = \{A, B, C\}$, $P_{fl} = \{E, F, G\}$, and $d = D$.

The following example re-classifies the decision attribute D from $d_2 \rightarrow d_1$. The algorithm Fig. 2. first uses the LERS method to extract the classification rules that are certain and next generates Action Rule schema as given in the following equations " 5" ," 6".

$$[\mathrm{B_1} \wedge \mathrm{C_1} \wedge (\mathrm{F}, \rightarrow \mathrm{F_1}) \wedge (\mathrm{G}, \rightarrow \mathrm{G_1})] \rightarrow (\mathrm{D}, \mathrm{D_2} \rightarrow \mathrm{D_1}). \quad (5)$$

$$[(\mathrm{E}, \rightarrow \mathrm{E_1})] \rightarrow (\mathrm{D}, \mathrm{D_2} \rightarrow \mathrm{D_1}). \quad (6)$$

For each Action Schema, the algorithm then creates sub-table. For example " 5", generates the following sub-table shown in table IV.

The Hybrid Action Rule Mining Algorithm applies the Association Action Rule extraction algorithm in parallel on each of the sub-tables. The algorithm generates the following Action Rules "7" based on the sub-table shown in table IV.

TABLE IV
SUBTABLE FOR ACTION RULE SCHEMA

X	B	C	F	G	D
x_1	B_1	C_1	F_2	G_1	D_1
x_3	B_1	C_1	F_2	G_3	D_2
x_6	B_1	C_1	F_3	G_1	D_2
x_8	B_1	C_1	F_2	G_3	D_2

$$[\mathrm{B_1} \wedge \mathrm{C_1} \wedge (\mathrm{F}, \rightarrow \mathrm{F_1}) \wedge (\mathrm{G}, \mathrm{G_3} \rightarrow \mathrm{G_1})] \rightarrow (\mathrm{D}, \mathrm{D_2} \rightarrow \mathrm{D_1}). \quad (7)$$

```
Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
    (where certainRules are provided by algorithm LERS)
    for each rule r in certainRules
        if consequent(r) equals decisionTo
            Form ActionRuleSchema(r)
            ARS <- ActionRuleSchema(r)
        end if
    end for
    for each schema in ARS
        Identify objects satisfying schema
        Form subtable
        while subtable size > Theta θ
            Divide subtable until subtable < Theta θ
        Generate frequent action sets using Apriori
        Combine frequent action set to form Action Rules
        (Such that the frequent action sets satisfy the
            decisionFrom -> decisionTo)
        Output <- Action Rules
    end for
```

Fig. 4. Hybrid Action Rule Mining with Threshold Algorithm.

This Hybrid Action Rule algorithm is implemented using Spark [10] and it runs separately on each sub-table and does transformations like map(), flatmap(), join(). The entire methodology of this algorithm is shown in Fig. 3.

C. Modified Hybrid Action Rule Mining

Hybrid Action Rule mining method has a major disadvantage. If the Size of the Intermediate Table becomes very large it affects the performance and the scalability of this method. To solve this problem, we propose a Threshold θ to control the size of the table and increase the computational speed.

Our proposed modified version of the Hybrid Action Rule Mining algorithm is presented in the Fig. 4 and the proposed methodology is depicted in the Fig. 5.

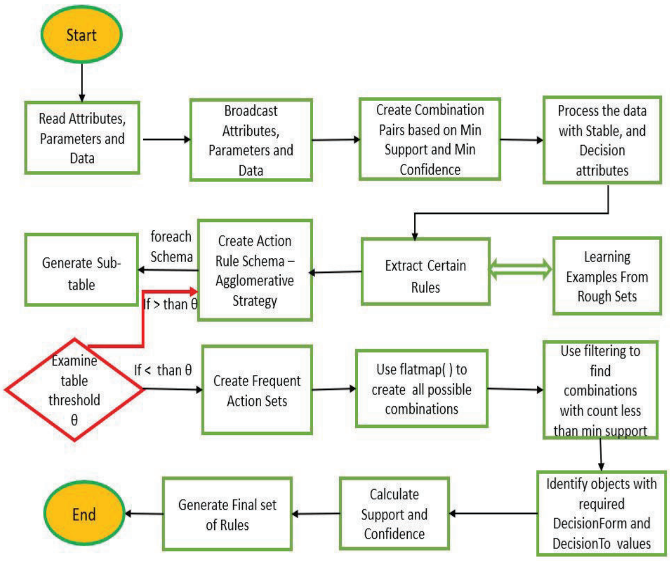

Fig. 5. Hybrid Action Rule Algorithm Mining with Table size threshold θ - Flowchart.

V. Experiments and Results

In this work we use, student survey data to evaluate the student emotions and their overall satisfaction on the teaching methods, group work experiences with respect to a course. The survey is designed to extract some meaningful insights on students' feelings towards the Active Learning methodologies and other factors that can help students in their learning process. This data is collected in the courses which implement the Active Learning methodology and teaching style. This survey dataset contains 50 attributes. There are 549 instances and 59 attributes in the original data. Data is collected in classes that implement Active Learning methodologies to assess student opinions about their learning experience in the years 2019, 2020. The data size on disk is 59 Kilobytes. To test the performance of our proposed method with BigData, 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.

We compare our proposed method with the Vertical Split - Data Distribution for Scalable Association Action Rules method and Hybrid Action Rule mining method. We achieve faster computational time through our new proposed method.

A. Experiment 1 - Vertical Data Split Method Implementation in Spark AWS Cluster

In this experiment we apply the Vertical Data Split Method on the Student Survey Data - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB of memory and EBS Storage of 64GiB. For very large data this method requires additional resources. We find we must provide extra 32 Gigabytes of memory to complete computation on the replicated data in 2400 seconds. Otherwise, the method receives OutOfMemory Exception with our replicated Student Survey Data. This occurs because of iterative nature of the algorithm with large data that causes computational overhead and requires extra hardware memory resources to work successfully. This method only works for Association Action Rules because it considers only subset of the attributes.

Selected Action Rules generated by this experiment are shown in table V.

Enhance Student Emotion - Sadness $\rightarrow$ Joy

1) $AR1SadnesstoJoy : (GroupAssignmentBenefit, SharedKnowledge \rightarrow SocialLearning) \wedge (LikeTeamWork, 1Don't \rightarrow 5VeryMuch) \wedge (TeamMemberResponsibility, HelpfulMembers \rightarrow ResponsibleMembers) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 75.0\%]$
2) $AR2SadnesstoJoy : (GroupAssignmentBenefit, None \rightarrow None) \wedge (LikeTeamWork, 3Somewhat \rightarrow 5VeryMuch) \wedge (TeamMemberResponsibility, TechnicallyIneffectiveMembers \rightarrow FriendlyMembers) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 50.0\%]$
3) $AR3SadnesstoJoy : (NumberofTeamMembers, 8to10 \rightarrow 10orMore) \wedge (LikeTeamWork, 3Somewhat \rightarrow 5VeryMuch) \wedge (GroupAssignmentBenefit, None \rightarrow SocialLearning) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 16.6\%]$

TABLE V
Sample Action Rules ::: Sadness to Joy ::: - Student Survey Data - Vertical Data Split Method.

The Action Rule 1 says that when GroupAssignmentBenefit changes from SharedKnowledge to SocialLearning and LikeTeamWork changes from 1Don't to 5VeryMuch and TeamMemberResponsibility changes from HelpfulMembers to ResponsibleMembers then the StudentEmotion changes from Sadness to Joy. This shows that when the Student likes TeamWork and the group contains Responsible TeamMembers and benefits from GroupAssignment then it enhances the Student's Emotion from Sadness to Joy.

B. Experiment 2 - Hybrid Method Implementation in Spark AWS Cluster

In this experiment we apply the Hybrid Action Rule Mining Method on the Student Survey Data - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB of memory and EBS Storage of 64GiB. This method takes 5088 seconds to complete computation on our replicated Student Survey Data.

Selected Action Rules generated by this experiment are shown in VI .

Enhance Student Emotion - Sadness $\rightarrow$ Joy
1) $AR1SadnesstoJoy : (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam \rightarrow 3AverageSenseofBelongingtotheTeam) \wedge (NumberofTeamMembers, 5to7 \rightarrow 10orMore) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 20.0, Confidence : 59.0\%]$ 2) $AR2SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (TeamWorkHelpedDiversity, 2Occasionally \rightarrow 3Often) \wedge (GroupAssignmentBenefit, None \rightarrow AllofThem) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 20.0, Confidence : 100\%]$ 3) $AR3SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (GroupAssignmentBenefit, None \rightarrow SharedKnowledge) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 34.0, Confidence : 85.0\%]$

TABLE VI
SAMPLE ACTION RULES ::: SADNESS TO JOY ::: - STUDENT SURVEY DATA - HYBRID METHOD.

The Action Rule 1 in table VI says when TeamSenseOfBelonging changes from 2BelowAverageSenseofBelongingtotheTeam to 3AverageSenseofBelongingtotheTeam and the NumberofTeamMembers changes from 5to7 to 10orMore then the StudentEmotion transforms from Sadness to Joy. This rule has confidence of 59% and support of 20. This shows that when the Student's sense of belonging to the Team is average and the team contains 10orMore members then it upgrades the Student's Emotion from Sadness to Joy.

C. Experiment 3 - Modified Hybrid Method Implementation in Spark AWS Cluster

In this experiment we apply our proposed Modified Hybrid Action Rule Mining Method on the Student Survey Data - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB of memory and EBS Storage of 64GiB. Our proposed method takes 4002 seconds to complete computation on the replicated Student Survey Data. We experiment with 3 different Threshold values of θ ::: 10, 15 and 20.

The runtime comparison for different Threshold values is shown in the below table VII

Threshold θ	Time Taken
10	4628 seconds
15	4002 seconds
20	8670 seconds

TABLE VII
THRESHOLD VALUES - θ RUN TIME COMPARISON.

Threshold value θ = 15 provides optimum performance.

Selected Action Rules generated by this method are shown in VIII .

Enhance Student Emotion - Sadness $\rightarrow$ Joy
1) $AR1SadnesstoJoy : (TeamFormation, 2BelowAverage \rightarrow 4Perfect) \wedge (NumberofTeamMembers, 5to7 \rightarrow 8to10) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 21.0, Confidence : 62.0\%]$ 2) $AR2SadnesstoJoy : (LikeTeamWork, 1Don't \rightarrow 3Somewhat) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 21.0, Confidence : 91.0\%]$ 3) $AR3SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (GroupAssignmentBenefit, None \rightarrow SocialLearning) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 34.0, Confidence : 85.0\%]$

TABLE VIII
SAMPLE ACTION RULES ::: SADNESS TO JOY ::: - STUDENT SURVEY DATA - HYBRID METHOD WITH THRESHOLD.

The Action Rule 1 in table VIII says when TeamFormation changes from 2BelowAverage to 4Perfect and the NumberofTeamMembers changes from 5to7 to 8to10 then the StudentEmotion changes from Sadness to Joy. This rule has support of 21 and confidence of 62%. This shows how having a good team and increased number of Team Members enhances a Student's Emotion from Sadness to Joy.

D. Runtime Comparison of the above 3 implementations with respect to Student Survey Data in Spark AWS Cluster

We compare the execution runtime of the above described implementations:Vertical Data Split Method in Spark AWS Cluster, Hybrid Method Implementation in Spark AWS Cluster and Hybrid Method with Threshold Implementation in Spark AWS Cluster. The runtimes are given in below table IX

Method	Time Taken
Vertical Data Split Method _* with additional resources: 32 GB cluster memory _* with standard memory	 2400 seconds OutOfMemoryException
Hybrid Method	5088 seconds
Modified Hybrid Method	4002 seconds

TABLE IX
RUNTIME COMPARISON OF THE ABOVE 3 IMPLEMENTATIONS WITH RESPECT TO STUDENT SURVEY DATA.

Our proposed Hybrid Method with Threshold (Modified Hybrid Method) shows improved performance over the previous Hybrid Method, and shows the best performance with standard memory.

VI. CONCLUSION

Emotions play a very important role in the lives of people all over the world. Today we have multiple platforms available for electronic communication. The expansion of social media, online surveys, customer surveys, blogs, industrial and educational data generates large amounts of data. Hidden in the data are valuable insights on people's opinions and their emotions. Recognizing emotions from the text data through Action Rules can benefit a lot of industries [11], including

Social Media and Education. In this work we mine Student Survey Dataset. We experiment with Student Survey Data to test our proposed Modified Hybrid Action Rule Mining Algorithm to suggest ways for improving Student Emotions. The data contains student opinions regarding the use of Active Learning methods, Teamwork and class experiences. The discovered Action Rules help to enhance the student Emotion and learning experience from negative to positive and from neutral to positive.

VII. Future Work

Our proposed method improves the processing time. However, the quality of rules may decrease. In the future, we plan to use Correlation of Attributes and run classical Clustering Algorithm. This obtains optimal Vertical Partitioning which is flexible. We plan to apply Agglomerative strategy to change levels of vertical partitions. We also plan to examine the Quality of the Action Rules using F-Score.

References

[1] Z. W. Ras, A. Dardzinska, L.-S. Tsay, and H. Wasyluk, "Association action rules," in *IEEE/ICDM Workshop on Mining Complex Data (MCD 2008), Pisa, Italy, ICDM Workshops Proceedings,*. IEEE Computer Society, 2008, pp. 283–290.

[2] J. W. Grzymala-Busse, "A new version of the rule induction system lers," *Fundamenta Informaticae*, vol. 31, no. 1, pp. 27–39, 1997.

[3] A. Dardzinska and Z. W. Ras, "Extracting rules from incomplete decision systems: System erid," in *Foundations and Novel Approaches in Data Mining, (Eds. T.Y. Lin, S. Ohsuga, C.J. Liau, X. Hu), Studies in Computational Intelligence*. Springer, 2006, vol. 9, pp. 143–154.

[4] Z. W. Ras and L.-S. Tsay, "Discovering extended action-rules (system dear)," in *Intelligent Information Systems 2003, Advances in Soft Computing, Proceedings of the IIS'2003 Symposium, Zakopane, Poland*. Springer, 2003, pp. 293–300.

[5] L.-S. Tsay and Z. W. Ras, "Action rules discovery: system dear2, method and experiments," *in the Special Issue on Knowledge Discovery, Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 119–128, 2005.

[6] Z. W. Ras and E. Wyrzykowska, "Arogs: Action rules discovery based on grabbing strategy and lers," in *Proceedings of 2007 ECML/PKDD Third International Workshop on Mining Complex Data (MCD 2007), Univ. of Warsaw, Poland*. Citeseer, 2007, pp. 95–105.

[7] S. Im and Z. W. Raś, "Action rule extraction from a decision table: Ared," in *Foundations of Intelligent Systems, Proceedings of ISMIS'08, A. An et al. (Eds.), Toronto, Canada, LNAI*, vol. 4994. Springer, 2008, pp. 160–168.

[8] A. Bagavathi, V. Rao, and A. A. Tzacheva, "Data distribution method for scalable actionable pattern mining," in *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*. ACM, 2018, p. 3.

[9] A. Tzacheva and J. Ranganathan, "Pattern discovery from student feedback - identifying factors to improve student emotions in learning." IOP Publishing, jan 2021.

[10] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863113

[11] A. F. A. Nasir, E. S. Nee, C. S. Choong, A. S. A. Ghani, A. P. P. A. Majeed, A. Adam, and M. Furqan, "Text-based emotion prediction system using machine learning approach," *IOP Conference Series: Materials Science and Engineering*, vol. 769, p. 012022, jun 2020. [Online]. Available: