

# Computationally Efficient Approximation Algorithm of Dominance Based Rough Set Approach

Aleena Ahmad\*, Usman Qamar\*, Summair Raza\*\*

\*Computer Engineering Department, College of Electrical and Mechanical Engineering  
National University of Sciences and Technology (NUST), Islamabad, Pakistan

\*\* Department of Computer Science, Virtual University of Pakistan

aahmad17@ce.ceme.edu.pk, usmanq@ceme.nust.edu.pk, sraza@vu.edu.pk

**Abstract**— Information systems contain preference ordered decision attributes now a day. To deal with them Dominance based Rough Set Approach (DRSA) is used. Two important concepts in DRSA are upper and lower approximations, both of them are computationally heavy algorithms. Reduction in its computational complexity helps in many advance applications of DRSA. Our proposed algorithm reduces its complexity by making changes in conventional procedure. Our solution stops the algorithm from precomputing the same thing from scratch. We tested our proposed solution on 5 datasets from UCI. There is 50% reduction in complexity of comparison step, which will lead to overall reduction of computational complexity for calculating lower and upper approximations.

**Keywords**—Dominance-based Rough Set Approach, Computational complexity, Lower approximation, Upper approximation, Preference ordered decision attributes

## I. INTRODUCTION

A huge number of data is accessible in today's world. Data mining tools and techniques are used to process this data and obtain useful information from it. Data is processed to perform text mining, sequence pattern mining and classification [1].

Classification is used to predict class labels and also helpful in forming association rules. Many types of classification algorithms are being used. Rough Set Theory (RST) is one of them. The idea of RST was given by Palwak [2] [3] [4]. RST is a mathematical tool which plays an important role in dealing with inconsistent, imprecise and incomplete knowledge. RST is used for classification, finding relevant features, reduction of redundant objects and facilitates rule extraction. In RST indiscernibility relation, equivalence class, lower and upper approximations are some key concepts. If a data set can be fully divided in equivalence classes, then it is called a crisp set otherwise it is called a rough set. Lower and upper approximations are to be calculated for rough sets. Many applications of RST such that image segmentation [5], power system's fault diagnosis and in enhancement classification [6] [7] exists.

Conventional RST is utilized for classifying binary class labels. People now a day have preference in their daily work life. For this reason, most of the information systems contain decision attributes with preference order [8]. The extended

version of classical RST is introduced by Greco et al named as Dominance Based Rough Set Approach (DRSA) [9].

In DRSA, dominance relation replaces the indiscernibility relation. A major benefit of DRSA is its usage for multi-criteria analysis as it processes information of preference ordered domain.

A core step in DRSA is to compute lower and upper approximation. Computational cost of this step is high because it traverses the instances twice while comparing for dominance relationship [10]. To overcome this problem, we have proposed a solution which contains some changes in traditional approximation algorithm. Our proposed solution decreases the computation time of finding lower and upper approximations.

In rest of the paper, preliminaries are defined first. After that related work is shown and challenges in computation of approximations are described. A solution is then proposed and its results are shown. At the end conclusion of the paper is given.

## II. PRELIMINARIES

Some terms for conventional RST are as follows:

Universe = A decision system which contains finite items or objects = U.

Conditional attributes = Objects' characteristics = C.

Decision attribute = class label = D

Mathematically representation is as follows:

$$\alpha = (U, C \cup D) \quad (1)$$

Some terms for conventional DRSA are as follows:

Universe = A decision system which contains finite items or objects = U

$Q = (C \cup D)$  = Attributes having ordinal and finite domain and domain is finite. It contains both conditional and decision attributes.

$V = V_q$  is a value set for criteria  $q = U_{q \in Q} V_q$ .

$f(x, q)$  = To an item  $x$  of attribute  $q$  a value  $V_q$  is assigned in that function.

Mathematically representation is as follows:

$$\alpha = (U, Q, V, F) \quad (2)$$

Table 1 contains a sample decision system. This is taken from Wikipedia [11].

There are 10 items ( $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}$ ) in universe U of Table 1. Mathematics ( $q_1$ ), physics ( $q_2$ ) and

literature (q3) are conditional attributes and global score (d) is decision attribute.

TABLE 1. HIGH SCHOOL EVALUATION

| Object (student) | q1 (mathematics) | q2 (physics) | q3 (literature) | d (global score) |
|------------------|------------------|--------------|-----------------|------------------|
| x1               | M                | M            | B               | B                |
| x2               | G                | M            | B               | M                |
| x3               | M                | G            | B               | M                |
| x4               | B                | M            | G               | B                |
| x5               | B                | B            | M               | B                |
| x6               | B                | M            | M               | M                |
| x7               | G                | G            | B               | G                |
| x8               | G                | M            | M               | M                |
| x9               | M                | M            | G               | G                |
| x10              | G                | M            | G               | G                |

#### A. Dominance

For each criterion in  $P \subseteq C$ , item  $g$  dominates  $h$  if item  $g$  is better than item  $h$ . There are two kinds of dominance negative dominance  $D_p^-(g)$  ( $g$  is dominated by  $h$ ) and positive dominance  $D_p^+(g)$  ( $g$  dominates  $h$ ).

$$D_p^+(g) = \{h \in U; g D_p h\} \quad (3)$$

$$D_p^-(g) = \{h \in U; h D_p g\} \quad (4)$$

For the item x5 in sample Table 1 and  $P = \{q1, q2, q3\}$

$$D_p^+(x) = \{x4, x5, x6, x8, x9, x10\}$$

$$D_p^-(x) = \{x5\}$$

#### B. Classes

In case of DRSA, Class downward and upward unions are found by using following mathematical equations:

$$Cl_t^{\geq} = U_{s \geq t} Cl_s \quad (5)$$

$$Cl_t^{\leq} = U_{s \leq t} Cl_s \quad (6)$$

#### C. Lower Approximation

It contains a set of items which with certainty belongs to the class label of decision attribute. Mathematically finding lower approximation is represented as:

$$\underline{P}(Cl_t^{\geq}) = \{x \in U; D_p^+(x) \subseteq Cl_t^{\geq}\} \quad (7)$$

$$\underline{P}(Cl_t^{\leq}) = \{x \in U; D_p^-(x) \subseteq Cl_t^{\leq}\} \quad (8)$$

There are three steps to calculate lower approximation:

**Step 1:** Finding union classes is the first step.

For example, in sample Table 1 for bad = 1 union class is:

$$Cl_1^{\leq} = \{x1, x4, x5\}$$

**Step 2:** According to the greater than and less than operators, dominance is calculated. We have to find  $D_p^+(x)$ , in our case.

$$D_p^+(x1) = \{x1, x2, x3, x7, x8, x9, x10\}$$

$$D_p^+(x4) = \{x4, x9, x10\}$$

$$D_p^+(x5) = \{x4, x5, x6, x8, x9, x10\}$$

**Step 3:** In this step, lower approximation is calculated using proper subset property. For example, by applying above formulas  $\underline{P}(Cl_1^{\leq})$  result is as follows.

$$\underline{P}(Cl_1^{\leq}) = \{x1, x5\}$$

#### D. Upper Approximation

In upper approximation, a set of items are generated that may or may not belong to the certain class label of decision attribute. Mathematically finding upper approximation is represented as:

$$\overline{P}(Cl_t^{\geq}) = \{x \in U; D_p^-(x) \cap Cl_t^{\geq} \neq \emptyset\} \quad (9)$$

$$\overline{P}(Cl_t^{\leq}) = \{x \in U; D_p^+(x) \cap Cl_t^{\leq} \neq \emptyset\} \quad (10)$$

There are three steps to calculate upper approximation:

**Step 1:** Finding union classes is the first step.

For example, in sample Table 1 for bad = 1 union class is:

$$Cl_1^{\leq} = \{x1, x4, x5\}$$

**Step 2:** According to the greater than and less than operators, dominance is calculated. We have to find  $D_p^-(x)$ , in our case.

$$D_p^-(x1) = \{x1\}$$

$$D_p^-(x4) = \{x4, x5, x6\}$$

$$D_p^-(x5) = \{x5\}$$

**Step 3:** In this step, upper approximation is calculated using intersection property. For example, by applying above formulas  $\overline{P}(Cl_1^{\leq})$  result is as follows.

$$\overline{P}(Cl_1^{\leq}) = \{x1, x4, x5\}$$

### III. RELATED WORK

There are three ways to calculate upper and lower approximation of DRSA. Conventional algorithm is the first way. But its cost of computation is high.

In second way, conventional algorithm is divided into chunks and then these chunks run on different hardware processor in parallel. This will help in reducing time of computation. It is an efficient way to speed up the whole process of computation.

In [10] [12] [13], researchers proposed an algorithm for parallel computation of lower and upper approximation using multiple processors. They presented the comparison on different runtime systems (MapReduce) [14] [15]. In [16], researchers used GPU framework to accelerate the heavy computational task of approximations. In [17], researchers divided information systems into many subsystems. Their algorithm ran parallel on each subsystem and each subsystem was processed under different processors. In [18], composition and decomposition of granules took place. According to them having more core processor would lead to more reduction of complexity.

Third way is the incremental approach. In this approach, when number of instances or number of attributes increases, approximations are not calculated from scratch.

In [19] [20], authors achieved dynamic incremental approximation updates by using the former results of conventional DRSA. This approach worked efficiently because mostly data instances were more in ratio as compared to attributes.

In [16], authors worked for multi dimension variations by using the incremental approach. They considered memory consumption in their algorithm too. But still their algorithm has complex architecture.

In [21] [22], authors applied incremental technique on attributes that were hierarchical. They formulized the dynamic nature of attributes and then coarsening by using taxonomies of attributes.

Some recent applications of DRSA still using the conventional algorithm because of its simplicity, but they have to pay for computation then.

In [23], authors improved the DRSA to classify medical data. They updated DRSA for nominal data. Their algorithm consisted of five steps for improved classification of medical domain data classification.

In [24], authors used DRSA for spare parts management. Their framework consisted of three steps, in which rules were generated and classification was performed by using DRSA. They tested their framework on real world data and their efficiency came out to be 96%.

In [25], authors predicted customer behavior and response in airline companies. They generated three kinds of reducts (L,B,U). On the basis of discernibility matrices, reducts could be generated comprehensively.

In [26], authors predicted the number of students who dropped or likely to drop the online courses offered. It consisted of two steps and class consisted of active and at-risk learners. They used the conventional DRSA algorithm.

TABLE 2. RELATED WORK

| Paper                        | Advantages                           | Disadvantages                                                                             |
|------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------|
| [10][11][12][15]<br>[16][17] | Decrease in computational complexity | More hardware utilization (multiple processing units)                                     |
| [18][19]                     | Decrease in computational complexity | Use of conventional DSRA                                                                  |
| [20][21][12]                 | Decrease in computational complexity | 1) single dimension variation<br>2) Complex architectural structure of proposed algorithm |

|                  |                                                                         |                                                          |
|------------------|-------------------------------------------------------------------------|----------------------------------------------------------|
| [15]             | 1) Decrease in computational complexity<br>2) multi dimension variation | Complex architectural structure of proposed algorithm    |
| [22]             | Decrease in computational complexity                                    | Complex architectural structure of proposed algorithm    |
| [23][24][25][26] | Advance applications                                                    | Conventional DRSA (increase in computational complexity) |

#### IV. CHALLENGES IN COMPUTATION

Following is the pseudo code for second step of calculating approximation:

```

//For dominance positive
For k=1 to  $|Cl_t^{\leq}/Cl_t^{\leq}|$ 
  For c=1 to  $|U|$ 
    If  $X_c \geq X_{kt}$ 
       $D_P^+(X_k) = D_P^+(X_k) \cup X_c$ 
    End-if
  End-For
End-For

//For dominance negative
For k=1 to  $|Cl_t^{\geq}/Cl_t^{\geq}|$ 
  For c=1 to  $|U|$ 
    If  $X_c \leq X_{kt}$ 
       $D_P^-(X_k) = D_P^-(X_k) \cup X_c$ 
    End-if
  End-For
End-For

```

Figure 1. Conventional Pseudo Code for Dominance Positive and Dominance Negative

In this step, dominance positive and dominance negative is calculated against each item in  $Cl_t^{\leq}$  or  $Cl_t^{\geq}$ . It is a must step and this will lead to slow processing. Also in conventional algorithm due to repetition of items in class union, items positive and negative dominance is calculated again and again.

In case of our sample of table (2) union classes are as follows:

$$\begin{aligned}
 Cl_1^{\leq} &= \{x1, x4, x5\} \\
 Cl_2^{\leq} &= \{x1, x2, x3, x4, x5, x6, x8\} \\
 Cl_2^{\geq} &= \{x2, x3, x6, x7, x8, x9, x10\}
 \end{aligned}$$

$$Cl_3^{\geq} = \{x7, x9, x10\}$$

When we follow the conventional algorithm all the items pass through step 2 twice. This increases the computational time and cost.

Second issue is some items who doesn't have any dominance relation i.e. positive or negative, still goes through the process again. For example, in our case x1 and x7 has no dominance positive and dominance negative relationship still after processing x1, we have to process x7 against x1 again. Although there is no need of that overhead.

To overcome both the issues we proposed a solution that will help in reducing the computational complexity of calculating approximations. This will lead to overall reduction in computational time of DRSA.

## V. PROPOSED SOLUTION

Instead of calculating  $D_p^+(x)$  and  $D_p^-(x)$  for all union classes of decision attribute criteria, just calculate  $D_p^+(x)$  and  $D_p^-(x)$  for all items in the data set once. Store the values in a 2D array so that when you need them in 3<sup>rd</sup> step, you can retrieve them without comparing the items again. By doing this you don't have to compute  $D_p^+(x)$  and  $D_p^-(x)$  again for any item  $x$  in the universe  $U$ .

For second issue maintain a value 1 in case of no relation between two items in 2D array. Check the value in the array before comparing so that you don't have to compare again the same thing.

Even after that we can check negative dominance easily by using positive dominance results in 2D array.

Pseudo code of proposed solution:

```
//For dominance positive
For k=1 to |U|
  For c=1 to |U|
    If (arr[xc][xk]!=1)
      If  $X_c \geq X_{kt}$ 
         $D_p^+(X_k) = D_p^+(X_k) \cup X_c$ 
        //store in 2D array
      End-if
    End-For
  End-For

//For dominance negative
For k=1 to |U|
  For c=1 to |U|
    If (arr[xc][xk]!=1)
      If  $X_c \leq X_{kt}$ 
         $D_p^-(X_k) = D_p^-(X_k) \cup X_c$ 
        //store in 2D array
      End-if
    End-For
  End-For
```

**Figure 2.** Proposed Pseudo Code for Dominance Positive and Dominance Negative

After that you have to perform step 1 and 3 as described in conventional algorithm to find approximations.

Comparison part plays the most important role. The more time this part run the more time code will take to run. In the table given below is the comparison of conventional vs our proposed solution of the sample Table 1.

**TABLE 3.** COMPARISON

|                                                | Conventional Algorithm | Proposed solution | Proposed solution with no relation extension | Proposed solution with no relation extension and dominance negative previous checking |
|------------------------------------------------|------------------------|-------------------|----------------------------------------------|---------------------------------------------------------------------------------------|
| <b>Dominance positive comparison statement</b> | Runs 200 times         | Runs 100 times    | Runs 78 times                                | Runs 78 times                                                                         |
| <b>Dominance negative comparison statement</b> | Runs 200 times         | Runs 100 times    | Runs 78 times                                | Runs 78 times                                                                         |
| <b>Total</b>                                   | 400                    | 200               | 156                                          | 78                                                                                    |

You can clearly see the no of times comparison statement run in proposed solution is half the no of times it run in conventional algorithm. And if we apply no relation condition it reduces it to 156. You can also see by applying negative dominance part of our solution this statement runs only 78 times.

We can say that this solution reduces the time complexity of this step of algorithm to 50 % and may be more.

## VI. ANALYSIS AND RESULTS

We implemented our proposed solution part 1, that is calculating dominance on the start of the algorithm and then calculate the rest part, on 5 data sets taken from UCI repository (UCI) [27].

Data sets are described in the table below. All the data sets have ordinal preference domain.

**TABLE 4.** DATASET

| Dataset                 | Instances | Attributes | Dataset Attribute Characteristics / |
|-------------------------|-----------|------------|-------------------------------------|
| <b>Car Evaluation</b>   | 1728      | 6          | Categorical                         |
| <b>Poker Hand</b>       | 1025010   | 11         | Categorical, Integer                |
| <b>DeliciousMIL</b>     | 12234     | 8519       | Integer                             |
| <b>Gas Sensor Array</b> | 13910     | 128        | Real                                |

|                             |     |    |      |
|-----------------------------|-----|----|------|
| <b>Glass Identification</b> | 214 | 10 | Real |
|-----------------------------|-----|----|------|

We used visual studio 2015 for implementing the pseudo code of our proposed algorithm described above. We didn't implement the part 2 (no relation) and part 3 (dominance negative) now, we will cover this in future. We used diagnostic tool window to find time stamps.

Comparison according to execution time is given in the table below.

**TABLE 5.** COMPUTATION TIME COMPARISON

| Data Set                    | Conventional | Proposed | Reduction in complexity |
|-----------------------------|--------------|----------|-------------------------|
| <b>Car Evaluation</b>       | 286 ms       | 142 ms   | More than 50%           |
| <b>Poker Hand</b>           | 500 mins     | 240 mins | More than 50%           |
| <b>DeliciousMIL</b>         | 190 mins     | 90 mins  | More than 50%           |
| <b>Gas Sensor Array</b>     | 4 mins       | 1.5 mins | More than 50%           |
| <b>Glass Identification</b> | 28 ms        | 10 ms    | More than 50%           |

You can clearly see that in specific part of algorithm there is more than 50% reduction in time complexity.

Following are the graphical representation of our results for all 5 data sets.

In figure 4, data sets having large number of instances and attributes are covered and unit taken is minutes for comparison. In figure 3, data sets having less number of instances and attributes are covered and unit taken is milliseconds for comparison.

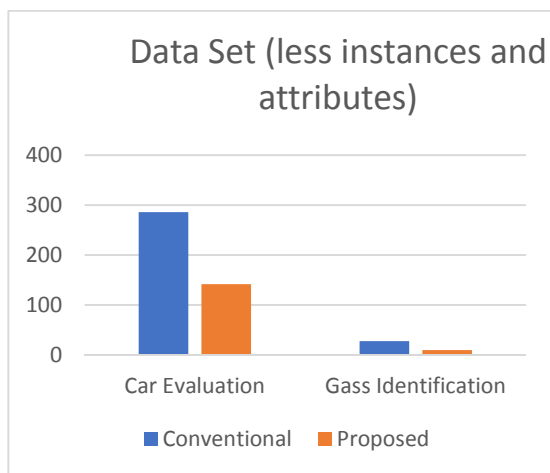


**Figure 3.** Comparison graph 1

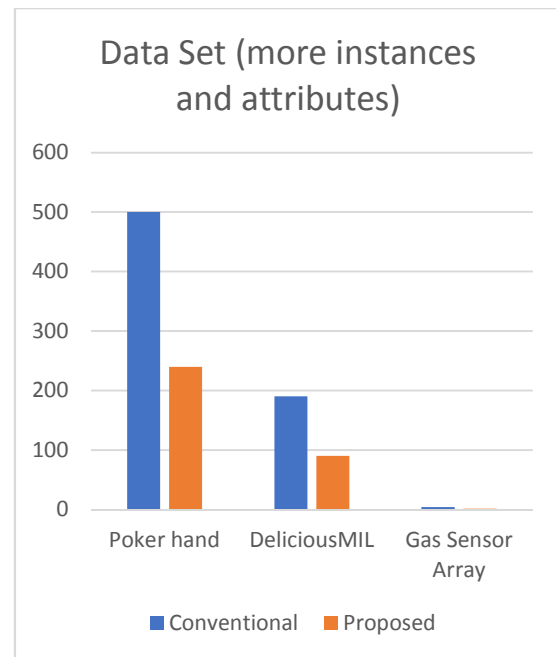


**Figure 4.** Comparison graph 2

It is clear from the above graphs that by manipulating the steps and by performing it on right time will lead to reduction of computational complexity. It will also reduce the cost overall.

Till now we reduce the computational complexity of this step by 50%. We can further reduce this by applying our 2<sup>nd</sup> and 3<sup>rd</sup> part of proposed algorithm.

## VII. CONCLUSIONS

Due to huge amount of data in today's world, computational complexity is an important factor to deal with. In DRSA finding lower and upper approximations are computationally heavy in conventional algorithm. By our proposed algorithm this heavy computational part lost its computational complexity up to 50%. We simply avoid repetition in computation of dominance. We have also proposed two more extensions which we will prove on large data in future.

Some people use incremental approach as described in our related work. One can use our proposed algorithm to reduce the complexity of full algorithm in incremental update approach.

## REFERENCES

- [1] Zucco, Chiara. (2018). Data Mining in Bioinformatics. Encyclopedia of Bioinformatics and Computational Biology, ABC of Bioinformatics, 328.
- [2] Pawlak, Z., Grzymala-Busse, J., Slowinski, R. and Ziarko, W. (1995). Rough sets. *Communications of the ACM*, 38(11), pp.88-95.
- [3] Pawlak, Z. and Skowron, A. (2007). Rudiments of rough sets. *Information Sciences*, 177(1), pp.3-27.
- [4] Pawlak, Z. and Skowron, A. (2007). Rough sets: Some extensions. *Information Sciences*, 177(1), pp.28-40.
- [5] Roy, Payel, Srijan Goswami, Sayan Chakraborty, Ahmad Taher Azar, and Nilanjan Dey. "Image segmentation using rough set theory: a review." *International Journal of Rough Sets and Data Analysis (IJRSDA)* 1, no. 2 (2014): 62-74.

- [6] Rawat, S., Patel, A., Celestino, J. and dos Santos, A. (2016). A dominance based rough set classification system for fault diagnosis in electrical smart grid environments. *Artificial Intelligence Review*, 46(3), pp.389-411.
- [7] Zhang, Q., Xie, Q. and Wang, G. (2016). A survey on rough set theory and its applications. *CAAI Transactions on Intelligence Technology*, 1(4), pp.323-333.
- [8] Chakhar, S., Ishizaka, A., Labib, A. and Saad, I. (2016). Dominance-based rough set approach for group decisions. *European Journal of Operational Research*, 251(1), pp.206-224.
- [9] Greco, S., B. Matarazzo, and R. Slowinski. (1999). Advances in multiple criteria decision making. *The Use of Rough Sets and Fuzzy Sets in Multi Criteria Decision Making*, 14(1).
- [10] Zhang, J., Li, T., Ruan, D., Gao, Z. and Zhao, C. (2012). A parallel method for computing rough set approximations. *Information Sciences*, 194, pp.209-223.
- [11] En.wikipedia.org. (2020). *Dominance-based rough set approach*. [online] Available at: [https://en.wikipedia.org/wiki/Dominance-based\\_rough\\_set\\_approach](https://en.wikipedia.org/wiki/Dominance-based_rough_set_approach) [Accessed 6 Jan. 2020].
- [12] Zhang, J., Wong, J., Li, T. and Pan, Y. (2014). A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems. *International Journal of Approximate Reasoning*, 55(3), pp.896-907.
- [13] Zhang, J., Wong, J., Pan, Y. and Li, T. (2015). A Parallel Matrix-Based Method for Computing Approximations in Incomplete Information Systems. *IEEE Transactions on Knowledge and Data Engineering*, 27(2), pp.326-339.
- [14] Qian, J., Lv, P., Yue, X., Liu, C. and Jing, Z. (2015). Hierarchical attribute reduction algorithms for big data using MapReduce. *Knowledge-Based Systems*, 73, pp.18-31.
- [15] Qian, J., Miao, D., Zhang, Z. and Yue, X. (2014). Parallel attribute reduction algorithms using MapReduce. *Information Sciences*, 279, pp.671-690.
- [16] Wang, S., Li, T., Luo, C. and Fujita, H. (2016). Efficient updating rough approximations with multi-dimensional variation of ordered data. *Information Sciences*, 372, pp.690-708.
- [17] LI, SHAOYONG, TIANRUI LI, and HONGMEI CHEN. (2014). A parallel approach for computing approximations of dominance-based rough sets approach. *Decision Making and Soft Computing: Proceedings of the 11th International FLINS Conference*, pp. 580-585.
- [18] Zhang, J., Zhu, Y., Pan, Y. and Li, T. (2016). Efficient parallel boolean matrix based algorithms for computing composite rough set approximations. *Information Sciences*, 329, pp.287-302.
- [19] Li, S., Li, T. and Liu, D. (2013). Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set. *Knowledge-Based Systems*, 40, pp.17-26.
- [20] Li, S. and Li, T. (2015). Incremental update of approximations in dominance-based rough sets approach under the variation of attribute values. *Information Sciences*, 294, pp.348-361.
- [21] Huang, Q., Li, T., Huang, Y., Yang, X. and Fujita, H. (2020). Dynamic dominance rough set approach for processing composite ordered data. *Knowledge-Based Systems*, 187, p.104829.
- [22] Du, W. and Hu, B. (2016). Dominance-based rough set approach to incomplete ordered information systems. *Information Sciences*, 346-347, pp.106-129.
- [23] Azar, A., Inbarani, H. and Renuga Devi, K. (2016). Improved dominance rough set-based classification system. *Neural Computing and Applications*, 28(8), pp.2231-2246.
- [24] Hu, Q., Chakhar, S., Siraj, S. and Labib, A. (2017). Spare parts classification in industrial manufacturing using the dominance-based rough set approach. *European Journal of Operational Research*, 262(3), pp.1136-1163.
- [25] Liou, J. and Tzeng, G. (2010). A Dominance-based Rough Set Approach to customer behavior in the airline market. *Information Sciences*, 180(11), pp.2230-2238.
- [26] Bouzayane, Sarra, and I. Saad. (2017). Weekly predicting the at-risk MOOC learners using dominance-based rough set approach. *European Conference on Massive Open Online Courses*, pp. 160-169.
- [27] Anon, (2020). [online] Available at: <https://archive.ics.uci.edu/ml/index.php>. [Accessed 6 Jan. 2020].

**Aleena Ahmad** has done BS in Software Engineering (2013-2017) from Punjab University, Lahore. She is currently doing MS in Software Engineering from NUST College of Electrical and Mechanical Engineering. Her field of interest is Data Engineering, specifically dominance based rough set approach.

**Dr. Usman Qamar** has over 15 years of experience in data engineering and decision sciences both in academia and industry having spent nearly 10 years in the UK. He has a Masters in Computer Systems Design from University of Manchester Institute of Science and Technology (UMIST), UK. His MPhil in Computer Systems was a joint degree between UMIST and University of Manchester which focused on feature selection in big data. In 2008/09 he was awarded PhD from University of Manchester, UK. His PhD specialization is in Data Engineering, Knowledge Discovery and Decision Science. His Post PhD work at University of Manchester, involved various research projects including hybrid mechanisms for statistical disclosure (feature selection merged with outlier analysis) for Office of National Statistics (ONS), London, UK, churn prediction for Vodafone UK and customer profile analysis for shopping with the University of Ghent, Belgium. He has also done a post graduation in Medical & Health Research, from University of Oxford, UK, where he worked on evidence-based health care, thematic qualitative data analysis and healthcare innovation and technology. He is director of Knowledge and Data Science Research Centre, a Centre of Excellence at NUST, Pakistan and principal investigator of Digital Pakistan Lab, which is part of National Centre for Big Data and Cloud Computing. He has authored over 150 peer reviewed publications which includes 2 books published by Springer & Co, 21 Book Chapters, 38 Impact Factor Journal Publications with a combined impact factor of 104 (Clarivate Analytics Impact Factor) and over 100 Conference Publications. Many of his papers have been awarded best research paper awards by Higher Education Commission, Pakistan. Because of his extensive publications he is member of Elsevier Advisory Panel. He has successfully supervised 4 PhD students and over 70 master students. Dr. Usman has been able to acquire nearly PKR 100 million in research grants. He has received multiple research awards, including Best Book Award 2017/18 by Higher Education Commission (HEC), Pakistan, Best Researcher of Pakistan 2015/16 by Higher Education Commission (HEC), Pakistan, Best Overall NUST University Researcher Award 2016 and Best College of E&ME researcher award 2016 as well as gold in Research & Development category by Pakistan Software Houses Association (P@SHA) ICT Awards 2013 & 2017 and Silver award in APICTA (Asia Pacific ICT Alliance Awards) 2013 in category of R&D hosted by Hong Kong. He is also recipient of the prestigious Charles Wallace Fellowship 2016/17 as well as British Council Fellowship 2018, visiting research fellow at Centre of Decision Research, University of Leeds, UK and scientific director of Data and Text Mining Lab, Manchester Metropolitan. He is also an expert committee member of engineering & technology for the evaluation/recognition of national research journals for Higher Education Commission (HEC), Pakistan. Finally, he has the honour of being the finalist of the British Council's Professional Achievement Award 2016/17.

**Dr. Summair Raza** has completed his PhD from National University of Science and Technology (NUST) in 2018. Recently he is working as Assistant Professor in Computer Science Department in Virtual University of Pakistan. He has published various research articles in national/international level journals and conferences. His research interests include Feature Selection, Rough Set Theory, Software Architecture and Non-Functional requirements.