Transactions on Machine Learning and Artificial Intelligence - Vol. 10, No. 3
Publication Date: June, 25, 2022
DOI:10.14738/tmlai.103.12397.
Tzacheva, A., Chatterjee, S., & Ras, Z. (2022). Cloud Mining Actionable Pattern Discovery in Big Data: A Survey. Transactions on Machine Learning and Artificial Intelligence, 10(3). 42-102.

Cloud Mining Actionable Pattern Discovery in Big Data: A Survey

Angelina Tzacheva
Computer science, University of North Carolina
Charlotte, United States

Sanchari Chatterjee
Computer science, University of North Carolina
Charlotte, United States

Zbigniew Ras
Computer science, University of North Carolina
Charlotte, United States

ABSTRACT

Everyday several tons of data are generated by Social Media and 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 large 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 ObjectBased approach. This paper gives a brief survey of previous works on action rule mining algorithms, and discusses Action Rule Mining on BigData and in a distributed environment. The applications of actionable action rules in BigData include business opinion mining, educational opinion mining, medical data decision support systems, text emotion detection, and social media data recommendations.

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

INTRODUCTION

Complex and large data sets emerging pervasively poses significant challenges to cloud computing. Both structured and unstructured data have grown exponentially in a cloud. The main [1]challenge in big data is to handle the unstructured data in a cloud.Cloud with its vast resources in various capacities is not a new concept but is matured enough to be utilized for the huge amount of high velocity and variety data incurred by Data Science and related fields of science, technology and culture [2].In paper [3] the author research Big Data Mining

Approaches in Cloud Systems and provides knowledge related to cloud-compatible problems and computing techniques to promote Big Data Mining in Cloud Systems.Big Data refers to a collection of complex and heterogeneous data sources, technologies and methodologies that have come out from the exponential growth in data creation over the past few decades [4]. There is an increasing demand for executing the data in real or near-real time, or for predictive maintenance, in the period [5] where globalization is directly connected to digitalization of the world. [6]Data driven from the real life problem helps to provide actionable knowledge or deeper insights about it, which further helps the computing process smart and automatic.Big Data tools are already spreading its wing in this new complex field of science, for example, to understand climate change as a theory-guided data science paradigm , to monitor seasonal changes in climate change , and learn how to manage the risks of climate change [7], explore soft data sources, e.g., Twitter [8], or demonstrate the potential of Systems of Systems (SoS), for instance, the exploration of the structure and relationships across institutions and disciplines of a global Big Earth Data cyber-infrastructure.

The social media has become a sensation among the youth worldwide specially during the pandemic period where the student had their classes online. Students sometimes feels more comfortable in explaining their thoughts in social media rather than in person.They can share their thoughts, mental state, moments, stand on specific social/national/ international issues through text, photos, audio and video messages and posts, with families, friends , relatives, etc, [9]. Social media sites like Twitter , Facebook, Instagram, etc and personal blog sites provides a great way out for students to share joy [10],sadness, stress, struggle and seek social support. This emotions in return generates huge amount of unstructured data for the researchers. With the increasing growth of the digital platform in cyberspace, such as blogs and social networks, individuals and organizations are depending on public opinion for their decision-making. The paper [11] is a structured literature review proposing Kansei approach in classifying people's sentiments and emotion based on text in cyberspace. Kansei approach can measure people's impressions using artefacts based on senses including feeling ,sight and cognition reported precise results for the assessment of human emotion. This research suggests that the Kansei approach should be a complementary factor including in the development of a dictionary focusing on emotion in the national security domain.

Massive amount of texts are getting generated on the web, which contain a variety of viewpoints, attitudes, and emotions for products and services [12]. Information mining from online comments is essential for enterprises to improve their services or products and for consumers to make decisions to purchase. To solve the problem of poor comprehensibility, high time requirement, and low accuracy, while doing text-mining in models, the authors of the paper mentioned above finds a solution to it. The paper provides an unsupervised Topic-Specific Emotion Mining Model (TSEM), which adds corresponding relationship between aspect words and opinion words to express comments as a bag of aspect–opinion pairs. Companies providing products and services collect customer satisfaction data through surveys. Educational institutions collect data from students on their opinion for their learning experience in the courses. Analysis of this data may provide insights into the opinion of people as well as their feelings towards certain subjects, products or services. [13] There comes the concept of Opinion Mining that is the 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 and also improves the computational performance.

In the most recent years of 21st century almost every industry or company or institution is undergoing some digital transition, resulting in vast amounts of structured and unstructured data. [14] The enormous task for researchers is to transform unstructured data into meaningful insights that can help them in decision-making. Educational institutions like Schools, Universities, etc collect data from students on the basis of survey regarding the opinion for their learning experience in the courses. [15] Companies providing products and services also collect customer feedback data through surveys, like Amazon, Google, Intel, etc. [16],Analysis of this data may provide discernment about the opinion of student and the people participated in the survey ,as well as their feelings towards certain subjects, products or services. Moreover, People's active feedback is valuable not only to measure customer satisfaction or to keep track of the competition but also for consumers who want to learn more about a product or service before buying it [17].

Machine learning methods have advances a lot regarding helping machines understand human behavior better than ever. Out of all the human nature one of the most important aspects of human behavior is emotion. In paper [18] authors present a new network based on a bidirectional GRU model to show that capturing more meaningful information from text can significantly improve the performance of the old models. The results show significant improvement with an average of 26.8 point increase in F-measure on our test data and 38.6 increase on the totally new dataset.

Identifying discrete nature of positive and negative emotions can help improve many applications. [19] For example, the two emotions Fear and Anger both express negative opinion of a person toward something, however, it has been shown that fearful people tend to have pessimistic view of the future, while angry people tend to have more optimistic view. The authors in paper [19] presents a new approach to incorporate emotional information of words into embedding models. The method added a secondary training stage which uses an emotional lexicon and a psychological model of basic emotions leading to retrain models performance better than their original counterparts.

In the field of healthcare also social media like Twitter have become an important source of health-related information provided by healthcare professionals and citizens. For example, people have been sharing their thoughts, opinions, and feelings on the Covid-19 pandemic [20]. Patients were directed to stay isolated from their families, friends, loved ones, and others, which affected their mental health status. To save patients from mental health issues like depression, health practitioners must use automated emotion detection techniques [21]. People commonly share their feelings or beliefs on sites through their posts, and if someone seemed to be depressed or sad, people could reach out to them to help, thus averting deteriorated mental health conditions and can even save one's life. In paper [22] the authors proposed Emotion Ontology model that is efficient in extracting the full range of human emotions pertaining to COVID-19-related concepts. Their study is able to extract and detect almost 81 percent of TP and is able to detect a considerable amount of FN.

The paper [23] is a detailed study of 3 methods: Dijkstra's Shortest Path algorithm, Breadth First Search algorithm and Depth First Search algorithm, for searching the Action Graph for Rules of Lowest Cost. The study represented Action Rules as graphs called Action Graphs, which uncovers undiscovered relationship between actionable patterns in the recommended action rules, to search low cost Action Rules from Action Graphs in the distributed scenario using Spark framework. [23] [23]applied the proposed algorithm to three datasets in transportation, medical, and business domains. Results show how these domains can benefit from the discovered actionable recommendations of low cost. This study finds that BFS and DFS perform better in terms of processing time, for large datasets when incorporated into parallel frameworks like Spark GraphX, but for smaller datasets, all parallel algorithms perform almost similar. However, the Dijkstra's algorithm discovers higher number of Low Cost Action Rules. They implemented the DFS method in Spark GraphX, which was till then not implemented. For Action Graphs, there was a modification of DFS algorithm to work similar to neighborhood aggregation.

The data generated is called Big data and opinion mining was introduced to analyze this big data effectively and efficiently [24].Opinion Mining is the process of extracting valuable data from opinions of people in order to get hold of their feelings and thoughts .Mining the opinions of people has applications in several areas: understanding what people like or dislike is critical for the formation of good teaching and learning environment, creating informed business and making political decisions. In the work [25],the main area is to focus on data distribution phase of the distributed actionable pattern mining problem, where they primarily work on the vertical data splitting strategy using information granules, that provides meaningful representations of complex information systems as fine grained granules. The result of the vertical data partitioning were more logical using information granules instead of splitting the data in random. The downside of this approach became little complex when each partition produce large number of rule.

In the paper [26]the author discovered two new methods for Data Distribution for Cloud Parallel processing , which can be applied to Actionable Pattern Mining via Action Rules. The Method 1 can be applied to most Action Rules algorithms, including Action Rules, system DEAR, ARAS, systems ARED. Whereas, Method 2 is specifically designed for Association Action rules, which is the most complex and time-consuming Action Rules extraction method, however it discovers all possible Action Rules.The authors previous works divides the data by random using default partitioning provided by Hadoop MapReduce, and Apache Spark. For that,the calculation of Support and Confidence did not represented the original support and confidence very well for Action Rules extracted on the entire dataset, or the support and confidence may be incorrect all together, with respect to their new method support and confidence on the same datasets. Action Rule mining is one of the data mining approaches to recommend actions 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 one [27].In the article [28], the author of the article proposed a scalable action mining method to recommend hospitals and taxpayers on what actions would potentially reduce patient readmission to hospitals, as the readmission rates in hospital is increasing rapidly in past few years. The data they used to evaluate the approach is the Healthcare Cost and Utilization Project(HCUP)data. All their proposed scalable approaches are cloud based and they worked on Apache Spark to handle data processing inorder to make recommendations. Although this method proved efficiency in the execution time, but it is not very optimal in memory usage in the distributed setup. Also, the methods lack subject matter experts input to evaluate actionable recommendations. In future, they plan to address these problems by providing more optimal load balancing modules that are both memory and time optimal.Moreover, they also plan to use experts input to evaluate the final results.

The formal definition of Action Rule [27] is defined as in equation "(1)".

$$[(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow (\varphi \rightarrow \psi) \quad (1)$$

Where, ω represents the conjunction of fixed condition features shared by both groups, $(\alpha \rightarrow \beta)$ represents changes in flexible attributes, and $(\varphi \rightarrow \psi)$ is the desired change in the decision attribute such that it benefits the user. In this work, we are focusing on Opinion Mining from Text to suggest various Actionable Pattern Recommendations. The Actionable Patterns may suggest ways to alter the user's emotion or sentiment, which is the existing state to a more positive or desirable state, like sadness to happiness. Action Rule Mining literature mainly consists of two major frameworks, one is Rule-Based approach and the other is Object-Based approach. The third approach is known as Hybrid Action Rule mining that combines the above two frameworks having the advantage of working with large datasets. In this paper we propose a Modified Cloud Mining Actionable approach that further improves the computational performance. We apply our method to Student Survey Education data. Our aim is to suggest ways to improve the Teaching Methods and Student Learning. We implement and test our system in Scalable Environment with Big Data using Apache Spark platform.

DISTRIBUTED COMPUTING FRAMEWORKS

Apache Hadoop [29] MapReduce [30] and Apache Spark [31] are the two most common and famous parallel processing frameworks."Hadoop is a framework that allows distributed processing of large datasets across clusters of computers using single programming model" as described in [29].To process large datasets in a parallel fashion on top of Hadoop, there is a requirement of programming models those are MapReduce and Spark . One of the crucial components in Hadoop framework is Hadoop Distributed File System (HDFS) [32], that can split the large data and can manage small data chunks in multiple nodes. MapReduce and Spark are placed on top of HDFS to access and work with those smaller sections of data. Since the tasks run on such small sets of data and all the tasks run in parallel, total computational time is much less than compare to the same task running on the larger dataset in a single machine. There are also other features of these frameworks other than distributed and parallel computations like their data management ,architectures, etc are not within the scope of this paper. In the section below, we discuss about parallel processing design of these selected frameworks.

A. Hadoop MapReduce

Authors J. Dean and S. Ghemawat, from Google, in [30] proposed MapReduce, a parallel computing programming model,that has the ability to process large data in parallel with distributed algorithms in a fault tolerance way in a cluster of nodes or separate computers. The authors affirm that the goal of MapReduce is to make end-users to think about how to do put their methods or algorithms into MapReduce programming model instead of considering other complex programs such as task parallelization,data distribution, load balancing and fault-tolerance. Hadoop run all these complex functionalities in background of a given MapReduce program. MapReduce always works as the top layer of Hadoop Distributed File System (HDFS), a distributed file system to store the entire data as multiple splits into smaller datasets in different locations.

Hadoop MapReduce has two functions: *Map* and *Reduce*
Both the functions takes input and produces output as <Key, Value> pairs. MapReduce runs in a *Master Slave* architecture where both synchronizes well.Master node keeps track of all slaves both in HDFS and in MapReduce tasks and monitors the whole MapReduce execution process. Master node finds current number of available nodes and assign them as Slave nodes. Slave node in turn keeps sending heartbeat signal to its master to inform that it is still working.At the time of system or node failure, slave node stops sending the signal and the master can allocate the failed task to some other/ new slave node.Initially, the master node allocates slave nodes for running map tasks. Now the slave nodes getting their own tasks,as instructed by the master, MapReduce starts its Map phase on them. Since the map function runs on a small data set from the larger one, the results form are usually intermediate outputs. Each map function writes their intermediate output <Key, Value> pairs back into HDFS. The master node again finds an available slave node(s) to perform reduce task. Reduce function collects all values for a single key from the map task, sorts them, perform computations like aggregation on them and writes the output as <Key, Value> pair to the distributed file system.Hadoop [29] preserves data locality by confirming the distance between data node and slave node is minimum. In the HDFS, each small datasets exists in three different locations. Therefore, even if the node having the data is down, Hadoop can get the data from other locations. Other MapReduce tasks can use this result in order to perform other sort of computations. Fig. 1 gives an overview of the execution of MapReduce.

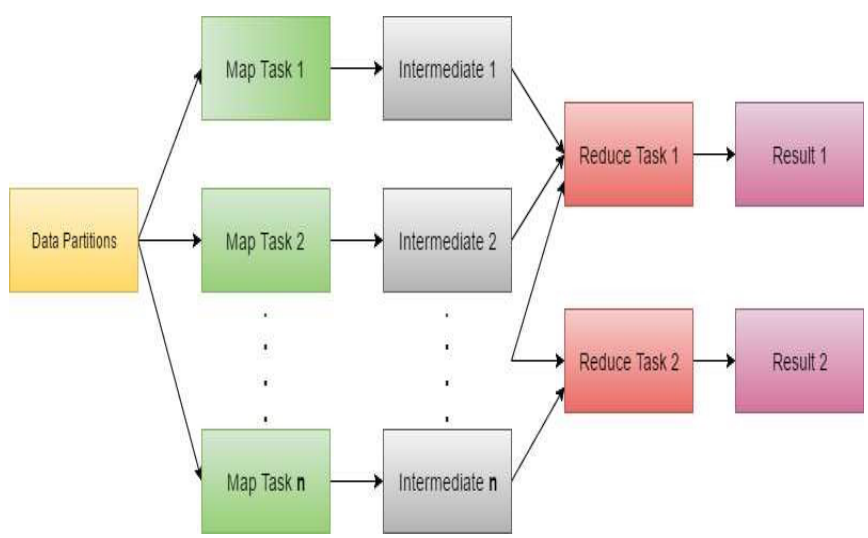

Fig. 1. Overview of MapReduce execution. The data partitions and results from Map and Reduce tasks reside in the HDFS. The Map tasks and Reduce tasks in the distributed systems run in a parallel fashion

B. Spark
Although Hadoop MapReduce is a good model to work with distributed parallel operations and it has some fault tolerance scheme, it also has some disadvantages of reading and writing

operations of intermediate outputs on HDFS . This results in frequent disk access for using and storing the data. This in turn makes many iterative machine-learning algorithms like clustering,classification and regression to work inefficiently. Spark [31]framework introduce a distributed memory abstraction method known as Resilient Distributed Datasets (RDD) to avoid the pitfall of MapReduce that is frequent disk access from the data nodes.

Resilient Distributed Datasets (RDD) basically performs in-memory computations unlike Hadoop MapReduce [30] to store the results in disks for a huge dataset in a fault tolerant approach. Spark reads the data from the given source, split them into small datasets and store each of them in node memory as an RDD. The driver node instructs tasks to the worker nodes where the data is placed. A task can be either *transformation* or *action*. During *transformation* phase, Spark perform computations on a data split and the results are stored in-memory of other worker nodes. Further the result of all worker nodes together forms another RDD. *Action* phase collects the resulting RDDs and send the collection to the driver node or save the collection to a storage system like databases. Fig. 2 gives an overview of the execution of Spark.

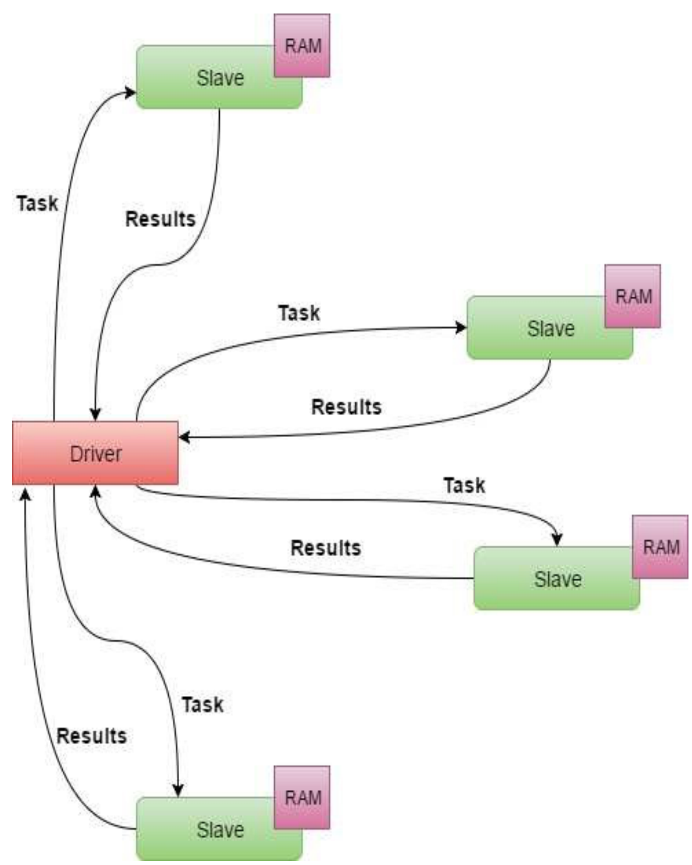

Fig. 2. Overview of Spark execution using Resilient Distributed Datasets (RDD). Slave nodes accept tasks such as transformations. After performing the tasks, slave nodes cache the result in RAM as an RDD. Other transformation tasks can use this RDD or the Driver node retrieve the results by initiating an action task

Spark can access such data much faster without any delay like in Hadoop MapReduce as the data and results are present in a system's memory. Therefore, Spark is more suitable for many iterative algorithms like machine learning algorithms. Spark handles fault tolerance by having a lineage graph of RDDs. Fig. 3 shows a simple lineage graph of combining values from two inputs. When certain portion of data is lost or a node fails at a certain Phase, Spark retrieve the lost or failed partition from the lineage graph and produce results to the lower levels. Spark avoids data replication over multiple nodes like MapReduce, because of the presence of lineage graph, which gives some free space in the cluster that the RDDs can use.

Spark also provide many packages like MLlib, GraphX, Spark Streaming and Spark SQL [33].MLlib makes logical machine learning methods scalable and easy. GraphX is the primary ground for graph processing and graph analytics.

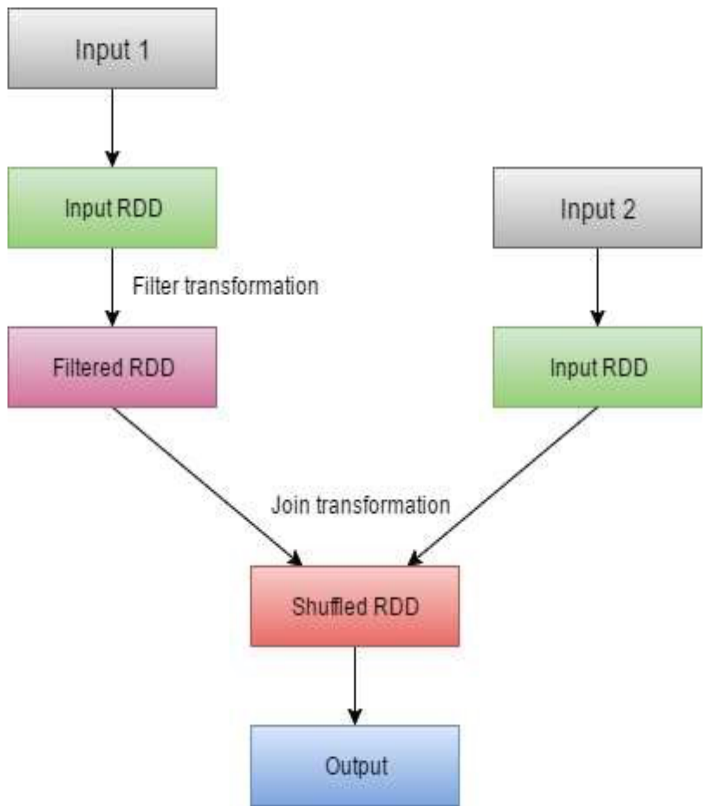

Fig. 3. Lineage graph of joining inputs from Input 1 with Input2. After reading from Input 1, Spark filters the read values to get only required values. Values from both inputs are combined and stored as an output

Spark Streaming can manage a data stream from Kafka [34] or Twitter Stream.Spark SQL can query structured data inside Spark programs.Spark SQL query any tables from databases like Hive, JSON,Cassandra, etc. In the other hand, it can create tables in the databases from the raw dataset . Spark collects the streaming data for small amount of time and create RDDs from the collected data that can be processed further using Spark SQL or MLlib. Fig. 4 is a comparison depiction of MapReduce and Spark when both runs Logistic Regression and K-Means clustering algorithms in different number of machines for a 100GB of data [31].

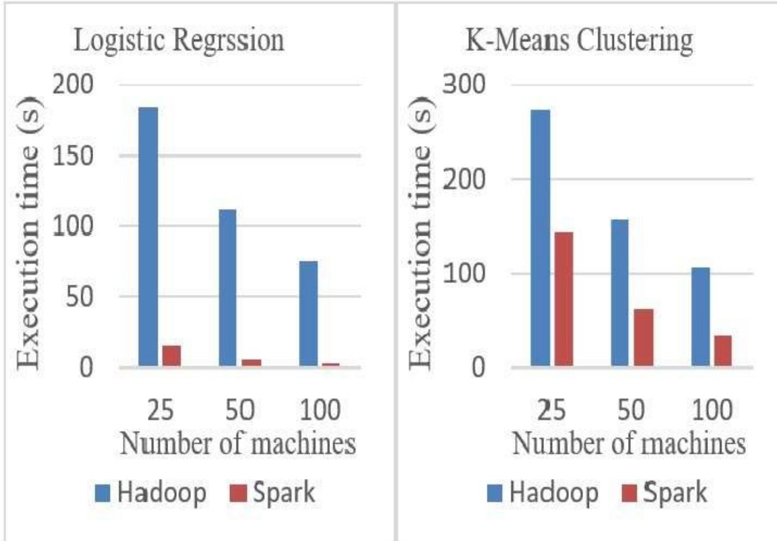

Fig. 4. Execution times of Logistic Regression and K-Means Clustering for 100GB of data.

C. Pig, Hive, HBase

Several database oriented frameworks like Pig [35], Hive [36] and HBase [37] resides in Hadoop community. Apache Pig provides an environment for data processing. Pig can process both structured and unstructured data. Apache pig contains its own data flow language: PigLatin. It is similar to a scripting engine. Apache Pig can connect SQL queries with other programming models such as MapReduce.

Hive is a data warehousing model useful for data analysis and summarization. It is used to load structured data into the Hadoop Distribute File System (HDFS), and provides SQL like queries called Hive Query Language (HiveQL) to query the data.It is also known as query engine. Hive converts the queries into MapReduce jobs in the background, to fetch and process data from different location or multiple nodes and return results.

Apache HBase is a column oriented, non-relational distributed database where the user data resides.HBase has no SQL language, and does not perform any data processing on its tables. Hbase support key value transaction.It is useful for storing matrices of WebPage links, or document term frequency tables.

D. Flink, Storm, Kafka

Apache provides specialized frameworks for big data stream processing, like Flink [38], Storm [39], Kafka [34] , SAMOA [40] and several others [41] apart from Hadoop [30] and Spark [31]. Apache Flink [38] is an open source platform for both stream and batch processing, providing developers two-in-one advantage.

Frameworks like MapReduce and Spark as mentioned earlier are suitable particularly for batch processing systems, that is they work on data that is already collected but are not directly suitable for live streaming data. Apache Storm [39], is a tool specifically designed for streaming data processing. Storm affirms that all tuples from the data stream will be processed.

Kafka [34] is a distributed data-streaming platform that acts as a broker to guide the data to pass reliably from the data producers to data consumers. Kafka processes the streaming data from the source, splits them into topics, stores them in a log structure, from which multiple consumers can subscribe to a topic and read them at a same time. Kafka API can be easily used along with Spark, Storm or Flink.

DISTRIBUTED MACHINE LEARNING

Machine learning algorithms are mostly expensive interms of computation for tuning up its parameters and producing an end model that fits best for a given dataset [42]. These algorithms, when taking large scale of data as an input, obtain large number of high dimensionality data instances, which in other way increases complexities of the algorithms. Such algorithms require data and task parallelization to run efficiently on the enormous set of data. This section describes a study on some of the available frameworks for supporting distributed and parallel computation techniques on machine learning. In the previous section the distributed frameworks as discussed contains their own built in machine-learning library.

A. Spark MLlib

One of the essential part of Apache Spark [31] is Spark MLlib [33] . Spark MLlib creates machine learning models in the distributed environment. MLlib is considered as the distributed machine-learning library that provides simple yet rich ecosystem for running many machine-learning algorithms including logistic regression,linear regression,linear SVM, Na¨ıve Bayes, k-means clustering,Principle Component Analysis, stochastic gradient descent ,decision trees and forests,. Spark, execute iterative algorithms faster due to its in-memory computations feature .Since many machine learning algorithms make series of iterations over the particular set of data, Spark is considered most suitable for many machine learning algorithms. Fig. 4 Compares speed test of Hadoop and Spark for Logistic Regression and K-means clustering.

B. FlinkML

Like Apache Spark, Apache Flink [38] has FlinkML, a machine-learning library, which is still an Apache's incubating project providing data scientists a platform that can create a model on a subset of local data and use the same model in a cluster for the streaming data measured in size megabytes or gigabytes or even beyond.FlinkML also provide facilities for many machine-learning algorithms after getting inspired by Spark MLlib,.

C. Mahout

Apache ecosystem provides a scalable framework called Apache Mahout [43].Mahout is dedicated for distributed machine learning on a batch of dataset. The recommender engine feature of Mahout is quite notable, whose sole purpose is for recommendation .It also have other classification and clustering algorithms. Initially Mahout supported only Hadoop MapReduce [30] jobs. However, presently it can bind with Apache Spark or Flink.

D. SAMOA

Similar to Mahout there is another Apache incubator project known as SAMOA [40].SAMOA is a distributed stream processing machine-learning platform that provides a pluggable machine learning programming abstractions for many machine learning tasks and data mining . It includes: classification, regression and clustering. SAMOA provides the option of implementing an algorithm and plugging it into multiple distributed stream processing engines such as: Spark, Flink, or Storm, without changing a single line of code.

E. GraphLab

Distributed GraphLab [44] framework is dedicated for running distributed and parallel machine learning algorithms. GraphLab performs parallel computations in both shared and distributed memory settings in a directed graph structure. It can be used for representing Neural Networks, for Page Ranking, Social Network mining, or any data in a graph format. Since machine learning models comprise dependencies in data, some of the parameters need to be updated for further computations. GraphLab has an advantage of allowing users to assign data and computation for each vertex or machine and edge in the graph. Each vertex can then interact with the respective neighboring vertices.The distributed GraphLab can perform multiple MapReduce computations concurrently as the communication is based on a graph structure.GraphLab also provide option for storing global variables that is common for all the vertices. Their experimental results show that the system outperforms Hadoop by 20-60 times. In comparison with Apache Spark [45], GraphLab outperforms Spark, which is more optimal than Hadoop due to in-memory computations, only in graph algorithms like PageRank and BFS. Although, GraphLab provides more or less equal performance to nongraph algorithms like SVM and K-means. In the graph processing field, Google provides an open source framework: TensorFlow [46], to efficiently handle graphs for highly scalable deep learning problems.

F. Parameter Server

Another fault-tolerant framework for efficient feature extraction from the huge volume of data distributed over multiple machines is called Parameter server [47] . Like MapReduce [30], Parameter server follows the master-slave architecture. However, Parameter server maintains a server group and slave group, where each group can contain multiple machines. Each machine in the server group can communicate each other ,but,each slave can communicate only with their respective master node. Master nodes distribute the data and tasks to their slave nodes. The results from the slave tasks are only local optimal. To get the global optimality, each master node stores global parameters of an algorithm and replicate the same to other master nodes for scaling and reliability, which the slave nodes can frequently access and update in this way. Parameter server performs well with algorithms like regression, topic modelling and deep learning particularly when the data contains more number of parameters.

MACHINE LEARNING METHODS USED FOR EMOTION CLASSIFICATION

Khan depicts the [48] review of common machine learning algorithms for text classification. Document classification can be categories into three parts as follows: supervised, unsupervised, semi-supervised methods. The automatic classification of documents into predefined categories has observed as an active attention, as the internet usage rate has rapidly increased [48]. Some of the known machine learning approaches are Support Vector Machine (SVM),Vector Space Model, Bayesian classifier, Decision Tree, K- Nearest Neighbour(KNN), Neural Networks, Latent Semantic Analysis etc. Since a broader part of the paper focuses on text classification, so, this section describes about specific classifiers widely used in the literature for text classification. In general, supervised learning techniques are used for automatic text classification. Here, firstly pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents.

A. Support Vector Machines - SVM

Vapnik introduced Support Vector Machine(SVM) is a statistical learning method [49]. In article [50]author details Support Vector Machine as follows: The main idea behind SVM is to find a decision surface that best separates the two class of documents in the n-dimensional space. The samples (documents/datas) that are close to decision surface are called *support vectors* shown in Fig.5 as shown by author Meyer [51]. Major advantage of SVM is its superior runtime-behavior during the categorization of new documents because only one dot product per new document must be computed. A disadvantage is the fact that a document could be assigned to several categories because the similarity is typically calculated individually for each category. Model SVM is a very powerful method and has outperformed other methods in several studies by Authors Dumais et al. [52],Joachims [53],Siolas and d'Alche-Buc [54]and Yang et al. [55]. The following works´ use Support Vector Machine for emotion classification from text [56], [57], [58] and [59].

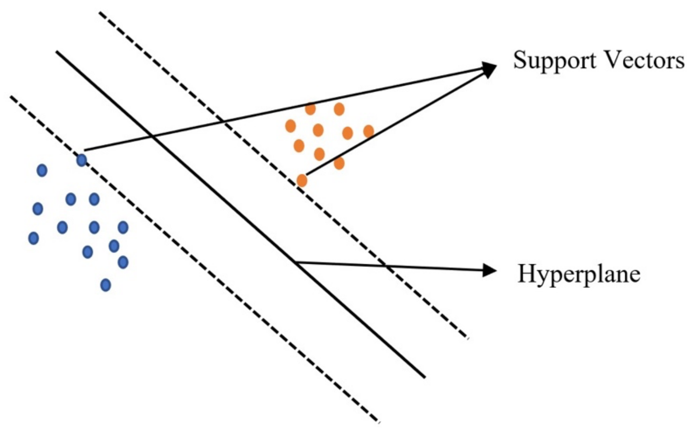

Fig. 5: Support Vector Example

B. Naive Bayes Classifier

Naive Bayes classifier is a simple classifier based on probabilistic events of applying Bayes Theorem with strong independence assumptions [48]. Authors Brucher et al. [50], theory talks about the fact that Independence assumption means the presence of one feature does not affect the presence of other features and the order of features is irrelevant. The conditional assumption is given in equation (2).

$$p(x|y = c) = \prod_{i=1}^{D} p(x_i|y = c) \qquad (2)$$

The computational features of Bayes classifier is efficient because of this independence assumption and also limited applicability. Due to its apparently over-simplified assumptions, the naive Bayes classifiers often work much better in many complex real-world situations [48]. The full Bayesian posterior predictive density on the class label Y given an input X and the training data D is given by equation (3) as explained by author Murphy in [60].

$$p(y = c|x,D) \approx p(y = c|x,\hat{\theta},\hat{\pi}) \propto p(x|y = c,\hat{\theta}_c)p(y = c|\hat{\pi}). \qquad (3)$$

But the performance is relatively low compared to the previous model that is Support Vector Machines. One of the advantages of Naive Bayes classifier is that it requires only small set of training data to determine the classification instances. Naive Bayes is easy to implement compared to other algorithms, however because of conditional independence assumption its performance is very poor when features are highly correlated and does not consider frequency of word occurrences. Authors Wang et al. [61] use Naive Bayes for Twitter emotion classification.

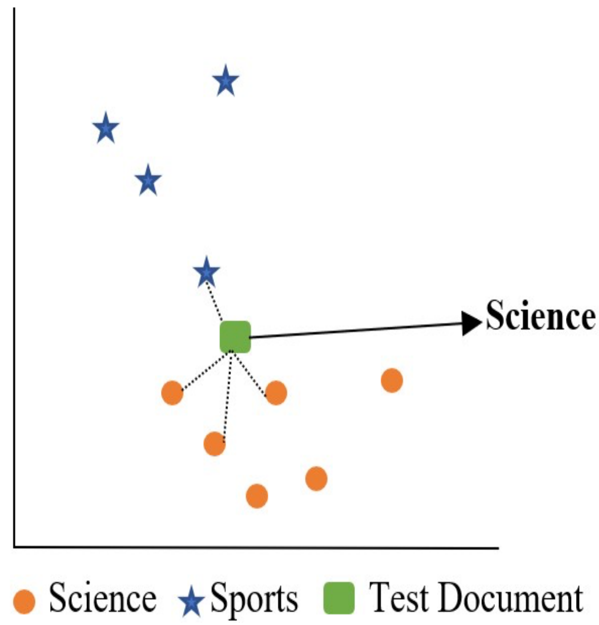

Fig. 6. Naïve Bayes Classifier

C. Vector Space Model

K-Nearest Neighbour is vector Space classification algorithm. Vector space classification algorithm represents each document as a vector with one real-valued component, for instance term frequency - inverse document frequency (tf-idf) [57] weight for each term. In general vector space model is based on contiguity hypothesis. The contiguity hypothesis states that documents in the same class form a contiguous region and regions of different classes do not overlap [62]. Given a test document, majority class that is the nearest neighbour close to the test document is assigned as the class for test document. One of the advantages of K-NN is that, it does not require explicit training data. Because the training phase involved determining the value of "k" and document pre-processing. K-NN is also called as memory-based learning or instance-based learning as it simply memorizes examples in the training set and then compares the test document to them. For example, if there are documents of type science and sports. Given a test document K-NN classifies it based on majority classes that are closest to the test document. Consider the example in Fig. 6. the test document is close to science document and hence classifies as *"science"*. Authors Jain et.al [63] and Kim et al. [64] use Vector Space Model for text emotion classification.

D. Decision Tree Classifier

The tree has root node, internal node and leaf node as shown in Fig.7.The classification problem can generally be solved with the help of tree. The leaves of the tree represent the document category and branches represent features that lead to the specific category. The root node is the document for classification. According to authors Khan et al. [48], main advantage of decision tree is its simplicity in interpreting and understanding, even for non-expert users. Text classification generally involves a greater number of features or attributes. Decision tree performs poorly with larger feature/data set. However, if the feature/data set is organized and limited according to the requirement then the performance of decision tree is an added advantage to the understandability and simplicity.

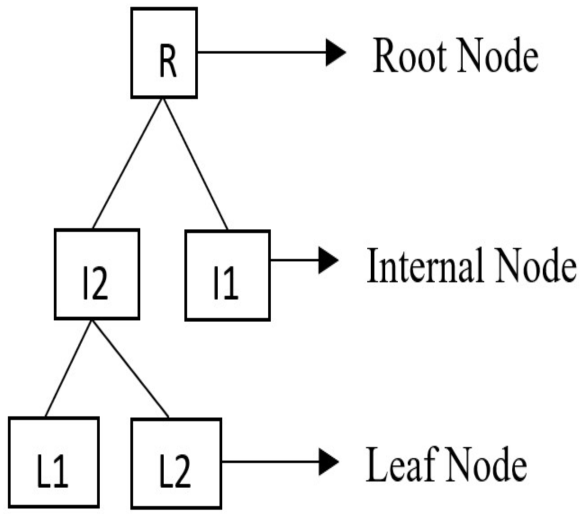

Fig. 7. Decision Tree Nodes

The Table I gives a summary of models used in the emotion classification studies.

TABLE I :SUMMARY – Research Works

Author	Data	Method	Evaluation
Neviarouskaya et.al [65]	Text Message	Rule-Based	Human Annotator
Ho and Cao [66]	ISEAR	Hidden-MarkovModel	cross-validation
Mishne [56]	Blog	SVM	Accuracy
Strapparava et.al [67]	News headlines	Knowledge and Corpus Based	Precision,Recall,F-Measure
Gupta et.al [68]	Customer care e-mail	Boostexter	Precision,Recall,F-Measure
Danisman and Alpkocak [57]	ISEAR	SVM	Kappa,F-Measure,Accuracy
Hancock et.al [69]	Text message	Survey	NA
Kim et.al [64]	SemEval2007,ISEAR,Fairy tales	VSM	Precision,Recall,F-Measure
Chaumartin et.al [70]	News headlines	Rule-Based Lingusitic	Precision,Recall F-Measure,Accuracy
Jain et.al [63]	ISEAR,WordNet-Affect,WPARD	VSM	NA
Lei et.al [71]	News Articles	Lexicon	
Mishne et.al [72]	blog	Pace Regression module WEKA	Cross Validation
Wang et.al [61]	Twitter data - hashtag	LIBLINEAR,Multinomial Naive Bayes	Precision,Recall,F-Measure
Mohammad et.al [73]	Twitter data - hashtag	SVM SMO	Precision,Recall,F-Measure
Hasan et.al [74]	Twitter data - hashtag	Emotex	Accuracy
Roberts et.al [58]	Twitter data	SVM	Precision,Recall,F-Measure
Bollen et.al [75]	Twitter data	POMS	Timeline of events
Purver et.al [59]	Twitter - Emoticons/Hashtag	SVM linear kernel	Precision,Recall,F-Measure

EMOTION MODELS AND THEORIES

Two major categories have been classified by the theorist. The first category demonstrates that emotions are discrete and second category states emotion is characterized as dimensional. Discrete emotion theory Fig.8. states that specific core emotions are sub served by independent neural system, on the other hand dimensional model Fig.9. states that all affective states or emotion arise from cognitive interpretations of core neural sensations [76]

6 basic forms has been characterised by Ekman [77].The forms are sadness, disgust, enjoyment, fear, anger and surprise. Plutchik [78] was agreed with Ekman's biologically driven perspective but also developed the wheel of emotions on bipolar axes: joy versus sadness, surprise versus anticipation ,anger versus fear and trust versus disgust. Shaver [79] introduced a hierarchical tree structure for the basic emotions like love, joy,fear surprise, anger and sadness and the leaves of the tree contains further categorization for each of these six basic emotions. Lovheim [80] present a three-dimensional model for monoamine neurotransmitters and emotions. According to his model, the monoamine systems are represented as orthogonal axes and the eight basic emotions, labeled according to Tomkins [81] [82], are placed at each of the eight possible extreme values, represented as corners of cube. There are also many other dimensional models for emotion, following are the widely accepted models as suggested by Rubin and Talarico [83]: circumplex model, vector model, and Positive Activation - Negative Activation model.

Fig. 8. Discrete Model

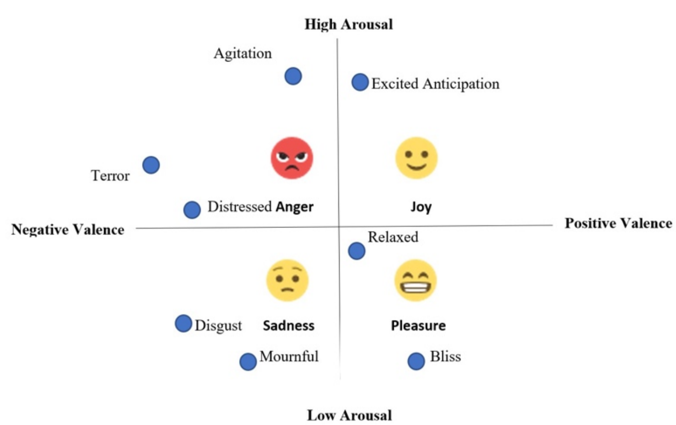

Fig. 9. Dimensional Model

According to author Russell in his article [84], circumplex model suggests that emotions are distributed in a two dimensional space, containing arousal and valence dimensions. Author Bradley [85], proposed a two-dimensional model in which the base dimension is arousal and that the valence determines the direction in which the emotion lies. Watson in his article [86], develop the Positive Activation - Negative Activation model in which there was a suggestion that positive affect and negative affect are the two separate systems.

EMOTION DETECTION AND OPINION MINING

This section provides a overview of existing research works related to emotion classification from text. It is divided into four subsections where, the first section elaborates few existing methods on general text classification of emotion and opinion, the second section describes detail studies with respect to social media text , the third section depicts emotion detection in the education system specifically Student Survey data, and the last section talks about the emotion mining in context to Business World specifically Business data.

A. Text Emotion Mining

Emotion mining has gain importance in the field of computer science due to the vast variety of systems that is useful in the field of remote healthcare system, customer services,smart phones that reacts on users's emotion [87].Emotion Analysis is a means of identifying different distinct human emotion types such as sadness ,joy , anger, depression. "Emotion Analysis," , "emotion detection," and "emotion identification" are all phrases that are sometimes used correspondently [88].

Author Seyeditabari, Tabari, and Zadrozny in their paper [89]proposes essential need to improve the design and architecture of current systems detecting emotion in text.The paper explains to pay attention to the linguistic intricacies of emotion expression, as human emotions are complex.

B. Social Media Mining

In today's world human perception [87] is not limited to only few fields like education, business,job, etc , but it has broaden its wings in respect to emotions.In the most recent years of 21st century almost every industry or company or institution is undergoing some digital transition, resulting in vast amounts of structured and unstructured data. The enormous task for researchers is to transform unstructured data into meaningful insights that can help them in decision-making.The opinions or sentiments or emotions mined from social network data helps understand the present state of the user. It does not directly provide intuitive insights on what actions to be taken to benefit the end user or business. Actionable Pattern Mining method provides recommendations on what changes or actions are required in order to benefit the end user.

In [90]paper analyze the sentiment of twitter data through mining actionable patterns via action rules. They implemented the experiment on both single machine and a cloud distributed environment for scalability purpose and compared the results with single machine implementation, distributed Hadoop MapReduce framework and Spark system.Their approach showed how with the volume of Twitter data, the processing of the proposed algorithm runs faster Spark system than on Hadoop system and single machine.The paper also suggested how users could increase theirs friends, favorites, and followers count in twitter.

In [91] paper the authors proposed automatic classification of emotions in twitter data using Recurrent Neural Network - Gated Recurrent Unit. The experiment provided them the result of training accuracy of 87.58 percent and validation accuracy of 86.16 percent.In adition to that they also extracted action rules with respect to the user emotion, that provided them the actionable suggestion on how to enhance emotion from negative or neutral to more positive emotion for the user. In [92]sentiment analysis there are two approaches to determine the polarity one is Corpus-based, and the other is Lexicon-based. The paper used corpus-based sentiment analysis approach, where sentiment values are generated based on sentence structure rather than words. Results showed how actionable patterns in tweet data are and also provided suggestions onto change sentiment of the twitter text to a more positive one. They compared the results with single machine implementation and distributed Hadoop MapReduce framework, where the experiment showed that processing of the algorithm runs faster on distributed environment than on single machine. The method can scale to accommodate large social media data size and also in future they can augment the data set with more syntactical parts including nouns and adjectives and to build lexicons for specific subjects:industry, medical, finance.

In the paper [87] automatically detect user emotion from the twitter data using the NRC Emotion Lexicon [93], [94] inorder to label the Emotion class for the data being provided. The paper talks about using the Support Vector Machine Linear implementation along with the features that includes word n-grams, character n-grams, brown word clusters, and parts of-speech tags and achieve a 10 percent improved accuracy compared to their previous method [95].

C. Education-Opinion Mining

The inception of big data in educational contexts has led to a new data-driven approach that is able to support informed decision making and efforts to improve educational effectiveness [96].Detecting Emotion in the education sector, plays a critical role for both Professors and Students. Moreover improved Emotions suggest a better student Learning experience [97].Taking timely feedback from students is the most effective technique for a teacher to improve teaching approach [98].Educational Data Mining(EDM) and Learning Analytics(LA) [99] are two interdisciplinary communities of computer scientists, learning scientists, psychometricians, and researchers from other areas with the same objective of improve learning starting from data. According to Author Romero et al. EDM/LA will soon become a mature area that will be widely used not only by researchers but also by instructors, educational administrators, and related business from all over the world. EDM/LA has impacted our understanding of learning and produced insights that have been translated to mainstream practice or contributed to theory. In their paper they mentioned some specific data mining methods as follow: Causal mining ,Clustering, Discovery with models, Distillation of data for human judgment, Knowledge tracing,Nonnegative matrix factorization, etc.One of the interesting fact regarding the future work of the paper is they are trying to analyse and mine data that are directly gathered from students' brain for a better understanding of the learning, which will give advantage to field of human neuroscience and pervasive neurotechnology. In paper [100] the author proposed an extension of their previous method [101] for assessing the effectiveness of the Lightweight team teaching model, through automatic detection of emotions in student feedback in computer science course by using Neural Network model. Neural Networks have been widely used for its high performance in variety of tasks by not limited to Text Classification and Image Classification. It is highly deemed to work great with huge volume of data. They further discussed how sequential model could be use with smaller data sets inorder to compare to the baseline models such as Support Vector Machines and Naive Bayes, to provide better results.They noticed the neural networks yields (76.7 percent) similar performance to traditional text classification models like Naive Bayes (74.79 percent) and Support Vector Machine (77.97 percent),with small size data set,which is considered to be a drawback when using neural networks for classification.In future the authors plan to extended their work by collecting student survey to identify actionable patterns, that ultimately help to improve the teaching model, learning environment to a better state.

Each year number of students enrolling in higher education is increasing significantly. [97]Students from diverse backgrounds can be found in almost all classes. Specially, for International Students online research is Considerably important to learn more about the potential institution, courses and professors [87]. They use blogs and other discussion forums to interact with students who share similar interests and to assess the quality of possible universities. In this way Emotion Detection can help the student to select the best institute or professor in their registration process [102].

Even now-a-days Student evaluation has become increasingly recognized feature for the institutional assessment of all matters related to teaching that includes teachers, teaching and academic programs. These evaluation results has advantage in certain fields apart from providing meaningful insights to improve the learning experience of students it also help the college administration in making faculty related decisions like retention, promotion and future employment. These days almost all educational institutions, including Schools and Universities collect the course evaluation feedbacks from students at the end of the semester. The evaluation usually consists of two types of questions: Quantitative and Qualitative type questions. In quantitative evaluation it can be noticed that the feedback taken is performed in terms of measurable outcomes that includes a [103]Likert-type scale to capture the level of agreement and disagreement, whereas, in qualitative evaluation the students can directly convey their feelings, thoughts/opinions or suggestions about the course, the course instructor, or their overall comments towards the course. In this way the qualitative feedbacks provide freedom for the students to express their honest thoughts on a course. The data collected in the qualitative form provides deeper insight into a student's emotional state of mind. [104] In this paper they mainly focused on mining the qualitative student feedbacks and then analyzed the student sentiments. They also realized the efficiency of Flipped Classroom approach and Light Weight teams which are called Active Learning methods. Results showed positively, where the implementation of these Active Learning methods is directly proportional to positivity in student emotions.By using Actionable Pattern Discovery methods the student Emotion can be changed from Negative to Positive and from Neutral to Positive [97].

In paper [105]it shows an evident of improving teaching style, material provided for class, along with that new learning methods were also being adopted.The authors used Actionable patterns that provides suggestion in the form of rules to help the user achieve better outcomes. The proposed method provides meaningful insight in terms of changes that might can be embrace in the Light-Weight team activities and also in which [101] team members have little direct impact on each other's final grades, with significant long-term socialization.But,this study showed some limitation regarding the number of participants, as it was conducted in a single institution with limited number of participants in courses offered in computer science.

D. Business-Opinion Mining

Knowledge discovery technique provides plenty of patterns, presented in the form of rules. Identifying those useful and interesting patterns is a tedious and time consuming process.One of the interesting question arises is:whether or not the pattern can be used in the decision making process of a business to increase profit.Action rules may suggest actions to be taken based on the discovered knowledge. In this way it contributes to business strategies and various scientific researches.The author in paper [106]focused on decreasing the space of action rules through the process of generalization,where, generalization algorithm used produces summaries by maximizing the diversity of rule pairs,and minimizing the cost of the suggested actions.They tried to present a new method for computing the lowest cost of action rules and their respective generalizations. While doing so they discovered action rules of lowest cost by taking into account the correlations between individual atomic action sets.They are suggesting to explore the method with short descriptions of action rules or summaries, and the use of hierarchical attributes.The paper also talks about the possible future work that could be done like employing a more generic approach for creating summaries, which would allow using non-hierarchical attributes as well. According to them, taking intervals with numerical values, or a subset for non-numerical ones could be a good one for future . Their work is applicable in the field of business, financial, medical and industrial.

Business often depends on the customer feedback and reviews, in the era of 21st century. Emotion detection or sentiment analysis helps to identify and extract information about the sentiment or emotion of the product or topic or document. Attribute selection is a challenging problem especially with large datasets in actionable pattern mining algorithms. In [107] paper the author present a Lexicon based weighted scheme approach to identify emotions from customer feedback data in the area of manufacturing business.Further they worked with Rough sets and explore the attribute selection method for large scale datasets. Then they applied Actionable pattern mining to extract possible emotion change recommendations.This kind of recommendations help business analyst to improve their customer service which leads to customer satisfaction and further increase sales revenue.

The author of the paper [108] put a discussion on big data mining for customer insights. The article covers two research methods.The first method is the systematic search in bibliographic repositories that focuses on identifying the concepts of big data mining for customer insights. The first method has been conducted in four steps as follows: search, selection, analysis, and synthesis. The second research method is the bibliographic verification of the obtained results. The Bibliographic verification consisted of querying the Scopus database with previously identified key phrases and then it performs trend analysis on the revealed Scopus results. The main contributions of this study are as follows:first to organize knowledge on the role of advanced big data analytics (BDA), mainly big data mining in understanding customer behavior; second to indicate the importance of the temporal dimension of customer behavior; and lastly to identify an interesting research gap: mining of temporal big data for a complete picture of customers.

The paper [109]reviewed impacts of social media and stock market, The authors classified stock market related tweets in two different ways; using Amazon Mechanical Turk, and a classification model with accuracy of 79.9 percent. Then the authors used these two sentiment scores and stock market returns to understand the causality between datasets. Granger Causality analysis of these two tweet datasets with various stock returns has shown that for many companies there is a statistical significant causality between stock and the sentiments driven from tweets. Finally they evaluate tweets sent by verified accounts in specific dates, the results showed that when stock return has a jump due to news regarding the stock, the amount of tweets sent on Twitter jumps in the same direction,thus adding value to the granger causality analysis.

LEXICONS

Lexicon is the words used in a language or by an individual speaker or group of speakers or a subject according to Merriam Webster dictionary.In text mining *lexicon* calculating the sentiment from the semantic orientation of word or phrases that occur in a text Lexicons are a useful resource in text mining specially for tasks like sentiment analysis. This section depicts some of the widely used lexicons in the sentiment analysis research works.

A. Affective Norms for English Words - ANEW

ANEW is developed by Centre for Emotion and Attention(CSEA) at University of Florida. They used Self Assessment Manikin (SAM) an affective rating system (nonverbal pictorial assessment) to assess the three dimensions of pleasure, arousal, and dominance with students from introductory psychology class. ANEW [85] provides a set of a normative emotional rating for many English language words. The rating is in terms of arousal, pleasure, and dominance that complements the existing International Affective Picture System(IAPS) [110] and International Affective Digitized Sounds (IADS) [111] which are collections of picture and sound stimuli respectively. One major advantage of ANEW is that it has been validated by several persons which makes it preferable for psycho-linguistic studies.

B. AFINN

Finn Arup Nielsen [112] in between 2009 to 2011 created
AFINN lexicon, that covers a list of English words including few phrases scored for valence in the range from -5(very negative) to +5(very positive) manually,with a mean of 0.59. The word list includes few positive words, words from public domain "Original Balanced Affective Word List" by Greg Siegle obscene words, internet slang words from urban dictionary including acronyms, and recent additions from "The Compass DeRose Guide to Emotion Words" by Steven J. DeRose. The initial version AFINN-96 includes 1468 unique words, including few phrases.As of today, the latest version AFINN-111 has 2477 unique words and phrases. Though AFINN has slightly better performance than ANEW, for studies in scientific psycholinguistics' ANEW is preferable as the scoring is validated across several persons worldwide.

C. Linguistic Inquiry Word Count - LIWC

LIWC is an emotion lexicon (program + dictionary) that contains close to 5000 words by Tausczik et al. [113] that can calculate the percentage of emotional word categories within a text. Initially there was a list of emotion word categories from dictionaries, thesauruses, questionnaires,etc, made by research assistants. Later group of three judges, independently validated the word and emotion category for each word in the initial list. After validation by the judges the word list was updated using set of rules as follows: word retained if two out of three judges agreed, word deleted if two of three judges agreed for the word to be excluded, word added to list if two of the three judges agreed for the same. A separate group of judges will again conduct the entire process. Initial judging happened between 1992 and 1994. Later a significant revision happened in 1997 and in 2007 to streamline the original program and dictionaries.

D. NRC Emotion Lexicon

Mohammad and Turney [114] [115] [116] create the NRC emotion lexicon called EmoLex with 14200 words. To generate the lexicon, they first collect list of words from sources as follows:Macquarie Thesaurus uni-grams and bi-grams that frequently occur in Google n-gram corpus, words with positive and negative semantic orientation from General Inquirer, emotion words from WordNet-Affect lexicon. Turker's through Amazon's Mechanical Turk annotated the word list based on the emotion they evoke. The author perform series of validations to make sure that the manual annotations are right by using additional word choice questions for Turker's. This effort helped them identify the annotators are not familiar with the word and ignore results from such annotators. They provided a prove that annotations by crowd-sourcing are of high-quality by comparison with gold standards.

E. NRC Emotion Hashtag

Saif M. Mohammad [117] created NRC emotion hashtag association of 16,862 words with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust generated automatically from tweets with emotion-word hashtags like happy and anger.In there paper they began with experiments for six basic emotions and showed that the hashtag annotations are consistent and matched with the annotations of trained judges.

F. WordNet-Affect

WordNet-Affect lexicon is an emotional lexicon for affective knowledge.Authors Strapparava and Valitutti [118] [119] [120] develop this resource WordNet-Affect on top of WordNet through the selection and labeling of synsets representing an affective concepts. They create an AFFECT database with 1093 terms directly or indirectly referring to emotional states, started with adjectives and then extended it by adding nouns, verbs, and adverbs. This AFFECT database was then projected to the WordNet-Affect as an affective label. WordNet-Affect contains 4,787 words and 2,874 synsets . An important advantage of this affective lexicon concerning mainly adjectival interpretation is the stative/causative dimension. An emotional adjective is said causative if it refers to some emotion that is caused by the entity represented by the modified noun (e.g. amusing food). In a similar way, an emotional adjective is said stative if it refers to the emotion owned or felt by the subject denoted by the modified noun (e.g. happy/cheerful child).

G. Clean Balanced Emotional Tweet - CBET

Shahraki and Zaiane [121] develop a dataset especially for Twitter data called CBET datset. They use hash-tags to collected 208,544 general-purpose tweets. After pre-processing they selected 3,000 sample tweets for each emotion category and created a dataset with 27,000 samples called the Clean Balanced Emotional Tweet dataset. The lexicon is V x E matrix, where the word at index (j,i) denotes the degree that the word w_j express the emotion e_i . Each word w_j has a corresponding weight vector calculated as the number of times the word w_j has occurred in tweets that have label e_i as emotion. This CBET is a new lexicon that has the ability to overcome some of the disadvantages of previous emotion lexicons.

ACTION RULE MINING

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 two rules,Rule-Based and Object-Based approaches to generate the larger sets Action Rules much faster.

A. Rule-Based Action Rule Mining

Rule-Based Action Rule mining approach extracts intermediate classification rules using Learning Based on Rough Sets (LERS) approach [122] and then uses it to extract classification rules from complete information system or using the Extracting Rules from Incomplete Decision (ERID) approach [123]. 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 [124], [125] and methods that generate Action Rules from single classification rule like the Action Rules Based on Agglomerative Strategy method [126].

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 [127] or using the Association Action Rules [27] without extracting intermediate classification rules.

ACTIONABLE PATTERN DISCOVERY

The property of the discovered knowledge is called *Actionability*. Patterns are often considered Actionable if the user can act upon them, and if this action can help them to accomplish their final goals in short benefiting the user. Author Cao in his paper [128] delve into the paradigm transfer of knowledge discovery from data to actionable knowledge discovery and delivery. Cao observes two perspectives one is micro-level (technical and engineering issues) and other is macro-level (methodological and fundamental issues). These perspectives lead to narrow down the gaps between delivered knowledge and desired knowledge and states that Actionable Knowledge Discovery and Delivery (AKD) framework help narrow the breach in Knowledge discovery process. The author further proposes that use of domain knowledge in the data mining process and engaging organizational and social intelligence in the KDD modeling process promotes the paradigm transfer.

In paper [129]authors Barrett et al. extract actionable knowledge from the larger dataset collected in schools that could be valuable to students, teachers, principals, district, state and national administrators. Authors Greco et al. [130] advocates patterns discovered from data are represented in the form of *'if ..., then...'* rules known as decision rules. These patterns provide information about past events and utilized for future decisions. In the case of medical diagnosis these rules can help identify the relationship between symptoms and sickness and also diagnose new patients based on these past records. Another essential prospective usefulness of decision rules is getting the desired effect on dependent variables by building strategy of intervention on the independent variables.For instance,in medical example, this can be explained as modifying symptoms to get out from the sickness.

As mentioned in the section above Action Rules Mining is a method that discovers Actionable Patterns from large data. Action Rules are rules that reclassify data from one category to another describe in a simpler language it is the possible transition of data from one state to desired one [131].

According to author Dardzinska [132], the frameworks for generating Action Rules from [133] are as follows: loosely coupled and tightly coupled. The loosely coupled framework is known as rule-based. Rule-based pairs certain classification rules which have to be discovered first by using for instance algorithms such as LERS [122] or ERID [123] [134]. The tightly coupled framework is known as object-based and it assumes that Action Rules are directly discovered from a database [135] [136] [127]. Traditional methods for discovering them follow algorithms either based on frequent sets (called action sets) and association rules mining [137] or they use algorithms such as LERS or ERID with atomic action sets used as their initial step. Action Rules are one way to mine Actionable knowledge from enormous data.

A. Action Rules Assumptions

Author Dardzinska [132],summarize the Action Rules, introduced by author Ras and Wieczorkowska [138] may be utilized by any type of industry maintaining huge databases, especially education,medical,business, and military,. They are constructed from classification rules which suggest ways to re-classify objects, such as students,patients, customers , or soldiers to a desired state. However, such a change cannot be done directly to a chosen attribute. Therefore, generally there is a need to learn definitions of such an attribute in terms of other attributes. The following describes the definitions of action terms, Action Rules, and their standard interpretation.

Atomic action term, means an expression $(a,a_1 \rightarrow a_2)$, where a is an attribute, and $a_1,a_2 \in V_a$.

If $a_1 = a_2$ then a is called stable on a_1. In this case, for simplicity reason, we use notation (a,a_1) instead of $(a,a_1 \rightarrow a_2)$. definition Set of action terms, mean the smallest set such that:

1. If t is an atomic action term, then t is an action term.
2. If t_1, t_2 are action terms, then $t_1 * t_2$ is an action term.
3. If t is an action term containing $(a,a_1 \rightarrow a_2)$, $(b,b_1 \rightarrow b_2)$ as its sub-terms, then a 6= b.

definition By the domain of an action term t, denoted by $Dom(t)$, we mean the set of all attribute names listed in t. definition By an Action Rule we mean an expression $r = [t_1 \rightarrow t_2]$, where t_1 is an action term, and t_2 is an atomic action term.

Additionally, we assume, that $Dom(t_2) = \{d\}$ and $Dom(t_1) \subseteq A$, where A is a set of attributes. The domain $Dom(r)$ of Action Rule r is defined as $Dom(t_1) \cup Dom(t_2)$.

Table Ii: Information System S

X	Attribute a	Attribute b	Attribute c	Attribute d
x1	a1	b1	c1	H
x2	a2	b2	c1	H
x3	a2	b1	c1	A
x4	a1	b1	c2	A
x5	a2	b1	c2	A
x6	a2	b2	c2	H
x7	a1	b1	c2	A
x8	a1	b2	c1	A
x9	a1	b1	c1	H
x10	a2	b2	c1	H

Consider the information system S in Table II. The following are examples of atomic action terms: $(a,a_2 \to a_2)$, $(b,b_2 \to b_1)$, $(c,c_2 \to c_2)$, $(c,c_3 \to c_3)$, $(d,H \to A)$. Also, consider the following expression in equation (4), (5), and (6). The values of a_2,c_2,c_3 of attributes a and c remain unchanged, while $(b,b_2 \to b_1)$ means that the value of attribute b is changed from b_2 to b_1.

$$(a,a_2 \to a_2) = (a,a_2) \quad (4)$$

$$(c,c_2 \to c_2) = (c,c_2) \quad (5)$$

$$(c,c_3 \to c_3) = (c,c_3) \quad (6)$$

Equation (7) and (8) are example Action Rules. According to equation (7) r_1 says that if the value a_2 remains unchanged and value b will change from b_2 to b_1 for a given object x, then it is expected that the value d will change from H to A for object x. Clearly, $Dom(r_1) = \{a,b,d\}$. In a similar way, the rule r_2 in equation (8) says that if the value c_2 remains unchanged and value b will change from b_2 to b_1, then it is expected that the value d will change from H to A, and $Dom(r_2) = \{b,c,d\}$.

$$r_1 = [((a,a_2 * (b,b_2 \to b_1)) \to (d,H \to A)] \quad (7)$$

$$r_2 = [((c,c_2 * (b,b_2 \to b_1)) \to (d,H \to A)] \quad (8)$$

definition Standard interpretation N_s of action terms in S = (X,A,V) is defined as follows:

1. If $(a,a_1 \to a_2)$ is an atomic term, then

$N_s((a,a_1 \to a_2)) = [\{x \in X : a(x) = a_1\},\{x \in X : a(x) = a_2\}]$

2. If $t_1 = (a,a_1 \to a_2) * t$ and $N_s(t) = [Y_1,Y_2]$, then

$N_s(t_1) = [Y_1 \cap \{x \in X : a(x) = a_1\}, Y_2 \cap \{x \in X : a(x) = a_2\}]$.
Now we define $[Y_1,Y_2] \cap [Z_1,Z_2]$ as $[Y_1 \cap Z_1,Y_2 \cap Z_2]$.
Assume that $N_s(t_1) = [Y_1,Y_2]$ and $N_s(t_2) = [Z_1,Z_2]$.

Then $N_s(t_1 * t_2) = N_s(t_1) \cap N_s(t_2)$. Let $r = [t_1 \to t_2]$ be an Action Rule, where $N_s(t_1) = [Y_1,Y_2]$, $N_s(t_2) = [Z_1,Z_2]$. definition By support and confidence of rule r we mean:

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

Now, let us consider the equation (7) for support and confidence with example. For the rule r_1 we have:

- $N_s(a,a_2 \to a_2) = [\{x_2,x_3,x_5,x_6,x_{10}\},\{x_2,x_3,x_5,x_6,x_{10}\}]$
- $N_s(b,b_2 \to b_1) = [\{x_2,x_6,x_8,x_{10}\},\{x_1,x_3,x_4,x_5,x_7,x_9\}]$

- $N_s(a,a_2 \to a_2)*(b,b_2 \to b_1) = [\{x_2,x_6,x_{10}\},\{x_3,x_5\}]$
- $N_s(d,H \to A) =$

$[\{x_1,x_2,x_6,x_9,x_{10}\},\{x_3,x_4,x_5,x_7,x_8\}]$ Therefore, for rule r_1, support $sup(r_1) = 2$, confidence $conf(r_1) = \frac{3}{3} \cdot \frac{2}{2} = 1$.

B. Action Rule

The expression r = $[t_1 \to t_2]$ 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 III. $[B_1 \wedge C_1 \wedge (F, F_3 \to F_1) \wedge (G, \to G_1) \to (D, D_2 \to D_1)$.

C. 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.

Table Iii: Information System Z

X	A	B	C	E	F	G	D
$x1$	A_1	B_1	C_1	E_1	$F2$	G_1	D_1
$x2$	A_2	B_1	C_2	E_2	$F2$	G_2	D_3
$x3$	A_3	B_1	C_1	E_2	$F2$	G_3	D_2
$x4$	A_1	B_1	C_2	E_2	$F2$	G_1	D_2
$x5$	A_1	B_2	C_1	E_3	$F2$	G_1	D_2
$x6$	A_2	B_1	C_1	E_2	$F3$	G_1	D_2
$x7$	A_2	B_3	C_2	E_2	$F2$	G_2	D_2
$x8$	A_2	B_1	C_1	E_3	$F2$	G_3	D_2

D. Action Rules from Classification Rules

Author Dardzinska [132] declares the Action Rules loosely coupled framework as follows: Finding useful and essential rules are really very important and extremely interesting task of knowledge discovery in database. Many researchers mainly focus on techniques for generating patterns, such as classification rules or association rules, from the given data sets. Researchers assume that the user should analyze these patterns and infer actionable solutions for specific problems within given domains. The classical knowledge discovery algorithms can identify large number of significant patterns from data. Therefore, people are overwhelmed by a enormous number of uninteresting patterns which are very difficult to analyze and is time consuming too in regards of giving solutions. For that reason, there are some ongoing research that look for new methods and tools, that has the ability to assist people in identifying rules with useful knowledge. There are two types of interestingness measure: one is subjective and the other is objective [139] [140] [141]. Subjective measures includes actionability [139] and unexpectedness [141]. When a rule contradicts,uncovers, or surprises new knowledge, it is classified as unexpected. A rule is deemed actionable, if the user can take action to gain an advantage based on this rule. Experts of this domain basically look at a rule and point out that this rule can be converted into an appropriate action.However, an objective measure is a data-driven approach for evaluating the quality of association patterns. It is domain-independent and requires minimal input from the user-end, other than to specify a threshold for filtering low-quality patterns [142]. An objective measure is generally computed based on the frequency counts that is tabulated in a contingency table.

E-Action Rules mining is a approach helping people in an automatic and intelligent way to obtain useful information from data. This acquired information can then be turned into actions. The method provides suggestions about how to change certain attribute values of a given set of objects to reclassify them according to final desire of the user.

Mining actionable knowledge has mainly two frameworks as follows: rule-based and object-based [133]. In the former one that is in rule-based approach [138] , the extraction of actionable knowledge is a consequence of using classification rules discovery, while in the later that is for object-based approach, Action Rules are extracted directly from a database [135] [136] [127].Rule-based is further subdivided into: methods generating Action Rules from certain sets of classification rules [143] [144] [145] [125], and methods generating Action Rules from single classification rules [146].Like for example, ARAS algorithm proposed in [144] generates groups of terms (produce from values of attributes) around classification rules and constructs the Action Rules directly from them.

Action Rules mining algorithms(mostly all) does not assure that the discovered patterns in the first step will show the path to achieve actionable knowledge that leads to maximizing benefits.Researchers found one way to handle this problem and that is to assign a cost function to all the variations or changes of the attribute values [147]. If variations or changes of attribute values in the classification part of an Action Rule are too complex, then they can replaced it by composing such rule with other Action Rules, as proposed by the authors in their paper [145]. A new Action Rule Each is defined uniquely with composition of these rules . The cost for objects supporting each new Action Rule of reassigning them is lower for the new rule while,the objects supporting each new Action Rule is same as objects supporting the Action Rule replaced by it.
E-Action Rule improve the actionability concept far better than Action Rule does [138]. E-Action Rules introduces a notion of its supporting class of objects and are created from certain sets of classification rules. They are constructed in such a manner that they not only help to evaluate discovered patterns but also reassign some objects in a dataset from one state into another that is more desirable.

For example, classification rules received from a grocery store's data can be very useful to describe who is good client (whom to offer some additional promotions) and who is bad client (whom to watch to minimize loses). But, if shop managers need to improve their understanding regarding customer's feedback and seek for specific actions to improve the services, only classification rules are not sufficient. New method is introduced using classification rules for connecting with action based on their condition features in order to get a desired effect on their decision feature. When one look into the grocery store's example again, the strategy of action would consist of reshaping some condition features to modify the general understanding of customers behavior and then improve the services.

The advantage of E-Action Rules prevails in many other fields,that includes medical diagnosis. In medical diagnosis field, e.g. in lactose intolerant problem, classification rules can explain the relationships between symptoms and sickness and help to predict the diagnosis of a new patient having the same problem. E-Action Rules helps in providing a hint to the doctor what symptoms have to be eliminated or modified in order to recover a certain division of patients with better prognoses in their illness.

System DEAR

Author Dardzinska [132],in his research talks about an algorithm known as system DEAR. The algorithm aim to move objects between classes by altering some flexible attributes. System DEAR identifies the optimal set of attributes needs to be altered, the new values, and computes support and confidence for each of the extended Action Rule.

Assuming a decision system with only one decision attribute, as seen in the method of treatment. Its domain contains integers values. This decision attribute classifies objects (patients) with respect to the prognoses for patients. The cardinality of the image d(X) = {di : d(x) = di for some x ∈ U} is called the rank of attribute d and is denoted by r(d):

Observe that the decision d determines the partition

CLASSs(d) = {X1,X2,,Xr(d)} of the group of objects X, where Xk = d−1({di}) for 1 ≤ di ≤ rd. CLASSs(d) is called the classification of objects in S determined by the decision d. As mentioned earlier, objects correspond to patients. Also, the patients in d−1({d1}) are better prognoses for a hospital than patients in d−1({d2}) for any d2 ≤ d1. The set d−1({r(d)}) represents the patients with prognoses for complete recovery. Clearly the aim of any medical centers or hospital is to maximize the number of recovered patients. It can be achieved by transferring some patients from group d−1({d(2)}) to d−1({d1}), for any d2 ≤ d1. Namely, through special methods of treatment offered by medical centers, values of flexible attributes of some patients can be altered and all these patients can be shift from a group of worse prognoses ranking to a group of best prognoses for them.

Now let us assume that for any two collections of sets X,Y , we write, X ⊆ Y if (∀x ∈ X)(∀y ∈ Y)(x ⊆ y).

Let S = (X,ASt ∪ AFl ∪ {d}) be a decision table and B ⊆ ASt ∪ AFl. We say that attribute d depends on B if CLASSs(B) ⊆ CLASSs(d), where CLASSs(B) is a partition of X generated by B [148]. definition Assume that attribute d depends on B where B ∈ ASt ∪AFl. The set B is called d-reduct in S if there is no proper subset C of B such that d depends on C.

The concept of d-reduct in S was first brought into attention in order to induce rules from S describing values of the attribute d depending on minimal subsets of ASt ∪AFl which preserve the confidence of extracted rules. In order to induce rules in which the then part consists of the decision attribute d and the if part consists of attributes belonging to ASt∪AFl, sub-tables (X,B ∪d) of S where B is a d-reduct in S should be used for rules extraction.

Considering the information system S in the table below Table IV. The set a,c is the set of stable attributes, b is flexible attribute and d is the decision attribute. Also, the author [132] assume that H denotes patients of good prognoses and L denotes patients of week prognoses.

Table Iv: Dear - Information System S

X	Attribute a	Attribute b	Attribute c	Attribute d
x1	a0	b3	c0	L
x2	a0	b2	c1	L
x3	a0	b3	c0	L
x4	a0	b2	c1	L
x5	a2	b1	c2	L
x6	a2	b1	c2	L
x7	a2	b3	c2	H
x8	a2	b3	c2	H

$L(r)$ means all attributes listed in the conditional part of rule

r. For instance, consider the rule below, then $L(r) = \{a,b\}$. • $r = [(a,a_1) * (b,b_2) \to (d,H)]$ $d(r)$ denote the decision value of a rule. In the above example $d(r) = H$. If r_i,r_j are rules and $B \subseteq A_{St} \cup AFl$ is set of attributes, then $r_i/B = r_j/B$ means that the conditional parts of rules r_i,r_j restricted to attributes B are the same. For instance consider the rules r_i,r_j below, then $r_i/(a,c) = r_j/(a,c)$.

- $r_i[[(a,a_1) * (b,b_0) * (c,c_3) \to (d,H)]$
- $r_j[[(a,a_1) * (b,b_2) * (c,c_3) \to (d,H)]$

Based on below CLASS values, $CLASS(\{a,b\}) \subseteq CLASS_s(d)$ and $CLASS(\{b,c\}) \subseteq CLASS_s(d)$.

- $CLASS_s(\{d\}) = \{\{x_1,x_2,x_3,x_4,x_5,x_6\},\{x_7,x_8\}\}$,
- $CLASS_s(\{a\}) = \{\{x_1,x_2,x_3,x_4\},\{x_5,x_6,x_7,x_8\}\}$,
- $CLASS_s(\{b\}) = \{\{x_1,x_3,x_7,x_8\},\{x_2,x_4\},\{x_5,x_6\}\}$,
- $CLASS_s(\{a,b\}) = \{\{x_1,x_3\},\{x_2,x_4\},\{x_5,x_6\}\{,x_7,x_8\}$,
- $CLASS_s(\{c\}) = \{\{x_1,x_3\},\{x_2,x_4\},\{x_5,x_6,x_7,x_8\}\}$,
- $CLASS_s(\{b,c\}) = \{\{x_1,x_3\},\{x_2,x_4\},\{x_5,x_6\}\{,x_7,x_8\}\}$

It can be easily checked that both $\{b,c\}$ and $\{a,b\}$ are dreducts in S. Rules can be directly derived from d-reducts and the information system S. In this example, the following optimal rules are obtained:

- $(a,a_0) \to (d,L)$
- $(b,b_2) \to (d,L)$
- $(b,b_1) \to (d,L)$ • $(c,c_0) \to (d,L)$
- $(c,c_1) \to (d,L)$
- $(a,a_2) * (b,b_3) \to (d,H)$
- $(b,b_3) * (c,c_2) \to (d,H)$

Assume now that $(a,v \to w)$ denotes the fact that the value of attribute a has been altered from v to w. Similarly, the term $(a,v \to w)(x_i)$ means that $a_{xi} = v$ has been changed to $a_{xi} = w$. In other words, the symptom (a,v) of a patient x_i has been altered to symptom (a,w).

Now, let us assume $S = (X,A_{St} \cup A_{Fl} \cup \{d\})$ is a decision table. Assume that rules r_1,r_2 have been extracted from S,B_1 is a maximal subset of A_{St} such that $r_1/B_1 = r_2/B_2$, $d(r_1) = d_1$, $d(r_2) = d_2$ and $d_1 \leq d_2$. Also, assume that $(b_1,b_2,...,b_p)$ is a list of all attributes in $L(r_1) \cap L(r_2) \cap A_{Fl}$ on which r_1,r_2 differ and

- $r_1(b_1) = v_1, r_1(b_2) = v_2, ..., r_1(b_p) = v_p$, • $r_2(b_1) = w_1, r_2(b_2) = w_2, ..., r_2(b_p) = w_p$.

By (r_1,r_2) Action Rule on $x \in X$:

- $((b_1,v_1 \to w_1)*(b_2,v_2 \to w_2)*...*(b_p,v_p \to w_p))(x) \to ((d,d_1 \to d_2))(x)$

The rule is valid if the value of rule on x is true or else it is false. Let $X_{(r1)}$ be the set of all patients in X supporting the rule r_1. If (r_1,r_2) Action Rule is valid on $x \in X_{(r1)}$ then the Action Rule supports the new profit ranking d_2 for object x.

Consider the rules in Table V. The rules representation is given as below:

Table V: Dear - Small Part Of Information System S

a-stable	b-flexible	c-stable	e-flexible	g-stable	h-flexible	d-decision
a1	b1	c1	e1			H
a1	b2			g2	h2	L

- $r_1 = (a_1 * b_1 * c_1 * e_1 \to H)$
- $r_2 = (a_1 * b_2 * g_2 * h_2 \to L)$

Assume that object x_i supports rule r_1 which means that it is classified as H. In order to re-classify x_i to class L, it is required to change the following values b from b_1 to b_2, $g_{(x)} = g_2$ and h to h_2. This is the meaning of the extended Action Rule (r_1,r_2) given below:

- $(a_1 * (b,b_1 \to b_2) * c_1 * e_1 * (g,g_2) * (h,\to h_2))(x) \to (d,H \to L)(x)$.

Support and confidence of the extended r_1,r_2 Action Rule is given as follows:

- $sup(r_1, r_2) = card[(a, a_1) * (b, b_1) * (c, c_1) * (e, e_1) * (g, g_2) * (d, H)]$.
- $conf(r_1, r_2) = \frac{card((a,a_1)*(b,b_1)*(c,c_1)*(e,e_1)*(g,g_2)*(d,H))}{card((a,a_1)*(b,b_1)*(c,c_1)*(e,e_1)*(g,g_2))} \cdot \frac{card((a,a_1)*(b,b_2)*(c,c_1)*(g,g_2)*(h,h_2)*(d,L))}{card((a,a_1)*(b,b_2)*(c,c_1)*(g,g_2)*(h,h_2))}$

For any extended (r_1,r_2) Action Rule support and confidence can be defined in a similar way.
System DEAR2

Authors Tsay and Ras in their work [125] summarize system DEAR2. System DEAR2 is an Action Tree algorithm helps to generate E-Action Rules. The algorithm finds the stable attribute with minimum number of values and use that attribute to split the set of rules re-iteratively. Once the stable attributes are processed, the final subsets are split further based on decision attribute [132]. This method construct an action tree which is used to generate E-Action Rules from the leaf nodes of the same parent. The algorithm DEAR consists of two main steps as described below.

Algorithm System DEAR2

Build Action-Tree

Divide the rule table R, taking into consideration all stable attributes.a) Find the domain Dom(a) of each attribute a ∈ A_{St} from the initial table R. b) Partition the current table into sub-tables containing only rules supporting values of stable attributes in the corresponding sub-tables. Build Action-Tree: Divide each lowest sub-table into new sub-tables containing rules with the same decision value. Build Action-Tree: Represent each leaf as a set of rules which do not contradict on stable attributes and also define decision value d_i. The path from the root to that leaf gives the description of objects supported by these rules. Generate Action Rules: Compare all unmarked leaf nodes of the same parent from Action Rules. Generate Action Rules: Calculate the support and confidence for each rule. If both are above minimal thresholds, the rule is extracted and added to knowledge base.

An Action-Tree consists of two types of nodes:non-leaf node and a leaf node. At a non-leaf node in the tree, the set of rules is partitioned along the branches and each child node gets its corresponding subset of rules. Leaf represent a set of rules, which define decision value d_i and do not contradict on stable attributes. The path from the root to that leaf gives the description of objects supported by these rules. This algorithm is explained with an example Decision System S in Table VI.

Let $A_{St} = \{a,b\}$ be the stable attributes and $A_{Fl} = \{c,d\}$ be the flexible attributes. The goal of system DEAR2 is reclassify objects from class (d,H) to class (d,A). Table VII shows the knowledge base of certain rules R extracted from Table VI. Sample certain rule for first row of Table VII is given in equation (9).

$$(a,a_2) \rightarrow (d,A) \quad (9)$$

Table Vi: Dear2 - Decision System S

X	Attribute a	Attribute b	Attribute c	Decision d
x1	a1	b2	c1	H
x2	a1	b2	c1	H
x3	a2	b1	c1	A
x4	a2	b1	c1	A
x5	a1	b2	c2	A
x6	a1	b2	c2	A
x7	a1	b0	c1	H
x8	a1	b0	c1	H
x9	a1	b0	c3	H
x10	a1	b1	c2	H
x11	a2	b2	c3	A
x12	a2	b0	c1	A

The algorithm starts with Table VII. Attribute a is used to split the table into two sub-tables Table VIII and Table IX defined by values a_1, a_2. The attribute a is chosen because $card[V_a] < card[V_b]$.

Table Vii: Knowledge Base T1

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x3, x4, x11, x12	a2			A
x1, x2, x7, x8	a1		c1	H
x7, x8, x9	a1	b0		H
x3, x4		b1	c1	A
x5, x6		b2	c2	A

Table Viii: Knowledge Base T2

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x1, x2, x7, x8	a1		c1	H
x7, x8, x9	a1	b0		H
x3, x4		b1	c1	A
x5, x6		b2	c2	A

It is evident that all objects in Table IX have the same decision attribute (d,A), therefore no Action Rules can be generated and this table is no longer considered for action. Table VIII has another stable attribute b and contains different decision values, it is divided further into Table X, Table XI, and Table XII.

There are no stable attributes at this point and Table X have same decision values, hence cannot be divided further. In Table XI there is only one value of flexible attribute $c = c_1$, so this table cannot be partitioned. Table XII is partitioned into two sub-tables Table XIII and Table XIV.

Thus, the path from root to the leaf described by d gives the information of rules and objects supported by them. For example, the path $[a = a_1]$,$[b = b_2]$ and $[d = H]$ leads to Table XIII. Similarly, the path $[a = a_1]$,$[b = b_2]$ and $[d = A]$ leads to Table XIV. The Action Rule generated by comparing the pairs of rules in the above-mentioned tables is given in equation (10). Support of rule r is given by $sup(r) = min\{4,2\} = 2$ and confidence is given by $conf(r) = 1.\frac{2}{3} = \frac{2}{3}$.

$$r = [[(a,a_1) * (c,c_1 \rightarrow c_2)] \rightarrow (d,H \rightarrow A)] \qquad (10)$$

The Action-tree algorithm proposed here requires the extraction of all classification rules from the decision system before any action rule is constructed and has $O(k^2)$ complexity in worst case, where k is the number of classification rules.

E. E-Action Rules - ARAS Algorithm

Author Dardzinska [132] conceptualize the facts of EAction Rules proposed by Ras et al. [149] for automatic analysis of discovered classification rules. E-Action Rules hint how to re-classify some objects in a data set from one state into different state that is more desired . Let $S = (X,A_{St} \cup A_{Fl} \cup d)$ be a decision system, where $d \notin A_{St} \cup A_{Fl}$. To improve the efficiency of the algorithm, when the number of attributes is large, we can extract rules from sub-tables $(X,B \cup d) \subseteq S$, where B is a d - reduct of the system S.

Assume that r_i,r_j are the rules extracted from S. The idea of extended Action Rule (e-action rule) was given by authors Tsay et al. [149] and extended by authors Ras et al. [150].

Table Ix: Knowledge Base T3

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x3, x4, x11, x12	a2			A
x3, x4		b1	c1	A
x5, x6		b2	c2	A

Table X: Knowledge Base T4

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x1, x2, x7, x8	a1		c1	H
x7, x8, x9	a1	b0		H

Formal definition of extended Action Rule is given below. We assume that:

1) $B_{St} \in A_{St}$ is maximal, such that $r_i/B_{St} = r_j/B_{St}$
2) $d(r_i) = d_i, d_{(rj)} = d_j$ and $d_i \leq d_j$
3) $(\forall a \in [A_{St} \cap L(r_i) \cap L(r_j)])[a(r_i) = a(r_j)]$
4) $(\forall m \leq q)(\forall e_m \in [A_{St} \cap (L(r_j) \backslash L(r_i))])[e_m(r_j) = u_j]$
5) $(\forall m \leq r)(\forall c_m \in [A_{Fl} \cap (L(r_j) \backslash L(r_i))])[c_m(r_j) = t_j]$
6) $(\forall m \in p)(\forall b_m \in [A_{Fl} \cap L(r_i) \cap L(r_j)])([b_m(r_i) = v_m] * [b_m(r_j) = w_m])$

Assume now that rules r_i,r_j are extracted from S and $r_i/$
$A_{St} = r_j/A_{St}$, $d(r_i) = d_i$, $d(r_j) = d_j$ and $d_i \leq d_j$. We also have the assumption that $(b_1,b_2,...,b_m)$ is list of all attributes in $Dom(r_i) \cap Dom(r_j) \cap A_{Fl}$ on which r_i and r_j differ and $r_i(b_1) = v_1, r_i(b_2) = v_2, ..., r_i(b_m) = v_m, r_j(b_1) = w_1, r_j(b_2) = w_2, ..., r_j(b_m) = w_m$.
definition By r_i,r_j Action Rule on $x \in X$, mean an expression of the form: $r = [(b_1,v_1 \rightarrow w_1) * (b_2,v_2 \rightarrow w_2) * ... * (b_m,v_m \rightarrow w_m)](x) \rightarrow (d,d_i \rightarrow d_j)(x)$.

Object $x \in X$ supports r_i,r_j extended Action Rule r in system $S = (X, A_{St} \cup A_{Fl} \cup \{d\})$ if the following conditions are satisfied:

1) $(\forall i \leq p)[b_i \in L(r)][b_i(x) = v_i] \Lambda d(x) = d_1$
2) $(\forall i \leq p)[b_i \in L(r)][b_i(y) = w_i] \Lambda d(y) = d_2$
3) $(\forall j \leq p)[a_j \in A_{St} \cap L(r_j)][a_j(x) = u_j]$ 4) $(\forall j \leq p)[a_j \in A_{St} \cap L(r_j)][a_j(y) = u_j]$ 5) Objects x,y support rules r_1,r_2 respectively.

Definition By the support of rule r, we mean the number of all objects in S satisfying the left side of the rule consisting of conditional parts of terms.
$sup(r) = card[(b_1,v_1) * (b_2,v_2) * ... * (b_m,v_m) * (d,d_i)]$
For computing the confidence of extended (r_1,r_2) Action

URL: http://dx.doi.org/10.14738/tmlai.103.12397

Rule we divide the number of objects supporting (r_1,r_2) Action Rule by the number of objects supporting left hand side of this rule and multiply it by the confidence of the classification rule r_2. Values of stable attributes listed in r_1 do not have to be considered at all.
definition The confidence of extended (r_1,r_2) Action Rule
is equal to:

$$conf((r_1,r_2)) = \frac{sup(r)}{sup(L(r))}.conf(r_2)$$

Let us consider the example extended Action Rule r in form: $r = [(a,a_1) * (c,c_1 \rightarrow c_2) \rightarrow (d,H \rightarrow A)]$ (11)

Table Xi: Knowledge Base T5

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x_1, x_2, x_7, x_8	a_1		c_1	H
x_3, x_4		b_1	c_1	A

Table Xii: Knowledge Base T6

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x_1, x_2, x_7, x_8	a_1		c_1	H
x_5, x_6		b_2	c_2	A

The support of this rule is $sup(r) = 4$, and the confidence is

$$conf(r) = \frac{4}{3}.\frac{2}{3} = \frac{2}{3} = 66\%.$$

Action Rules Based on Agglomerative Strategy - ARAS Algorithm

To construct Action Rules from pairs of classification rules is quite expensive and also, there are chances of generating their classification parts. Authors Ras and dardzinska [146] demonstrates that single classification rules are sufficient to build Action Rules.Author in [132] talks about a simple LERStype algorithm for constructing Action Rules from single classification rule. LERS were proposed by authors Grzymala et al. [122] as a classic example of bottom-up construction strategy which constructs rules with a conditional part of the length $k + 1$ after all the rules with a conditional part of length k have been constructed. System ARAS assumes that LERS can be utilized to extract classification rules.The overall complexity of the algorithm is decreased, by using LERS as the pre-processing module for ARAS. This algorithm was proposed by Ras et al in [151].

Consider the information system S in Table XV. The set $A_{St} = \{a,b,c\}$ constitute the stable attributes, and $A_{Fl} = \{e,f,g\}$ constitute the flexible attributes of system S. Using system LERS [122] to extract classification rules. The goal is to reclassify object (d,A) to either $(d,I) = d_I$ or $(d,E) = d_E$.

The following are the four certain classification rules extracted by using system LERS [122] on decision system S in Table XV.

- $r_1 = [(b_0 * c_H * f_1 * g_0) \rightarrow d_I]$
- $r_2 = [(a_2 * b_0 * e_3 * f_1) \rightarrow d_E]$
- $r_3 = [e_1 \rightarrow d_I]$

- $r_4 = [(b_0 * g_1) \to d_E]$

Action Rule schemas associated with r_1, r_2, r_3, r_4 and the reclassification task either $[d,(d_A \to d_I)]$ or $[d,(d_A \to d_E)]$ are:

- $r_{1[d_A \to d_I]} = [(b_0 * c_H * (f, \to f_1) * (g, \to g_0)] \to (d, d_A \to d_I]$
- $r_{2[d_A \to d_E]} = [(a_2 * b_0 * (e, \to e_3) * (f, \to f_1)] \to (d, d_A \to d_E]$
- $r3[dA \to dI] = (e, \to e1) \to (d, dA \to dI]$ • $r4[dA \to dE] = [(b0 * (g, \to g1)] \to (d, dA \to dE]$ We can show that:

Table Xiii: Knowledge Base T7

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x_1, x_2, x_7, x_8	a_1		c_1	H

Table Xiv: Knowledge Base T8

Set of objects	Attribute a	Attribute b	Attribute c	Decision d
x_5, x_6		b_2	c_2	A

- $Sup(r1[(dA \to dI)]) = \{x2, x3, x6\}$,
- $Sup(r_2[(d_A \to d_E)]) = \{x_2, x_6\}$,
- $Sup(r3[(dA \to dI)]) = \{x2, x3, x4, x5, x6, x7\}$,
- $Sup(r4[(dA \to dE)]) = \{x2, x3, x4, x6\}$
- Assuming that:
- $X(r_1, d_A) = Sup(r_{1[(dA \to dI)]})$,
- $X(r_2, d_A) = Sup(r_2[(d_A \to d_E)])$,
- $X(r_3, d_A) = Sup(r_{3[(dA \to dI)]})$,
- $X(r_4, d_A) = Sup(r_4[(d_A \to d_E)])$
- by applying ARAS algorithm we get:
- $(b_0 * c_H * a_1)^* = \{x_1\}$ 6⊂ $X(r_1, d_A)$,
- $(b_0 * c_H * a_2)^* = \{x_2, x_6\} \subseteq X(r_1, d_A)$,
- $(b_0 * c_H * f_2)^* = \{x_6\} \subseteq X(r_1, d_A)$,
- $(b_0 * c_H * g_1)^* = \{x_7, x_8\}$ 6⊂ $X(r_1, d_A)$,
- $(b_0 * c_H * g_2)^* = \{x_2, x_3\} \subseteq X(r_1, d_A)$

Algorithm ARAS will construct two Action Rules for the Action Rule schema:

- $[b_0 * c_H * (f, f_2 \to f_1) * (g, \to g_0)] \to (d, d_A \to d_I)$
- $[b_0 * c_H * (f, \to f_1) * (g, g_2 \to g_0)] \to (d, d_A \to d_I)$

In a similar way all the remaining Action Rules can be generated. The complexity of ARAS is lower when compared to the complexity of system DEAR. System DEAR [145] takes all possible sets of classification rules within each cluster (groups classification rules of non-conflicting rules) and tries to build Action Rules from them. ARAS algorithm treats each classification rule describing target decision value as a seed and holds other classification rules describing non-target decision values to form a cluster. Then it constructs decision rules automatically from them. Rules grabbed into a seed are only compared with that seed. So the number of sets of rules which needs to be checked is reduced greatly. The complexity of second module of ARAS is $O(k.n)$, where n is the number of classification rules extracted by LERS and k is the number of clusters. Whereas time complexity of DEAR system is $O(n.n)$, where n is number of classification rules extracted by LERS.

F. Action Rules Tightly Coupled - Object Based

According to author Dardzinska in [132] the tighlty coupled framework for Action Rules is describe as follows: In [152] by author Ras, we can find first steps taken with Action Rules mining problem without pre-existing classification rules which is resemblance with the Apriori algorithm proposed by authors Agrawal and Srikant in [137]. Authors Ras and Wieczorkowska mentioned about the changes to stable attributes in

Table Xv: Aras - Information System S

X	Attr. a	Attr. b	Attr. c	Attr. e	Attr. f	Attr. g	Decision d
x1	a1	b0	H	e1	f1	g0	I
x2	a2	b0	H	e2	f1	g2	A
x3	a3	b0	H	e3	f1	g2	A
x4	a1	b0	L	e3	f1	g0	A
x5	a1	b1	H	e2	f1	g0	A
x6	a2	b0	H	e3	f2	g0	A
x7	a2	b2	L	e3	f1	g1	A
x8	a2	b0	L	e3	f1	g1	E

their paper [138]. In general, to avoid unnecessary changes to stable attributes and to rule out Action Rules with such changes, very high cost is assigned.Later authors Ras et al. [27] and Tzacheva [153] propose there is cost associated with changes to stable or flexible attributes.

Authors Im et al. in their paper [154] propose a method that can directly extract Action Rules from the attribute values in incomplete information systems without using pre-existing conditional rules. The meaning of the previous statement is that they use pre-existing classification rules or construct rules using rule discovery algorithms such as LERS [122] or ERID [123], then generatet Action Rules either from certain pairs of rules or from a single classification rule. The methods in [136], [146], [151] do not formulate actions directly from existing classification rules. Actions are built as the effect of possible changes in classification rules. Thus, the extraction of classification rule during Action Rule formulation is inescapable.

The ARD algorithm [132] is required to construct Action Rules, as same as ERID. The main goal of algorithm ARD is to identify the relationships between granules defined by the indiscernibility relation on system's objects . Papers [135], [146], [155] introduces a new strategy for discovering Action Rules directly from the decision system. To present this method, it is sufficient to show how terms of length greater than one are built. Only positive marks yield Action Rules. Action terms of length k are built from unmarked action terms of length k-1 and unmarked atomic action terms of length one. Assume that S = (X, A ∪ {d}, V) is a decision system, and λ_1, λ_2 denote minimum support and minimum confidence respectively. Each attribute a ∈ A defines in a unique way the set $C_s(a) = \{N_s(t_a){:}t_a$ is an atomic action term built from elements in $V_a\}$.

Marking Strategy: $\forall N_s(t_a) \in C_s(a)$:

- If $L(N_s(t_a)) = 0$ or $R(N_s(t_a)) = 0$ or $L(N_s(t_a * t_d)) = 0$ or $R(N_s(t_a * t_d)) = 0$ then t_a is marked negative.
- If $L(N_s(t_a)) = R(N_s(t_a))$ then t_a stays unmarked.
- If $card(L(N_s(t_a * t_d))) < \lambda_1$ then t_a is marked negative
- If $card(L(N_s(t_a * t_d))) \geq \lambda_1$ and $conf(t_a \rightarrow t_d) < \lambda_2$ then t_a stays unmarked
- If $card(L(N_s(t_a * t_d))) \geq \lambda_1$ and $conf(t_a \rightarrow t_d) \geq \lambda_2$ then t_a is marked positive

From all marked forms, Action Rule $t_a \rightarrow t_d$ is taken into consideration.
Let us Consider the Decision System S in Table XVI. {a,c} are stable attributes denoted by A_{St} and {b,d} are flexible attributes denoted by A_{Fl}.

Table Xvi Decision System S

X	Attribute a	Attribute b	Attribute c	Decision d
x1	a1	b1	c1	H
x2	a2	b2	c3	H
x3	a2	b1	c3	A
x4	a3	b1	c2	A
x5	a2	b1	c2	A
x6	a2	b2	c2	H
x7	a3	b1	c2	A
x8	a1	b2	c1	A
x9	a1	b1	c3	H
x10	a2	b2	c3	H

We are interested in object reclassification from decision class (d, H) to (d, A). Assume that threshold for minimal support is $\lambda_1 = 2$ and for minimal confidence $\lambda_2 = 0.25$.
First Loop:
Building all atomic action terms for S. For the decision attribute {d} is S:
$N_s(t_{11}) = [\{x_1,x_2,x_6,x_9,x_{10}\},\{x_3,x_4,x_5,x_7,x_8\}]$ (12) For classification attributes, both stable and flexible, in S:
$t_2 = (a, a_2 \rightarrow a_2)$ $t_3 = (a, a_3 \rightarrow a_3)$ $t_4 = (c, c_1 \rightarrow c_1)$ $t_5 = (c, c_2 \rightarrow c_2)$ $t_6 = (c, c_3 \rightarrow c_3)$ $t_7 = (b, b_1 \rightarrow b_1)$ $t_8 = (b, b_1 \rightarrow b_2)$ $t_9 = (b, b_2 \rightarrow b_1)$ $t_{10} = (b, b_2 \rightarrow b_2)$

$$N_s(t_1) = [\{x_1,x_8,x_9\},\{x_1,x_8,x_9\}] \quad (13)$$

$$N_s(t_2) = [\{x_2,x_3,x_5,x_6,x_{10}\},\{x_2,x_3,x_5,x_6,x_{10}\}] \quad (14)$$

$$N_s(t_3) = [\{x_4,x_7\},\{x_4,x_7\}] \quad (15)$$

$$N_s(t_4) = [\{x_1,x_8\},\{x_1,x_8\}] \quad (16)$$

$$N_s(t_5) = [\{x_4,x_5,x_6,x_7\},\{x_4,x_5,x_6,x_7\}] \quad (17)$$

$$N_s(t_6) = [\{x_2,x_3,x_9,x_{10}\},\{x_2,x_3,x_9,x_{10}\}] \quad (18)$$

$$N_s(t_7) = [\{x_1,x_3,x_4,x_5,x_7,x_9\},\{x_1,x_3,x_4,x_5,x_7,x_9\}] \quad (19)$$

$$N_s(t_8) = [\{x_1,x_3,x_4,x_5,x_7,x_9\},\{x_2,x_6,x_8,x_{10}\}] \quad (20)$$

$$N_s(t_9) = [\{x_2,x_6,x_8,x_{10}\},\{x_1,x_3,x_4,x_5,x_7,x_9\}] \quad (21)$$

$$N_s(t_{10}) = [\{x_2,x_6,x_8,x_{10}\},\{x_2,x_6,x_8,x_{10}\}] \quad (22)$$

Equations (13), (14), (16), (17), (18), (19) are not marked as
$Y_1 = Y_2$.

Equation (15) is marked negative as card($Y1 \cap Y_2$) = 0. Equation (20) is not marked as sup = 2 but conf = 0.04 < λ_2.

Equation (21) is marked positive as sup = 3 and conf = 0.5. Equation (22) is not marked as sup = 3 but conf = 0.18 < λ_2.

Second Loop:
Building action terms of length two from all possible unmarked atomic action terms.

$$N_s(t_1 * t_4) = [\{x_1,x_8\},\{x_1,x_8\}] = N_s(t_4) \quad (23)$$
$$N_s(t_1 * t_5) = [\{\emptyset\},\{\emptyset\}] \quad (24)$$
$$N_s(t_1 * t_6) = [\{x_9\},\{x_9\}] \quad (25)$$
$$N_s(t_1 * t_7) = [\{x_1,x_9\},\{x_1,x_9\}] \quad (26)$$
$$N_s(t_1 * t_8) = [\{x_1,x_9\},\{x_8\}] \quad (27)$$
$$N_s(t_1 * t_{10}) = [\{x_8\},\{x_8\}] \quad (28)$$
$$N_s(t_2 * t_4) = [\{\emptyset\},\{\emptyset\}] \quad (29)$$
$$N_s(t_2 * t_5) = [\{x_5,x_6\},\{x_5,x_6\}] \quad (30)$$
$$N_s(t_2 * t_6) = [\{x_2,x_3,x_{10}\},\{x_2,x_3,x_{10}\}] \quad (31)$$
$$N_s(t_2 * t_7) = [\{x_3,x_5\},\{x_3,x_5\}] \quad (32)$$
$$N_s(t_2 * t_8) = [\{x_3,x_5\},\{x_2,x_6,x_{10}\}] \quad (33)$$
$$Ns(t2 * t10) = [\{x2,x6,x10,\{x2,x6,x10\}] \quad (34)$$
$$N_s(t_4 * t_7) = [\{x_1\},\{x_1\}] \quad (35)$$
$$N_s(t_4 * t_8) = [\{x_1\},\{x_8\}] \quad (36)$$
$$N_s(t_4 * t_{10}) = [\{x_8\},\{x_8\}] \quad (37)$$
$$N_s(t_5 * t_7) = [\{x_4,x_5,x_7\},\{x_5,x_7\}] \quad (38)$$
$$N_s(t_5 * t_8) = [\{x_4,x_5,x_7\},\{x_6\}] \quad (39)$$
$$N_s(t_5 * t_{10}) = [\{x_6\},\{x_6\}] \quad (40)$$
$$N_s(t_6 * t_7) = [\{x_3,x_9\},\{x_3,x_9\}] \quad (41)$$
$$N_s(t_6 * t_8) = [\{x_3,x_9\},\{x_2,x_{10}\}] \quad (42)$$
$$N_s(t_6 * t_{10}) = [\{x_2,x_{10}\},\{x_2,x_{10}\}] \quad (43)$$
$$N_s(t_7 * t_{10}) = [\{\emptyset\},\{\emptyset\}] \quad (44)$$
$$N_s(t_8 * t_{10}) = [\{\emptyset\},\{x_2,x_6,x_8,x_{10}\} \quad (45)$$

Equations (25), (26), (34), (35), (39), (40), (42), (43) are marked negative as card($Y_2 \cap Z_2$) = 0
Equations (24), (29), (37), (44), (45) are marked negative as

$card(Y_1) = 0$
Equations (30), (31) not marked as $Y_1 = Y_2$ Equations (32),
(33), (38) are marked as negative as $card(Y_1 \cap Z_1) = 0$
Equations (36), (41) are marked negative as sup = 1
Equation (23) is marked but no rule
Equation (27) is marked negative as sup = 2 and conf = 1 Third Loop:

Building action terms of length three, four etc. from all possible unmarked shorter terms. It is repeated until we reach the fix point. In our example the algorithm stops, as we cannot form any other action terms. Two rules we obtained are given below:

$$r_1 = [(b, b_2 \to b_1)] \to (d, H \to A) \qquad (46)$$

with $sup(r_1) = 3$ and $conf(r_1) = 0.5$

$$r_2 = [(a, a_1 * (b, b_1 \to b_2)] \to (d, H \to A) \qquad (47)$$

with $sup(r_2) = 2$ and $conf(r_2) = 1$

G. Apriori Based Association Action Rule Mining(AAR)

The Association Action Rules described by Ras et al. [27] generates Action Rules using frequent action sets in Apriori fashion. The frequent action set generation is divided in two steps: merging step and 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 III, the primary action sets generated by AAR are shown in table XVII. The frequent action sets generated by AAR are shown in table XVIII.

Table Xvii: 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), $(E, E_1 \to E_2)$, $(E, E_1 \to E_3)$, $(E, E_2 \to E_1)$, $(E, E_2 \to E_3)$, $(E, E_3 \to E_1)$, $(E, E_3 \to E_2)$
F	(F, F_2), (F, F_3), $(F, F_2 \to F_1)$, $(F, F_2 \to F_3)$, $(F, F_3 \to F_1)$, $(F, F_3 \to F_2)$
G	(G, G_1), (G, G_2), (G, G_3), $(G, G_1 \to G_2)$, $(G, G_1 \to G_3)$, $(G, G_2 \to G_1)$, $(G, G_2 \to G_3)$, $(G, G_3 \to G_1)$, $(G, G_3 \to G_2)$
D	(D, D_1), (D, D_2), (D, D_3), $(D, D_1 \to D_2)$, $(D, D_1 \to D_3)$, $(D, D_2 \to D_1)$, $(D, D_2 \to D_3)$, $(D, D_3 \to D_1)$, $(D, D_3 \to D_2)$

Table Xviii: 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 (D,2 → 1) is not present in it. From each frequent action set, the association Action Rules are formed. 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 III, the algorithm generates Association Action Rules, an example is shown below:

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

H. Association Action Rules

Association rules discovery was first introduced by Agrawal et al. [156].Association rule mining is used to find certain association relationships among a set of objects in large databases [132]. The association relationships are described as rules. A simple example of association rule given by authors Agrawal et al. in [156], is as follows: *"90% of transactions that purchase bread and butter also purchase milk"*. The antecedent of this rule consists of bread and butter and the consequent consists of milk alone. The number 90% is the confidence factor of the rule.

Data mining literature [132] contains lot of papers related to design scalable algorithms for mining association rules. For instance, in paper [137]authors agrawal et al. presents a more formal definition of association rule. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subset I$. Each transaction has a unique identifier called transaction ID. A transaction T contains X, where X is set of some items in the item set I, if $X \subset I$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$ and $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. The rule $X \rightarrow Y$ has a support s in the transaction set D if s% of transactions in D contain $X \cup Y$. New algorithms were proposed by them for discovering association rules between items in a large sales transaction database. Similarly authors Gouda et al. [157], Han et al. [158], Savasere et al. [159], and Zaki et al. [160] propose specific scalable algorithms for huge amount of data.

There are two categories of association rules according to authors Paul et al. [161]as follows: conventional positive association rules and irregular association rules. The patterns that occur frequently and represent routine decisions based on set of facts are called conventional positive association rules. On the other hand irregular association rules is known as when the patterns that represent rare decisions based on the same set of facts . They propose a level wise search algorithm based on action and non-action type data to find irregular association rules. These irregular patterns help identify the wrong or illegal practice or decision in applications including banking,medical,health-care, and others to improve the system. The proposed algorithm uses maximum confidence constraint to form rules.

Authors Wang et al. in paper [162], proposed to build a new model for promotion strategies to new customers, with the goal of maximizing the profit for a given set of transactions and pre-selected target items. They use association rule to generate recommendation action. Authors Padmanabhan et al. [163] and Wang et al. [164] suggested methods to generate interesting patterns by incorporating knowledge in the process of searching for patterns in data. They focus on providing methods to generate unexpected patterns with respect to intuition.

Imbalanced datasets are likely to prune most of the rules from the minority class and affect the classification accuracy [165]. Authors Zhang et al. [166] introduces an efficient algorithm to mine novel association rules known as combined association rules on imbalanced datasets. Formal definition of combined association rule is given below.

definition: Let T be a dataset where each tuple is described by a schema $S = (S_{D1},...,S_{Dm},S_{A1},...,S_{An},S_C)$, in which $S_D = (S_{D1},S_{D2},...,S_{Dm})$ are m non-actionable attributes, $S_A = (S_{A1},S_{A2},...,S_{An})$ are n actionable attributes, and S_C is a class attribute. Suppose itemset $D \subseteq I_D$, I_D is the itemset of any items with attributes $(S_{D1},S_{D2},...,S_{Dm})$, itemset $A \subseteq I_A$, I_A is the itemset of any items with attributes $(S_{A1},S_{A2},...,S_{An})$, C is 1-itemset of class attribute, a combined association rule is represented as $D + A_1 \to C_{k1} ... D + A_i \to C_{ki}$, here + means itemsets appearing simultaneously.

Authors Ras et al. [27] introduce the notion of action frequent action sets, and present methods of building Action Rules from frequent action sets to achieve the following objectives: to extract Action Rules directly from a decision system without using pre-existing classification rules, and to extract Action Rules that have minimal attribute involvement.

1) Frequent Action Sets

Let t_a is an atomic action set, where $Ns(t_a) = [Y_1, Y_2]$ and a ∈ A. We say that t_a is called frequent if $card(Y_1) \geq \lambda_1$ and $card(Y_2) \geq \lambda_2$. The operation of generating (k + 1) element candidate action sets from frequent k-element action sets is performed in two steps:
Merging Step: Merge pairs (t_1,t_2) of frequent k-element action sets into $(k + 1)$ element candidate action set if all elements in t_1 and t_2 are the same except the last element.

Pruning Step: Delete each $(k+1)$ element candidate action set t if either of it is not an action set or some k-element subset of f is not a frequent k-element action set.

Now, if t is a $(k + 1)$ element candidate action set, $N_s(t) = [Y_1,Y_2]$, $card(Y_1) \geq \lambda_1$ and $card(Y_2) \geq \lambda_2$, then t is a frequent $(k + 1)$ element action set. We say that t is a frequent action set in S if t is a frequent k-element action set in S, for some k.

Assume now that the expression $[t - t_2]$ denotes the action set containing all atomic action sets listed in t but not listed in t_1. The set AARS(λ_1, λ_2) of association Action Rules in S is constructed in the following way:

Let t be a frequent action set in S and t_1 is its subset. Any Action Rule $r = [(t-t_1) \rightarrow t_1]$ is an association Action Rule in AARS(λ_1, λ_2) if conf(r) ≥ λ_2.

Consider the information system in Table XVI. Assume $\lambda_1 = 2$ and $\lambda_2 = 0.4$. The following frequent action sets can be constructed.

Building pairs from all possible atomic terms: as shown in Table XX

Association Action Rules can be constructed from all frequent action sets. For instance, we can generate Action Rule [(a, a_2)* (b, $b_2 \rightarrow b_1$) * (c, c_2)] → (d, H → A) from the last frequent action set listed above. We can also construct simple association Action Rule, calculate the cost of association Action Rule, and give a strategy to construct simple association Action Rules of lowest cost.

Table Xix: Frequent Action Sets

Frequent Action Set	Support
(a, a_1)	3
(a, a_2)	5
(a, a_3)	2
(b, b_1)	6
(b, b_2)	4
(b, $b_1 \rightarrow b_2$)	6
(b, $b_2 \rightarrow b_1$)	4
(c, c_1)	2
(c, c_2)	4
(c, c_3)	4
(d, H)	5
(d, A)	5

Table Xx: Possible Action Sets

Possible Action Set	Support
$(a, a_1) * (b, b_1)$	2
$(a, a_1) * (b, b_2)$	1 (not frequent)
$(a, a_1) * (b, b_1 \rightarrow b_2)$	2
$(a, a_1) * (b, b_2 \rightarrow b_1)$	1 (not frequent)
$(a, a_1) * (c, c_1)$	2
$(a, a_1) * (c, c_2)$	0 (not frequent)
$(a, a_1) * (c, c_3)$	1 (not frequent)
$(a, a_1) * (d, H)$	2
$(a, a_1) * (d, A)$	1 (not frequent)
$(a, a_2) * (b, b_1)$	2
$(a, a_2) * (b, b_2)$	3
$(a, a_2) * (b, b_1 \rightarrow b_2)$	2
$(a, a_2) * (b, b_2 \rightarrow b_1)$	3
$(a, a_2) * (c, c_1)$	0 (not frequent)
$(a, a_2) * (c, c_2)$	2
$(a, a_2) * (c, c_3)$	3
$(a, a_2) * (d, H)$	3
$(a, a_2) * (d, A)$	2
...	...
$(b, b_1) * (c, c_1)$	1 (not frequent)
$(b, b_1) * (c, c_2)$	3
$(b, b_1) * (c, c_3)$	2
...	...
$(a, a_2) * (b, b_2 \rightarrow b_1) * (c, c_2) * (d, H \rightarrow A)$	2

I. Representative Association Action Rules

Author Dardzinska, [132], encapsulated the concept of representative association rules as follows: Author Kryszkiewicz [167] propose the concept of representative association rules. These representative association rules form a small subset of association rules from which the remaining association rules can be generated. Authors Ras et al. [27] and Authors Saquer and Jitender [168] confer similar approach for Action Rules. definition By a cover of association rule $r : (t_1 \rightarrow t)$ we mean $cov(r) = cov(t_1 \rightarrow t) = \{t_1 * t_2 \rightarrow t_3 : t_2, t_3$ are not overlapping subterms of $t\}$.

For instance, assume that $r : [(a, a_1 \rightarrow a_2) \rightarrow (b, b_1 \rightarrow b_2) * (c, c_1 \rightarrow c_2) * (d, d_1 \rightarrow d_2)]$ is an association rule. Then, $((a, a_1 \rightarrow a_2) * (b, b_1 \rightarrow b_2) * (c, c_1 \rightarrow c_2)) \in cov(r)$.

Property 1. If $r \in AAR_s(\lambda_1, \lambda_2)$, then each rule $r_k \in cov(r)$ also belongs to $AAR_s(\lambda_1, \lambda_2)$.

Proof: From the definition of $AAR_s(\lambda_1, \lambda_2)$ we have: $\sup(r) \geq \lambda_1$ and $\mathrm{conf}(r) \geq \lambda_2$.

$$r_k = (t_1 * t_2) \rightarrow t_4 \qquad (48)$$

URL: http://dx.doi.org/10.14738/tmlai.103.12397

$r = (t_1 * t_2) * t_3 * t_4$ (49)

$N_s(t_i) = [Y_i, Z_i]$ (50)

Where i ∈ {1, 2, 3, 4}

Consider equations (48), (49), and (50).

Since, $\frac{card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]}{card[Y_1]} \geq \lambda_1$ then, $\frac{card[Y_1 \cap Y_2 \cap Y_4]}{card[Y_1 \cap Y_2]} \geq \lambda_1$. It comes from the fact, that:

$card[Y_1 \cap Y_2 \cap Y_4] \geq card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]$ and $card[Y_1 \geq card[Y_1 \cap Y_2]$.

In a similar way we show that $\frac{card[Z_1 \cap Z_2 \cap Z_4]}{card[Z_1 \cap Z_2]} \geq \lambda_1$.

The same, $sup(r_k) \geq \lambda_1$.

Now assume, that: $conf(r) = \frac{card[Y_1 \cap Y_2 \cap Y_3 \cap Y_4]}{card[Y_1]} . \frac{card[Z_1 \cap Z_2 \cap Z_3 \cap Z_4]}{card[Z_1]} \geq \lambda_2$.

Clearly, $\frac{card[Y_1 \cap Y_2 \cap Y_4]}{card[Y_1 \cap Y_2]} . \frac{card[Z_1 \cap Z_2 \cap Z_4]}{card[Z_1 \cap Z_2]} \geq \lambda_2$.

The same $conf(r_k) \geq \lambda_2$.

Property 2. Representative association rules $RAAR_s(\lambda_1,\lambda_2)$ form a least set of representative association Action Rules that covers all association Action Rules
$AAR_s(\lambda_1,\lambda_2)$.

Proof: Assume that $r \in RAAR_s(\lambda_1,\lambda_2)$ and there exists $(r_k \in (t_1 \rightarrow t) \in AAR_s(\lambda_1,\lambda_2)$ such that r_k 6= r and $r \in cov(r_k)$. Since $r \in cov(r_1)$ then r is not in $RRAR_s(\lambda_1,\lambda_2)$.

Property 3. All association rules $AAR_s(\lambda_1,\lambda_2)$ can be derived from representative association Action Rules $RAAS_s(\lambda_1,\lambda_2)$ by means of cover operator.

Proof:

$r : (t \rightarrow s) \in AAR_s(\lambda_1,\lambda_2)$ (51)

$t = t_1 * t_2 * ... * t_n$ (52)

Where $\{ti\}_i \in \{1,2,...,n\}$

Assume that equations (51) and (52) are atomic action sets. It means that $sup(r) \geq \lambda_1$ and $conf(r) \geq \lambda_2$.

$r_i(t) = ((t - t_i \rightarrow s * t_i)$ (53)

Let equation (53) for any atomic action set t_i in t. Clearly support and confidence are given by equations (54) and (55). $sup(r_i(t)) = sup(r)$ (54) $conf(r_i(t)) \leq conf(r)$ (55)

Now we show how to construct representative association Action Rule from which r can be generated. It consists of two main steps: First we:

1. Find t_i in t such that $conf(r_i(t)) \geq \lambda_2$
2. If succeeded then $t := (t - t_i), s := s * t_i$ and we go back to step 1. Otherwise procedure stops.

In next step we extend the decision part of the rule generated in previous step. Assuming that $(t \to s)$ in such rule, and $T = \{t_1,t_2,...,t_m\}$ is a set of all atomic action terms not listed in s, we:

1. find t_i in T such that $sup(t \to s * t_i) \geq \lambda_1$
2. if succeeded then $T := T - t_i, s := s * t_i$ and we go back to step 1. Otherwise procedure stops.

The resulting association Action Rule is a representative rule from which the initial rule r can be generated.

J. Cost and Feasibility

Author Ras and Tzacheva [169], and author Tzacheva and Tsay [170], propose the notion cost and feasibility of an Action Rule. The notion of cost and feasibility is described as follows in [132]: Assume that S is an information system. Let $b \in B$ is flexible attribute and b_1,b_2 are values of b. By $\rho\ (b_1,b_2)$ mean any number from the open interval $(0,1) \cup \{+\infty\}$ which describes the cost to change the value from b_1 to b_2 by the user of the information system S.

- The value of $\rho(b_1,b_2 \approx 0)$ is interpreted that the change of values from b_1 to b_2 is quite trivial.
- The value of $\rho(b_1,b_2 \approx 1)$ is interpreted that the change of values from b_1 to b_2 is very difficult to be achieved.
- The value of $\rho(b_1,b_2 \approx +\infty)$ is interpreted that the change is not feasible.

Also, if $\rho(b_1,b_2) < \rho(b_3,b_4)$, then change of values from b_1 to b_2 is more feasible than the change from b_3 to b_4.

The values $\rho(b_i,b_j)$ are given by the user of information system and they should be seen as atomic values needed to introduce the notion of the feasibility of an Action Rule.

Assume now that equation (56) is a (r_1,r_2) Action Rule
[135] [171].
$r = [(b_1,v_1 \to w_1) * (b_2,v_2 \to w_2) * ... *$
(56)
$(b_m,v_m \to w_m)](x) = (d,d_1 \to d_2)(x)$ definition By the cost of rule r denoted by $cost(r)$ means value in equation (57)

$$cost(r) = {}^{X}\{\rho(v_i,w_i) : 1 \leq i \leq n\}. \qquad (57)$$

Rule r is feasible if $cost(r) < \rho(d_1,d_2)$, which means that $cost(r)$ has to be a finite number and the cost of the conditional part of the rule has to be a finite number and the cost of the conditional part of the rule has to be lower than the cost of the decision part of the rule.

Considering d to be a decision attribute, assume that $D_s[(d,d_1 \to d_2)]$ denotes the set of all Action Rules in S having the term $(d,d_1 \to d_2)$ on the decision site. Among all Action Rules in $D_s[(d,d_1 \to d_2)]$ we have to choose a rule with the smallest cost value. However it can still happen that the rule we chose has the cost value not acceptable by the user of the information system S. The cost of the Action Rule in equation (58) might be high only because the cost value of one of its sub-terms in the conditional part of the rule is high.
$r_i = [(b_1,v_1 \to w_1) * (b_2,v_2 \to w_2) * ... *$
(58)
$(b_m,v_m \to w_m)](x_i) \to (d,d_1 \to d_2)(x_i)$

Authors Tzacheva and Ras [169] introduce a heuristic procedure to find the low-cost Action Rule. Assume that $(b_i,v_i \rightarrow w_i)$ is the sub-term that increases the cost of Action Rule in (58). In this scenario, we may look for an Action Rule in $D_s[(b_i,v_i \rightarrow w_i)]$ with the smallest cost value.

Assume that equation (59) is feasible rule.

$$r_j = [(b_{i1}, v_{i1} \rightarrow w_{i1}) * (b_{i2}, v_{i2} \rightarrow w_{i2}) \\ *\ldots * (b_{im}, v_{im} \rightarrow w_{im})](x_j) \rightarrow (b_i, v_i \rightarrow w_i)(x_j) \quad (59)$$

Since objects x_i,x_j are coming from the same information system S, we can compose r_i with r_j getting a new feasible rule given below:

$$(ri,rj) = [(b1,v1 \rightarrow w1) * (bi1,vi1 \rightarrow wi1) * (bi2,vi2 \rightarrow wi2) \\ *\ldots * (b_{in},v_{in} \rightarrow w_{in}) * \ldots * \\ (b_m,v_m \rightarrow w_m)](x) \rightarrow (d,d_1 \rightarrow d_2)(x) \quad (60)$$

The cost of this new Action Rule (r_i,r_j) is lower than the cost of (r_i). However, if support of this rule is equal to 0, then it has no value for user. Otherwise, we can recursively follow this method looking for cheaper rules reclassifying objects from the group d_1 into the group d_2. Each successful step will produce a rule which is cheaper than the previous one. Obviously, this heuristic procedure has to end.

It may seem that if $D_s[(d,d_1 \rightarrow d_2)$ contains all Action Rules which reclassify objects from one group d_1 into the group d_2 then any new Action Rule obtained as the result of the proposed recursive strategy is already in that set. This statement is agreeable but practically $D_s[(d,d_1 \rightarrow d_2)$ never contains all such rules.

Firstly, to generate all possible rules from an information system is expensive and secondly even if we have such rules it is still too expensive to generate all possible Action Rules from them. Thus the author justifies the applicability of the heuristic.

K. Event Condition Action Rule

Classical database systems does not initiate operations on their own, instead they are passive and respond to user queries. Later due to significant increase in the amount of data, managing the huge volume of complex data was a bit complicated . Thus the passive databases were converted to active databases to respond independently to data-related events. Authors Goldin et al. [172] describes this behavior as eventcondition-action Rules (ECA). There are three components in ECA rules as follows: event, condition and action. The event is the happening to which the rule responds, condition examines the context of the event, if the relevant event and condition occurred then action denotes the task to be carried out [173].

Authors Qiao et al. [174] present graphical ECA rules with temporal events to specify real time constraints. They used the data from smart home application to validate the generated rules. In smart homes, sensors collect data about human movement and interaction to appliances which is then sent to a realtime active database. Researchers believe that with ECA rules, real-time active database will have enhanced capabilities to detect complicated events and contexts to differentiate between situations, thus anticipating potential hazardous situations and intelligently advise safety and living standards for person inside the monitored smart home.

L. Meta Action

Authors Touati et al. [175] and Tzacheva et al. [176]referred Meta-Actions as higher level concept.Let us Consider the medical example from section X-D, in order to move a patient from worst prognoses state to betetr prognoses state requires some treatment procedures to be changed or some medication to be changed. This actionable knowledge is represented by meta-actions. A more formal definition of meta-action given by authors Touati et al. in [175] is as follows:

definition (Meta-actions). Meta-actions associated with an information system *S* are defined as higher level concepts used to model certain generalization of Action Rules. Meta-actions, when executed, trigger changes in values of some flexible attributes in *S*.

Authors Tzacheva and Ras [176]in their paper present a strategy for constructing association Action Rules and action paths by introducing the use of Meta-actions and influence matrix [177]. According to [176] some higher-level actions called Meta-actions are required to trigger the alteration of flexible attributes in order to move undesirable objects into a desirable set of group. For example, if a patient is suffering from certain disease, then without proper treatment or drug it is never possible to get rid of the disease. The treatment or medicine in this is an example of Meta-action. They use Influence matrix to identify which candidate association Action Rule and action paths are valid with respect to meta-actions and hidden correlation between classification attributes and decision attributes. Action paths are a sequence of action terms as mentioned in section earlier in Action Rules Assumptions.

M. Meta Actions for Sentiment Analysis and Business Recommendations

As described in section X-L Meta Actions are higher level actions that can activate Action Rules. According to authors Ras et al. [178] Action Rules are a set of atomic actions for achieving expected result, Meta-actions are the actions that need to be executed in order to trigger corresponding atomic actions.

Authors Ras et al. [178] model a Net Promoter Score(NPS) Recommender system for driving business revenue mainly based on Action Rules and Meta Actions. Net Promoter Score (NPS) is a standard metric for measuring customer satisfaction. This system utilized around 400,000 records of the customer satisfaction telephone surveys containing details related to customer details, survey details and benchmark questions. Action Rules, knowledge in actionable format is collected from customers using a business and also from customers using semantically similar business. The concept of decision reducts (minimal set of attributes that keep the characteristics of the full dataset [178]) is used to choose critical benchmarks. The triggers (Meta Actions) for Action Rules are extracted based on aspect-based sentiment analysis [179] and text summarization of the customer text comments in the survey. Feature-opinion pairs are identified with Stanford Parser. They also performed feature clustering based on pre-defined list of seed words. Meta Actions are generated by dividing feature class into several subclasses. Authors Tarnowska et al. [180] also talks about the application of decision reducts theory to solve business problem. Similar to authors Ras et al. [178], his paper focuses on business recommendations to improve Net Promoter Score of companies. They detail the application area - Customer Loyalty Improvement, machine learning techniques used to develop the knowledge based system and visualization techniques for the interactive recommender system.

N. Meta Actions for Sentiment Analysis and a Sub-component Technology Recommendations

As described in section X-LMeta Actions also help in technology field or social media like twitter.In paper [181]author Kharde and Sonawane explains about sentiment predictor system that can be helpful in recommender systems.According to the authors by using some machine learning algorithms the recommender system will not recommend items that receive a lot of negative feedback or fewer ratings.In online communication, we also come across abusive language and other negative comments that can be detected easily by identifying a highly negative sentiment and correspondingly taking action against it.

ACTION RULES ON BIG DATA

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

Authors Bagavathi et al. [182] 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 [27] - an iterative method to extract all possible action rules. In order to overcome the computational complexity and expense, authors Bagavathi et al. [182] propose vertical environment by using vertical data split [182], where only subsets of the attributes are taken for scalability purpose. However, since this method is iterative it takes longer time to process huge datasets.

Hybrid Action Rule mining approach [183] 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 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.10 The algorithm runs separately on each partition, does transformations like map(), flatmap() functions and combine results with join() and groupBy() operations.

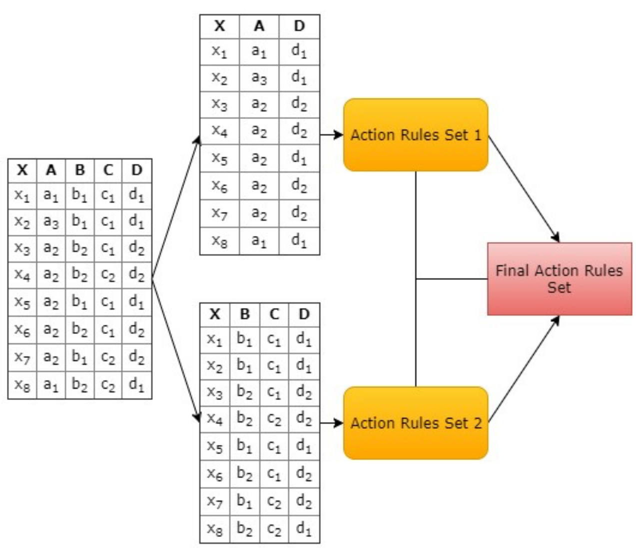

Fig. 10.Example Data Distribution using Vertical Data Distribution.

B. Hybrid Action Rule Mining

The Rule-Based method using LERS [122] has the disadvantage of computing preexisting decision rules in order to generate the Action Rule. For that it requires complete set of attributes which makes it difficult to implement it in a distributed cloud environment.

The Object-Based method can be implemented in distributed cloud environment by using vertical data split [182], where only subsets of the attributes are taken for scalability purpose. However, since this method is iterative it takes longer time to process huge datasets.

Hybrid Action Rule mining approach [183] 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 Association Action Rule approach. The pseudocode of the algorithm is given in Fig.11.

The Hybrid Action Rule Mining Algorithm works with the Information System as follows. The information system in table III has the following attributes: flexible Pf l , stable Pst and decision d, P = (Pst, Pf l, {d}). From table III Pst ={A, B, C}, Pf l = {E, F, G}, and d = D.

The following example re-classifies the decision attribute D from d2 → d1. The algorithm Fig. 11. initially uses the LERS method to extract the classification rules that are certain and then generates Action Rule schema as given in the following equations " (61)" ," (62)".

[B1 ∧C1 ∧(F,→ F1)∧(G,→ G1)] → (D,D2 → D1). (61)

[(E,→ E1)] → (D,D2 → D1). (62)

The algorithm then creates sub-table for each Action Schema. For example " (61)", generates the following subtable shown in table XXI.

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 "(63)" based on the sub-table shown in table XXI.

Table Xxi

X	B	C	F	G	D
$x1$	B_1	C_1	$F2$	G_1	D_1
$x3$	B_1	C_1	$F2$	G_3	D_2
$x6$	B_1	C_1	$F3$	G_1	D_2
$x8$	B_1	C_1	$F2$	G_3	D_2

Association Action Rule approach. [B1 ∧ C1 ∧ (F,→ F1) ∧ (G,G3 → G1)] → (D,D2 → D1). (63)

```
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. 11.Hybrid Action Rule Mining Algorithm.

This Hybrid Action Rule algorithm is implemented in Spark [184] and runs separately on each sub-table and performs transformations like map(), flatmap(), join(). The methodology of this algorithm is shown in Fig. 12.

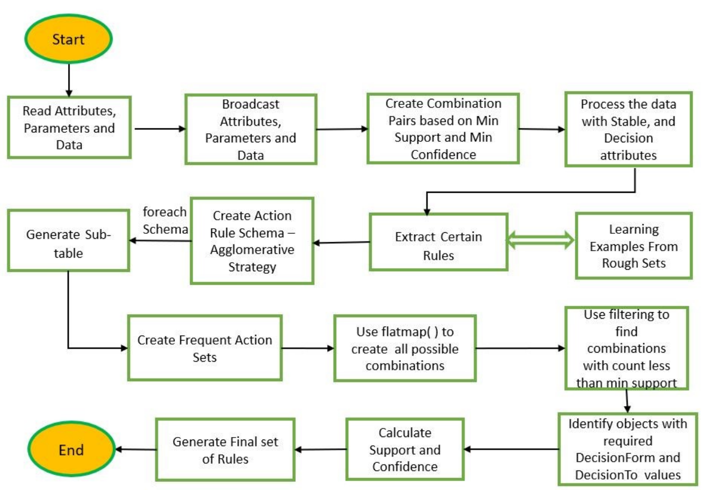

Fig. 12. Hybrid Action Rule Mining Algorithm - Flowchart

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. 13 and the proposed methodology is depicted in the Fig. 14.

```
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. 13. Hybrid Action Rule Mining with Threshold Algorithm

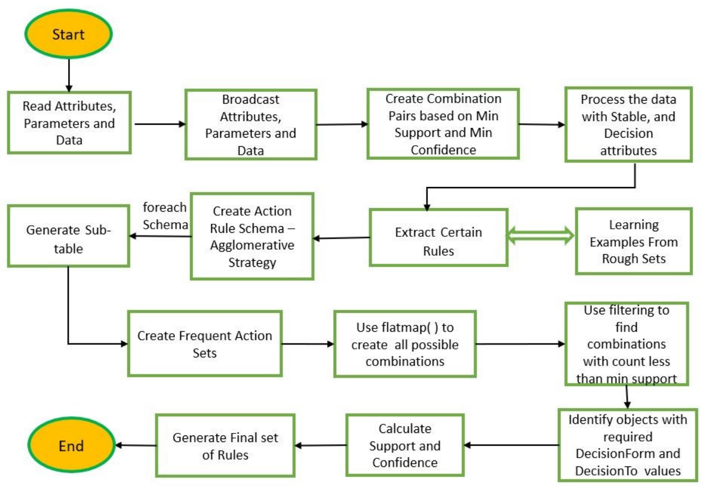

Fig. 14. Hybrid Action Rule Algorithm Mining with Table size threshold θ – Flowchart

ACTIONABLE RECOMMENDATIONS for EMOTION MINING

Actionable patterns in related to Emotion mining, is able to suggest a way to alter the user's negative Emotion or user's neutral Emotion to a more positive Emotion, or a desirable state/attitude. For example, for customer care services, recommendation systems for online shopping, or smart phones that are able to recognize human emotions, Emotion altering Actionable Patterns include: suggesting calming music, playing mood enhancing movie, changing the background colors to suiting ones, or calling caring friends (for smart phones).

In the paper of Ranganathan et al. [185] the primary intent of the Action Rules generated is to provide viable suggestions on how to make a twitter user feel more positive.For Twitter social network data, Actionable Recommendations may include - how to increase user's friends count, how to increase the user's follower's count, and how to change the overall sentiment from negative to positive, or from neutral to positive.

CONCLUSION

Action Rules mining is useful for discovering actionable patterns in datasets from several domains such as: medical, financial, industrial, educational, and social media. Actionable patterns extracted from the data are of critical importance for solving the problems in these domains. Currently, Action Rules applications in medical [186] and business [107] and social media [20] fields are of great interest and significance, because of the innumerable growth of the data size every day. There is active research involve during past decade to extract Action Rules on datasets in these fields. Multiple methods both in rule-based and object-based approaches have been proposed to extract Action Rules efficiently. Nowadays, in the epoch of Big Data, where resources like social media and IoT are becoming dominant and fast in providing data, the growth of immense amount data is inevitable. Running even the most efficient Action Rules algorithms on a single machine for such a huge data is a tedious task and consumes unreasonable time.

Cloud Computing is becoming crucial for the better performance of many machine learning algorithms to handle rapid generation of big data in distributed and parallel fashion. Action Rules extraction using distributed computing frameworks [30] [31] [38] and publicly available Cloud platforms [187]is a beneficial upgrade to the current Action Rule mining algorithms. Providing a scalable algorithm design for Action Rule mining in distributed environment would allow many applications to benefit from extracting Action Rules in a time efficient manner with large volumes of data.

Important application of actionable pattern mining is emotion detection in text. 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 benefits Social Media and Education domains[188]. We review mining of Student Survey Data, including the Modified Hybrid Action Rule Mining Algorithm, which suggests 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.

Emotions and feelings accompany us throughout the span of our lives and color the way we build and maintain the basis for interactions with people in a society [65]. Additional applications include: customer care services, recommendation systems for online shopping, text messages, E-Learning, and student teaching evaluations, as well as the smart phones and technology of the future, which is able to detect and recognize human emotions. Mining for Actionable knowledge and providing Actionable Recommendations, which can alter emotions from negative to positive is a challenging and important subject, that benefits all emotion recognition systems.

FUTURE WORK

The Modified hybrid action rule mining methods improves the processing time with Big Data. 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. We plan to test this method with medical data HCUP [28].

In the future, we plan to apply Action Rules in order to suggest ways to alter specific negative emotions such as: sadness, fear, anger to more positive ones such as: joy, trust, surprise. We plan to apply this method to Twitter social network data, as well E-Learning, Student-Teacher Evaluations, and Amazon Customer Satisfaction Surveys.

References

[1] A. Kuvvarapu, A. Kunuku, and G. Saggurthi, Data Mining on CloudBased Big Data, 04 2020, pp. 51–59.

[2] H. B. Barua, "Data science and machine learning in the clouds: A perspective for the future," 2021.

[3] Z. Salih Ageed, S. R. M. Zeebaree, M. Mohammed Sadeeq, S. Fattah Kak, H. Saeed Yahia, M. R. Mahmood, and I. Mahmood Ibrahim, "Comprehensive survey of big data mining approaches in cloud systems," Qubahan Academic Journal, vol. 1, no. 2, p. 29–38, Apr. 2021. [Online]. Available: https://journal.qubahan.com/index.php/qaj/article/view/46

[4] B. A. Ojokoh, O. W. Samuel, O. M. Omisore, O. A. Sarumi, P. A. Idowu, E. R. Chimusa, A. Darwish, A. F. Adekoya, and F. A. Katsriku, "Big data, analytics and artificial intelligence for sustainability," Scientific African, vol. 9, p. e00551, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468227620302891

[5] J. Abasova, P. Tanuska, and S. Rydzi, "Big data—knowledge discovery in production industry data storages—implementation of best practices," Applied Sciences, vol. 11, no. 16, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7648

[6] I. Sarker, "Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective," 04 2021.

[7] V. Sebestyen, T. Czvetk ´ o, and J. Abonyi, "The applicability ´of big data in climate change research: The importance of system of systems thinking," Frontiers in Environmental Science, vol. 9, p. 70, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fenvs.2021.619092

[8] W. Shi, H. Fu, P. Wang, C. Chen, and J. Xiong, "Climatechange vs. global warming: Characterizing two competing climate discourses on twitter with semantic network and temporal analyses," Int. J. Environ. Res. Public Health, vol. 17, no. 3, p. 1062, feb 2020.

[9] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from twitter text," Journal of Computational Science, vol. 36, p. 101003, 2019.

[10] S. M. N. C. K. Manjunath B Talawar, Prasad.M.R, "Analyzing social media data in educational sectors using data mining techniques," International Journal of Computer Science and Information Technologies, vol. 7, no. 3, pp. 2125–2129, 2016.

[11] N. Matrazali, N. Malizan, N. Hasbullah, M. Wook, N. Zainuddin, K. Ishak, S. Ramli, and S. Sukardi, "Opinion mining for national security: techniques, domain applications, challenges and research opportunities," Journal of Big Data, vol. 8, 12 2021.

[12] X. Luo and Y. Yi, "Topic-specific emotion mining model for online comments," Future Internet, vol. 11, no. 3, 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/3/79

[13] E. A. Tzacheva, A., "Modified hybrid scalable action rule mining for emotion detection in student survey data," January 21-22, 2022.

[14] Z. Ahmad, R. Jindal, A. Ekbal, and P. Bhattachharyya, "Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding," Expert Systems with Applications, vol. 139, p. 112851, 2020.

[15] N. Nandal, R. Tanwar, and J. Pruthi, "Machine learning based aspect level sentiment analysis for amazon products," Spatial Information Research, vol. 28, no. 5, pp. 601–607, 2020.

[16] P. A. A. Tzacheva, A.Easwaran, "Detecting and improving student emotions using actionable pattern discovery in student survey data," Advances in Engineering : an International Journal, vol. 3, no. 2, 2021.

[17] H.-J. Jang, J. Sim, Y. Lee, and O. Kwon, "Deep sentiment analysis: Mining the causality between personality-value-attitude for analyzing business ads in social media," Expert Systems with applications, vol. 40, no. 18, pp. 7492–7503, 2013.

[18] A. Seyeditabari, N. Tabari, S. Gholizadeh, and W. Zadrozny, "Emotion detection in text: Focusing on latent representation," 2019.

[19] A. Seyeditabari, N. Tabari, S. Gholizade, and W. Zadrozny, "Emotional embeddings: Refining word embeddings to capture emotional content of words," 2019.

[20] K. Garcia and L. Berton, "Topic detection and sentiment analysis in twitter content related to covid-19 from brazil and the usa," Applied Soft Computing, vol. 101, p. 107057, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620309959

[21] M. Singh, A. K. Jakhar, and S. Pandey, "Sentiment analysis on the impact of coronavirus in social life using the bert model," Social Network Analysis and Mining, vol. 11, 2021.

[22] S. K. Narayanasamy, K. Srinivasan, S. Mian Qaisar, and C.-Y. Chang, "Ontology-enabled emotional sentiment analysis on covid-19 pandemic-related twitter streams," Frontiers in Public Health, vol. 9, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpubh.2021.798905

[23] A. A. Tzacheva, A. Bagavathi, and A. K. Datta, "In search of actionable patterns of lowest cost - a scalable graph method," Int. J. Database Manag. Syst., vol. 10, no. 3, pp. 01–19, jun 2018.

[24] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," Social Network Analysis and Mining, vol. 11, no. 1, p. 81, Aug 2021. [Online]. Available: https://doi.org/10.1007/s13278-021-00776-6

[25] A. Bagavathi, A. Tripathi, A. A. Tzacheva, and Z. W. Ras, "Actionable pattern mining-a scalable data distribution method based on information granules," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2018, pp. 32–39.

[26] R. V. T. A. Bagavathi, A., "Data distribution method for scalable actionable pattern mining," October 1-3 2018.

[27] 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.

[28] A. Bagavathi and A. Tzacheva, "Scalable action mining for recommendations to reduce hospital readmission," 07 2019, pp. 159–166.

[29] C. D. S. A. M. K. R. E. T. G. J. L. H. S. S. S. B. S. C. C. O. O. S. R. B. R. V. Vavilapalli, A. Murthy and E. Baldeschwieler, "Apache Hadoop yarn: yet another resource negotiator," Proceedings of the 4th annual Symposium on Cloud Computing (SOCC '13), 2013.

[30] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, p. 107–113, jan 2008. [Online]. Available: https://doi.org/10.1145/1327452.1327492

[31] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," ser. NSDI'12. USA: USENIX Association, 2012, p. 2.

[32] "Hadoop apache project," vol. 53, p. 2, 2008.

[33] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "Mllib: Machine learning in Apache spark," 2015.

[34] G. Nishant, "Apache kafka," p. 88, 2013.

[35] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig Latin: a not-so-foreign language for data processing," in Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 1099–1110.

[36] Y. Huai, A. Chauhan, A. Gates, G. Hagleitner, E. N. Hanson, O. O'Malley, J. Pandey, Y. Yuan, R. Lee, and X. Zhang, "Major technical advancements in apache hive," in Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1235–1246. [Online]. Available: https://doi.org/10.1145/2588555.2595630

[37] M. N. Vora, "Hadoop-hbase for large-scale data," Proceedings of 2011 International Conference on Computer Science and Network Technology, vol. 1, pp. 601–605, 2011.

[38] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink™: Stream and batch processing in a single engine," IEEE Data Eng. Bull., vol. 38, pp. 28–38, 2015.

[39] A. Jain and A. Nalya, "Learning storm," 2014.

[40] G. D. F. Morales and A. Bifet, "Samoa: Scalable advanced massive online analysis," Journal of Machine Learning Research, vol. 16, no. 5, pp. 149–153, 2015. [Online]. Available: http://jmlr.org/papers/v16/morales15a.html

[41] "Apache projects. [online]."

[42] R. Bekkerman, M. Bilenko, and J. Langford, "Scaling up machine learning: parallel and distributed approaches," in KDD '11 Tutorials, 2011.

[43] R. Anil, S. Owen, T. Dunning, and E. Friedman, Meet Apache Mahout, Manning Publications Co. Sound View Ct. 3B Greenwich, CT 06830, 2010. [Online]. Available: http://manning.com/owen/

[44] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," Proc. VLDB Endow., vol. 5, no. 8, p. 716–727, apr 2012. [Online]. Available: https://doi.org/10.14778/2212351.2212354

[45] J. Wei, K. Chen, Y. Zhou, Q. Zhou, and J. He, "Benchmarking of distributed computing engines spark and graphlab for big data analytics," in Proceedings, 2016 IEEE Second International Conference on Big Data Computing Service and Applications, BigDataService 2016. United States: IEEE, May 2016, pp. 10–13, -; 2nd IEEE International Conference on Big Data Computing Service and Applications, Bigdata Service 2016 ; Conference date: 29-03-2016 Through 01-04-2016.

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, ´S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, ´Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf

[47] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14). Broomfield, CO: USENIX Association, oct 2014, pp. 583–598. [Online]. Available: https://www.usenix.org/conference/osdi14/technicalsessions/presentation/limu

[48] B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," J. Adv. Inf. Technol., vol. 1, no. 1, feb 2010.

[49] V. N. Vapnik, "An overview of statistical learning theory," IEEE Trans. Neural Netw., vol. 10, no. 5, pp. 988–999, 1999.

[50] H. Brucher, G. Knolmayer, and M. A. Mittermayer, ¨ Document classification methods for organizing explicit knowledge, 2002.

[51] "Support vector machines," vol. 1, no. (3), (2001).

[52] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in Proceedings of the seventh international conference on Information and knowledge management- CIKM '98. New York, New York, USA: ACM Press, 1998.

[53] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in Machine Learning: ECML-98, ser. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142.

[54] G. Siolas and F. d'Alche Buc, "Support vector machines based on a semantic ´kernel for text categorization." vol. 5, 02 2000, pp. 205 – 209 vol.5.

[55] Y. Yang and X. Liu, “A re-examination of text categorization methods,” in Proceedings of the 22nd annual international ACM SIGIRconference on Research and development in information retrieval - SIGIR ’99. New York, New York, USA: ACM Press, 1999.

[56] G. Mishne, “Experiments with mood classification in blog posts,” Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access, vol. 19, pp. 321–327, 2005.

[57] T. Danisman and A. Alpkocak, “Feeler: Emotion classification of text using vector space model,” AISB 2008 Convention Communication, vol. 1, 2008.

[58] K. Roberts, M. A. Roach, J. Johnson, J. Guthrie, and S. M. Harabagiu, “EmpaTweet: Annotating and detecting emotions on twitter,” in LREC, 2012, pp. 3806–3813.

[59] M. Purver and S. Battersby, “Experimenting with distant supervision for emotion classification,” in Proceedings of the 13th Conference of the European Chapter. Association for Computational Linguistics, 2012, pp. 482–491.

[60] K. P. Murphy, Naive bayes classifiers., 2006.

[61] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, “Harnessing twitter“ big data” for automatic emotion identification,” in Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom). IEEE, 2012, pp. 587–592.

[62] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. Cambridge, England: Cambridge University Press, jun 2012.

[63] M. C. Jain and V. Y. Kulkarni, “TexEmo: Conveying emotion from text-the study,” International Journal of Computer Applications, vol. 86, no. 4, 2014.

[64] S. M. Kim, A. Valitutti, and R. A. Calvo, “Evaluation of unsupervised emotion models to textual affect recognition,” in Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Association for Computational Linguistics, 2010, pp. 62–70.

[65] A. NEVIAROUSKAYA, H. PRENDINGER, and M. ISHIZUKA, “Affect analysis model: novel rule-based approach to affect sensing from text,” Natural Language Engineering, vol. 17, no. 1, p. 95–135, 2011.

[66] D. T. Ho and T. H. Cao, “A high-order hidden markov model for emotion detection from textual data,” in Knowledge Management and Acquisition for Intelligent Systems, ser. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 94–105.

[67] C. Strapparava and R. Mihalcea, “Learning to identify emotions in text,” in Proceedings of the 2008 ACM symposium on Applied computing - SAC ’08. New York, New York, USA: ACM Press.

[68] N. Gupta, M. Gilbert, and G. Di Fabbrizio, “Emotion detection in email customer care,” in Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, ser. CAAGET ’10. USA: Association for Computational Linguistics, 2010, p. 10–16.

[69] J. T. Hancock, C. Landrigan, and C. Silver, “Expressing emotion in text-based communication,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’07. New York, New York, USA: ACM Press, 2007.

[70] F.-R. Chaumartin, “UPAR7: A knowledge-based system for headline sentiment tagging,” in Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). Prague, Czech Republic: Association for Computational Linguistics, jun 2007, pp. 422–425. [Online]. Available: https://aclanthology.org/S07-1094

[71] J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, “Towards building a social emotion detection system for online news,” Future Generation Computer Systems, vol. 37, pp. 438–448, 2014, special Section: Innovative Methods and Algorithms for Advanced Data-Intensive Computing Special Section: Semantics, Intelligent processing and services for big data Special Section: Advances in Data-Intensive Modelling and Simulation Special Section: Hybrid Intelligence for Growing Internet and its Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X13002033

[72] G. Mishne and M. Rijke, “Capturing global mood levels using blog posts.” 01 2006, pp. 145–152.

[73] p. . A. v. . . p. . . y. . . Mohammad, S M", "Emotional tweets," in Proceedings of the First Joint Conference on Lexical and Computational Semantics.

[74] M. Hasan, E. O. Agu, and E. A. Rundensteiner, "Using hashtags as labels for supervised learning of emotions in twitter messages," 2014.

[75] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," Proceedings of the International AAAI Conference on Web and Social Media, vol. 5, no. 1, pp. 450–453, Aug. 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14171

[76] J. Posner, J. A. Russell, and B. Peterson, "The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology," Dev. Psychopathol., vol. 17, no. 3, pp. 715–734, 2005.

[77] P. Ekman, "An argument for basic emotions," Cognition and Emotion, vol. 6, no. 3-4, pp. 169–200, 1992. [Online]. Available: https://doi.org/10.1080/02699939208411068

[78] R. Plutchik, Emotion, psychopathology, and psychotherapy, R. Plutchik and H. Kellerman, Eds. Elsevier Science, jan 2013.

[79] P. Shaver, J. Schwartz, D. Kirson, and C. O'Connor, "Emotion knowledge: Further exploration of a prototype approach," J. Pers. Soc. Psychol., vol. 52, no. 6, pp. 1061–1086, 1987.

[80] H. Lovheim, "A new three-dimensional model for emotions and monoamine neurotransmitters," Med. Hypotheses, vol. 78, no. 2, pp. 341–348, feb 2012.

[81] S. Tomkins, Affect imagery consciousness: Volume I: The positive affects. Springer publishing company, 1962.

[82] ——, Affect imagery consciousness: Volume II: The negative affects. Springer publishing company, 1963.

[83] D. C. Rubin and J. M. Talarico, "A comparison of dimensional models of emotion: evidence from emotions, prototypical events, autobiographical memories, and words," Memory, vol. 17, no. 8, pp. 802–808, nov 2009.

[84] J. A. Russell, "A circumplex model of affect," J. Pers. Soc. Psychol., vol. 39, no. 6, pp. 1161–1178, 1980.

[85] M. M. Bradley, M. K. Greenwald, M. C. Petry, and P. J. Lang, "Remembering

pictures: Pleasure and arousal in memory," J. Exp. Psychol. Learn. Mem. Cogn., vol. 18, no. 2, pp. 379–390, 1992.

[86] D. Watson and A. Tellegen, "Toward a consensual structure of mood," Psychol. Bull., vol. 98, no. 2, pp. 219–235, sep 1985.

[87] J. Ranganathan and A. Tzacheva, "Emotion mining in social media data," Procedia Computer Science, vol. 159, pp. 58–66, 2019, knowledgeBased and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919313389

[88] M. Munezero, C. S. Montero, E. Sutinen, and J. Pajunen, "Are they different? affect, feeling, emotion, sentiment, and opinion detection in text," IEEE transactions on affective computing, vol. 5, no. 2, pp. 101–111, 2014.

[89] A. Seyeditabari, N. Tabari, and W. Zadrozny, "Emotion detection in text: a review," 2018.

[90] J. Ranganathan, A. S. Irudayaraj, A. Bagavathi, and A. A. Tzacheva, "Actionable pattern discovery for sentiment analysis on twitter data in clustered environment," J. Intell. Fuzzy Syst., vol. 34, no. 5, pp. 2849–2863, may 2018

[91]S. V. M. K. S. T. A. Ranganathan, J., "Emotion classification using recurrent neural network and scalable pattern mining," 2021.

[92] A. Bagavathi, J. Ranganathan, and A. A. Tzacheva, "Action rules for sentiment analysis using twitter," Int. J. Soc. Netw. Min., vol. 3, no. 1, p. 35, 2020.

[93] S. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," CoRR, vol. abs/1308.6297, 2013. [Online]. Available: http://arxiv.org/abs/1308.6297

[94] "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon], author = Mohammad, Saif and Turney, Peter, booktitle = Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text."Los Angeles, CA: Association for Computational Linguistics, jun 2010, pp. 26–34. [Online]. Available: https://aclanthology.org/W10-0204

[95] R. J. M. S. Tzacheva, A., "Actionable pattern discovery for tweet emotions," 2019.

[96] C. Fischer, Z. A. Pardos, R. S. Baker, J. J. Williams, P. Smyth, R. Yu, S. Slater, R. Baker, and M. Warschauer, "Mining big data in education: Affordances and challenges," Review of Research in Education, vol. 44, no. 1, pp. 130–160, 2020. [Online]. Available: https://doi.org/10.3102/0091732X20903304

[97] "Detecting and improving student emotions using actionable pattern discovery in student survey data," 2021.

[98] K. Sangeetha and D. Prabha, "Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for lstm," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 4117–4126, 2021.

[99] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," WIREs Data Mining and Knowledge Discovery, vol. 10, no. 3, p. e1355, 2020. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1355

[100] "Student evaluations in teaching – emotion classification using neural networks," 2020.

[101] R. J. J. R. Tzacheva, A., "Multi-label emotion mining from student comments," July 10-12 2019.

[102] A. R. P. N. . K. Baglodi, "Role of sentiment analysis in education sector in the era of big data: a survey," International Journal of Latest Trends in Engineering and Technology, pp. 022–024, 2017.

[103] J. Tzacheva, A. Ranganathan, "Emotion mining from student comments a lexicon basedapproach for pedagogical innovation assessment." July 10-12 2018.

[104] A. A. Tzacheva and A. Easwaran, "Emotion detection and opinion mining from student comments for teaching innovation assessment," 2021.

[105] A. A. Tzacheva and J. Ranganathan, "Pattern discovery from student feedback: Identifying factors to improve student emotions in learning," International Journal of Educational and Pedagogical Sciences, vol. 14, no. 12, pp. 1260– 1265, 2020. [Online]. Available: https://publications.waset.org/vol/168

[106] A. A. Tzacheva, R. A. Shankar, S. Ramachandran, and A. Bagavathi, "Action rules of lowest cost and action set correlations," vol. 172, no. 4, pp. 399–412, 2020.

[107] R. P. T. A. R. Z. Ranganathan, J., "Emotion mining and attribute selection for actionable recommendations - to improve customer satisfaction," 2021.

[108] M. Mach-Krol and B. Hadasik, "On a certain research gap in big data mining ´for customer insights," Applied Sciences, vol. 11, no. 15, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/15/6993

[109] N. Tabari, P. Biswas, B. Praneeth, A. Seyeditabari, M. Hadzikadic, and W. Zadrozny, "Causality analysis of Twitter sentiments and stock market returns," in Proceedings of the First Workshop on Economics and Natural Language Processing. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 11–19. [Online]. Available: https://aclanthology.org/W18-3102

[110] P. J. Lang, M. M. Bradley, and B. N. Cuthbert, "International affective picture

system (IAPS): Technical manual and affective ratings. NIMH center for the study of emotion and attention," pp. 39–58, 1997.

[111] M. M. Bradley and P. J. Lang, The International Affective Digitized Sounds: IADS-2:Affective ratings of sounds and instruction manual, Gainesville, FL, Tech, 2007.

[112] F. A. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis ˚in microblogs," mar 2011.

[113] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," J. Lang. Soc. Psychol., vol. 29, no. 1, pp. 24–54, mar 2010.

[114] S. M. Mohammad and P. D. Turney, “Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon,” in Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text. Association for Computational Linguistics, 2010, pp. 26–34.

[115] ——, “Crowdsourcing a word-emotion association lexicon,” Comput. Intell.,vol. 29, no. 3, pp. 436–465, aug 2013.

[116] ——, “Nrc emotion lexicon,” NRC Technical Report, 2013.

[117] S. Mohammad, “#emotional tweets,” in *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). Montreal, Canada: Association for Computational Linguistics, 7-8 June 2012, ´pp. 246–255. [Online]. Available: http://www.aclweb.org/anthology/S12-1033

[118] C. Strapparava and A. Valitutti, “Wordnet-affect: an affective extension of wordnet,” Vol 4., vol. 4, pp. 1083–1086, 01 2004.

[119] A. Valitutti, C. Strapparava, and O. Stock, “Developing affective lexical resources,” PsychNology Journal, vol. 2, pp. 61–83, January 2004. [Online]. Available: https://www.microsoft.com/en-us/research/publication/developingaffective-lexical-resources/

[120] C. Strapparava, A. Valitutti, and O. Stock, “The affective weight of lexicon,” Proceedings of LREC, pp. 423–426, 01 2006.

[121] A. G. Shahraki and O. R. Za¨ıane, “Lexical and learning-based emotion mining from text,” 2017.

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

[123] 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, vol. 9. Springer, 2006, pp. 143–154.

[124] 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.

[125] 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.

[126] 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.

[127] S. Im and Z. W. Ras, “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.

[128] L. Cao, “Actionable knowledge discovery and delivery,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, 2012.

[129] J. Barrett, E. B. Caruthers, K. L. German, E. Hamby, M. R, Lofthus, S. Srinivas, Xerox, E. Ells, and Klem, “From data to actionable knowledge: A collaborative effort with educators,” 2011.

[130] S. Greco, B. Matarazzo, N. Pappalardo, and R. S. *, “Measuring expected effects of interventions based on decision rules,” Journal of Experimental & Theoretical Artificial Intelligence, vol. 17, no. 1-2, pp. 103–118, 2005. [Online]. Available: https://doi.org/10.1080/09528130512331315864

[131] A. Bagavathi and A. Tzacheva, “Rule based systems in a distributed environment: Survey,” 06 2017

[132] A. Dardzinska, Action rules mining. Springer, 2012, vol. 468.

[133] H. Kaur, “Actionable rules: Issues and new directions.” 01 2005, pp. 61–64.

[134] R. Z. W. Dardzinska, A., “On rules discovery from incomplete information systems,” in ICDM’03 Workshop on Foundations and New Directions of Data Mining, November 2003.

[135] A. Dardzinska and Z. W. Raundefined, “Cooperative discovery of ´interesting action rules,” in Proceedings of the 7th International Conference on Flexible Query Answering Systems, ser. FQAS’06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 489–497. [Online]. Available: https://doi.org/10.1007/1176625441

[136] Z. He, X. Xu, S. Deng, and R. Ma, “Mining action rules from scratch,” Expert Syst. Appl., vol. 29, no. 3, p. 691–699, oct 2005. [Online]. Available: https://doi.org/10.1016/j.eswa.2005.04.031

[137] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in Proceedings of the 20th International Conference on Very Large Data Bases, ser. VLDB ’94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.

[138] Z. Ras and A. Wieczorkowska, “Action-rules: How to increase profit of a company,” vol. 1910, 09 2000, pp. 587–592.

[139] G. Adomavicius and A. Tuzhilin, “Discovery of actionable patterns in databases: the action hierarchy approach,” ERN: Technology (Topic), 1997.

[140] B. Liu, W. Hsu, and S. Chen, “Using general impressions to analyze discovered classification rules,” 04 1999.

[141] A. Silberschatz and A. Tuzhilin, “On subjective measures of interestingness in knowledge discovery,” in Proceedings of the First International Conference on Knowledge Discovery and Data Mining, ser. KDD’95. AAAI Press, 1995, p. 275–281.

[142] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Pearson Education, 2006

[143] Z. W. Ras and A. A. Tzacheva, “Discovering semantic inconsistencies to improve action rules mining,” in Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM 03 Conference held in Zakopane, Poland, June 2-5, 2003, ser. Advances in Soft Computing, M. A. Klopotek, S. T. Wierzchon, and K. Trojanowski, Eds. Springer, 2003, pp. 301–310.

[144] Z. Ras, A. Tzacheva, L.-S. Tsay, and O. Giirdal, “Mining for interesting action rules,” in IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2005, pp. 187–193.

[145] A. A. Tzacheva and Z. W. Ras, “Action rules mining,” ´ International Journal of Intelligent Systems, vol. 20, no. 7, pp. 719–736, 2005. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.20092

[146] Z. W. Raundefined and A. Dardzinska, “Action rules discovery, a new ´simplified strategy,” ser. ISMIS’06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 445–453. [Online]. Available: https://doi.org/10.1007/1187560451

[147] A. Tzacheva and Z. Ras, “Constraint based action rule discovery with single classification rules,” vol. 4482, 05 2007, pp. 322–329.

[148] Z. Pawlak, “Rough sets: Theoretical aspects of reasoning about data] vol=9, booktitle=Springer Science & Business Media,,” 2012.

[149] T. L. S. . D. A. Ras, Z. W., “Mining e-action rules. mining complex data,85.”

[150] Z. Ras, L.-S. Tsay, and A. Dardzinska, ´ Tree-Based Algorithms for Action Rules Discovery, 10 2008, vol. 165, pp. 153–163.

[151] Z. Ras, E. Wyrzykowska, and H. Wasyluk, “Aras: Action rules discovery based on agglomerative strategy,” vol. 4944, 09 2007, pp. 196–208.

[152] Z. W. Ras, “Cooperative knowledge-based systems,” Intelligent Automation & Soft Computing, vol. 2, no. 2, pp. 193–201, 1996. [Online]. Available: https://doi.org/10.1080/10798587.1996.10750667

[153] A. A. Tzacheva, “Summaries of action rules by agglomerative clustering.” 2010.

[154] S. Im, Z. Ras, and H. Wasyluk, “Action rule discovery from incomplete data,” ´Knowl. Inf. Syst., vol. 25, no. 1, pp. 21–33, oct 2010.

[155] Z. Ras and A. Dardzinska, "Action rules discovery without pre-existing ´classification rules," vol. 5306, 10 2008, pp. 181–190.

[156] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between ´sets of items in large databases," in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '93. New York, NY, USA: Association for Computing Machinery, 1993, p. 207–216. [Online]. Available: https://doi.org/10.1145/170035.170072

[157] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in Proceedings of the 2001 IEEE International Conference on Data Mining, ser. ICDM '01. USA: IEEE Computer Society, 2001, p. 163–170.

[158] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53–87, jan 2004.

[159] A. Savasere, E. Omiecinski, and S. B. Navathe, "An efficient algorithm for mining association rules in large databases," in Proceedings of the 21th International Conference on Very Large Data Bases, ser. VLDB '95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, p. 432–444.

[160] M. Zaki, "Scalable algorithms for association mining," IEEE Transactions on Knowledge and Data Engineering, vol. 12, no. 3, pp. 372–390, 2000.

[161] R. Paul and A. S. M. L. Hoque, "Mining irregular association rules based on action amp; non-action type data," in 2010 Fifth International Conference on Digital Information Management (ICDIM), 2010, pp. 63–68.

[162] K. Wang, S. Zhou, and J. Han, "Profit mining: From patterns to actions," 03 2002, pp. 70–87.

[163] B. Padmanabhan, "A belief-driven method for discovering unexpected patterns," in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, ser. KDD'98. AAAI Press, 1998, p. 94–100.

[164] K. Wang, Y. Jiang, and L. V. S. Lakshmanan, "Mining unexpected rules by pushing user dynamics," in KDD 03, 2003.

[165] B. Arunasalam and S. Chawla, "Cccs: a top-down associative classifier for imbalanced class distribution," in KDD '06, 2006.

[166] H. Zhang, Y. Zhao, L. Cao, and C. Zhang, "Combined association rule mining," 05 2008, pp. 1069–1074.

[167] M. Kryszkiewicz, Representative association rules, 04 2006, vol. 1394, pp. 198–209.

[168] J. Saquer and J. S. Deogun, "Using closed itemsets for discovering representative association rules," in ISMIS, 2000.

[169] Y. Qiao, K. Zhong, H. Wang, and X. Li, "Developing event-condition-action rules in real-time active database," in Proceedings of the 2007 ACM Symposium on Applied Computing, ser. SAC '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 511–516. [Online]. Available: https://doi.org/10.1145/1244002.1244120

[170] A. Tzacheva and L.-S. Tsay, "Tree-based construction of low-cost action rules," Fundamenta Informaticae, vol. 86, 01 2008.

[171] Z. Ras and A. Dardzinska, "From data to classification rules and actions," ´ Int. J. Intell. Syst., vol. 26, pp. 572–590, 06 2011.

[172] D. Goldin, S. Srinivasa, and V. Srikanti, "Active databases as information systems," in Proceedings. International Database Engineering and Applications Symposium, 2004. IDEAS '04., 2004, pp. 123–130.

[173] N. W. Paton and O. D´ıaz, "Active database systems," ACM Comput. Surv., vol. 31, no. 1, p. 63–103, mar 1999. [Online]. Available: https://doi.org/10.1145/311531.311623

[174] Y. Qiao, K. Zhong, H. Wang, and X. Li, "Developing event-condition-action rules in real-time active database," in Proceedings of the 2007 ACM Symposium on Applied Computing, ser. SAC '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 511–516. [Online]. Available: https://doi.org/10.1145/1244002.1244120

[175] H. Touati, Z. W. Ras, J. Studnicki, and A. Wieczorkowska, “Mining surgical meta-actions effects with variable diagnoses’ number,” in ISMIS, 2014.

[176] A. A. Tzacheva and Z. W. Ras, “Association action rules and action paths triggered by meta-actions,” in 2010 IEEE International Conference on Granular Computing, 2010, pp. 772–776.

[177] K. Wang, J. Yuelong, and A. Tuzhilin, “Mining actionable patterns by role models,” 05 2006, pp. 16– 16.

[178] Z. Ras, K. Tarnowska, J. Kuang, L. Daniel, and D. Fowler, “User friendly nps-based recommender system for driving business revenue,” 07 2017, pp. 34–48.

[179] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’04. New York, NY, USA: Association for Computing Machinery, 2004, p. 168–177. [Online]. Available: https://doi.org/10.1145/1014052.1014073

[180] K. Tarnowska, Z. Ras, L. Daniel, and D. Fowler, Visual Analysis of Relevant Features in Customer Loyalty Improvement Recommendation, 01 2018, pp. 269–293.

[181] V. A. and S. Sonawane, “Sentiment analysis of twitter data: A survey of techniques,” International Journal of Computer Applications, vol. 139, no. 11, p. 5–15, Apr 2016. [Online]. Available: http://dx.doi.org/10.5120/ijca2016908625

[182] 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.

[183] A. Tzacheva and J. Ranganathan, “Pattern discovery from student feedback -identifying factors to improve student emotions in learning.” IOP Publishing, Jan 2021.

[184] 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.cfmid=1863103.1863113

[185] J. Ranganathan, A. Irudayaraj, and A. Tzacheva, “Action rules for sentiment analysis on twitter data using spark,” 11 2017, pp. 51–60.

[186] M. Al-Mardini, A. Hajja, Z. Ras, L. Clover, D. Olaleye, Y. Park, J. Paulson, and Y. Xiao, “Reduction of readmissions to hospitals based on actionable knowledge discovery and personalization,” vol. 613, 04 2016, pp. 39–55.

[187] 2022 February 8.

[188] 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