

# FUZZY ROUGH SETS BERDASARKAN METODE MAPREDUCE PADA BIG DATA

Mohamad Abduh<sup>1</sup>

Teknik Informatika, Sekolah Tinggi Teknologi Mandala Bandung

## Abstrak

*Fuzzy Rough sets* dapat menangani dataset dengan atribut diskret dan kontinu serta sudah banyak digunakan dalam teknik *Data Mining* diantaranya dalam proses pengenalan pola, klasifikasi gambar, seleksi atribut dan lain-lain, tetapi masih terbatas pada pengolahan dataset yang berskala kecil dan bagaimana apabila *Fuzzy Rough sets* digunakan untuk mengolah dataset yang berskala besar atau *Big Data* yang masih sedikit penerapannya. Salah satu cara untuk mengolah dataset yang berskala besar adalah dengan metode *MapReduce* pada aplikasi Hadoop atau Apache Spark dimana *MapReduce* bekerja dalam sistem terdistribusi atau *Cluster*. Tujuan dari penelitian ini dimaksudkan agar *Fuzzy Rough sets* dapat diterapkan pada pengolahan dataset yang berskala besar berdasarkan metode *MapReduce* dan dapat meningkatkan keakuratan dalam seleksi atribut dengan mereduksi atribut yang tidak dibutuhkan berdasarkan *Fuzzy Quick Reduct* yang sangat bermanfaat dalam *Feature Selection*, sehingga metode ini sangat dibutuhkan dalam pengolahan data yang sangat besar atau *Big Data*.

**Kata Kunci :** Fuzzy Rough Set, MapReduce, Big Data, Fuzzy Set.

## Abstract

*Fuzzy Rough sets* can handle datasets with discrete and continuous attributes and have been widely used in data mining techniques including in the process of pattern recognition, image classification, attribute selection and others, but they are still limited to processing small-scale datasets and what if *Fuzzy Rough sets* used to process large-scale datasets or *Big Data* which are still of little application. The way to process large-scale datasets is the *MapReduce* method in the Hadoop or Apache Spark application where *MapReduce* works in a cluster distributed system. The purpose of this research is that *Fuzzy Rough sets* can be applied to large-scale dataset processing based on the *MapReduce* method and can improve the accuracy in attribute selection by reducing dimensionality of attributes based on *Fuzzy Quick Reduct* which is very useful in *Feature Selection*, so this method is needed in processing very large dataset or *Big Data*.

**Keywords :** Fuzzy Rough Set, MapReduce, Big Data, Fuzzy Set.

## 1. PENDAHULUAN

*Fuzzy Rough sets* merupakan pengembangan dari metode *Fuzzy sets* [1] dan metode *Rough sets* [2], *fuzzy sets* lebih menitikberatkan pada *vagueness* sedangkan *Rough sets* lebih menitikberatkan pada *indiscernibility* dimana *Fuzzy Rough sets* dapat menangani dataset yang mempunyai atribut diskret maupun dataset dengan atribut kontinu serta dapat mendeteksi derajat keanggotaan dari setiap objek yang ekuivalen,

sehingga pada metode ini tidak diperlukan proses diskretisasi [7]. Penerapan *Fuzzy Rough sets* dalam teknik data mining masih terbatas pada dataset yang berskala kecil sedangkan untuk mengolah dataset yang berskala besar atau *Big Data* masih sedikit penerapannya. Salah satu cara untuk mengolah dataset yang berskala besar adalah dengan menggunakan metode *MapReduce* [3] yang sudah diimplementasikan pada aplikasi Hadoop dan Apache Spark [10].

Penerapan *Fuzzy Rough Set* dengan menggunakan pendekatan batas bawah atau *lower approximation* dan batas atas atau *upper approximation* serta dengan mengkonstruksi matrik *indiscernibility* dan matrik *similarity* telah berhasil diujicobakan pada mesin terdistribusi dengan menggunakan metode *Message Passing Interface (MPI)* [4], tetapi metode ini akan menghabiskan sumber daya memori *cluster* dan masih menggunakan *harddisk* untuk ruang penyimpanan pada setiap urutan prosesnya sehingga akan menghabiskan ruang penyimpanan *cluster*. Kemudian adanya dua program yaitu program *master* dan program *slave* sehingga akan menyulitkan dalam pengoperasiannya.

Selanjutnya penerapan *Fuzzy Rough sets* pada seleksi atribut dengan menggunakan pendekatan *k-Nearest Neighbor Classifier (KNN)* [5] yang di implementasikan pada sistem terdistribusi Hadoop dan Apache Spark dapat meningkatkan keakuratan dalam regresi, tetapi metode ini pun akan menghabiskan sumber daya memori komputer karena adanya proses looping yang terus menerus apabila penentuan *k* yang tidak tepat akan mengurangi keakuratan dalam prediksi.

Seleksi atribut dimaksudkan untuk mengatasi permasalahan dalam memilih atribut input suatu prediksi dalam mesin pembelajaran dan atribut yang tidak berpengaruh terhadap prediksi dapat diabaikan sehingga prediksi akan lebih akurat.

Seleksi atribut dengan menggunakan *Fuzzy Rough sets* sangat efektif untuk mengurangi dimensi dari dataset [6] sehingga proses pelatihan dan pengujian dataset.

## 1.1 Tinjauan Pustaka

### 1.1.1 Fuzzy Rough Set

Konsep *Fuzzy rough set* didasarkan pada kehidupan sehari-hari untuk menyatakan nilai atau ukuran dari suatu kondisi, selain dinyatakan dalam bentuk diskret misalnya: rendah, sedang dan tinggi, juga sering dinyatakan dalam bentuk numerik atau kontinu misalnya untuk menyatakan ukuran tinggi dari seseorang atau untuk menyatakan suhu air mendidih.

Sebenarnya dengan *Rough set* data dalam bentuk kontinu dapat diproses dengan menggunakan teknik diskretisasi tetapi proses diskretisasi akan menyebabkan beberapa informasi yang hilang.

Pada *Fuzzy rough set* setiap kelas yang ekuivalen dibagi secara tepat untuk memperluas partisi *Fuzzy* dengan mendefinisikan tingkat keanggotaan *Fuzzy* setiap objek untuk masing-masing kelas yang ekuivalen dan mempunyai tingkat keanggotaan antara 0 sampai 1 serta dapat mempertahankan informasi dalam bentuk kontinu tanpa perlu proses diskretisasi [9][11].

Pada *Rough set* misalkan pasien yang mempunyai gejala sakit Flu dapat dikategorikan sebagai orang yang benar-benar sakit Flu atau dapat pula dikategorikan sebagai orang yang tidak sakit Flu, artinya bahwa tingkat keanggotaan-nya adalah dalam bentuk biner "Ya" atau "Tidak" sedangkan pada *Fuzzy rough set* untuk menentukan apakah pasien dengan gejala sakit "Flu" benar-benar sakit atau tidak, dapat dikategorikan dengan tingkat keanggotaan [5].

### 1.1.2 Information System

*Information system* pada *Fuzzy rough set* merupakan sebuah ruang aproksimasi atau perkiraan dari himpunan data yang direpresentasikan dalam sebuah tabel di mana baris merepresentasikan objek dan kolom

merepresentasikan atribut dari objek, dan didefinisikan sebagai  $S = (X, A)$ , di mana  $X = \{x_1, \dots, x_n\}$  adalah himpunan semesta yang tidak kosong dan  $A = \{a_1, \dots, a_n\}$  adalah atribut dari objek-objek tersebut. Nilai dari atribut tersebut dapat di representasikan sebagai matrik  $Q = n \times m$  di mana  $q_{it} = (i \in \{1, \dots, n\} \text{ dan } t \in \{1, \dots, m\})$  [5].

Pada *Information system* juga terdapat hasil keluaran dari klasifikasi yang telah diketahui dan disebut dengan atribut keputusan atau *Decision system*, tabel 2.1 merupakan contoh dari *Information system* dengan atribut kondisi dan atribut keputusan berupa data numerik dan diskret serta atribut keputusannya terdapat pada kolom "d" dan didefinisikan sebagai  $S = (X, A \cup \{d\})$ .

### 1.1.3 Fuzzy Similarity

*Fuzzy Similarity* dalam *Fuzzy rough sets* direpresentasikan sebagai matrik  $R = n \times m$ , di mana  $r_{ij}$  merupakan tingkat keanggotaan dari *Fuzzy Similarity* antara objek  $x_i$  dan  $x_j$  ( $i, j \in \{1, \dots, n\}$ ). Nilai dari  $r_{ij}$  antara 0 sampai 1. *Fuzzy Similarity* dari matrik  $R$  dapat diturunkan dari *Information system*  $S = (X, A)$ .

*Fuzzy Similarity* dari setiap atribut direpresentasikan menggunakan fungsi *manhattan* dengan menggunakan formula:

$$R(x, y) = 1 - \frac{|a(x) - a(y)|}{a_{\max} - a_{\min}} \quad (2)$$

*Fuzzy Similarity* akan menentukan sejauh mana relasi dari dua elemen dalam himpunan U.

### 1.1.4 Fuzzy Indiscernibility Relation

*Fuzzy Indiscernibility Relation* dalam *Fuzzy Rough Sets* direpresentasikan sebagai matrik  $R = n \times m$ , di mana  $r_{ij}$  merupakan tingkat

keanggotaan dari *Indiscernibility relation* antara objek  $x_i$  dan  $x_j$  ( $i, j \in \{1, \dots, n\}$ ). Nilai dari  $r_{ij}$  antara 0 sampai 1, apabila nilai  $x_i$  dan  $x_j$  sama dengan 0 maka  $x_i$  dan  $x_j$  tidak mempunyai hubungan yang erat atau *Dissimilar* dan apabila  $x_i$  dan  $x_j$  bernilai 1 maka dipastikan mempunyai hubungan yang tidak dapat dipisahkan atau *Indiscernibility relation*. *Indiscernibility relation* dari matrik  $R$  dapat diturunkan dari *Information system*  $S = (X, A)$  yang didefinisikan dari fungsi *t-norm Lukasiewicz* [6].

### 1.1.5 Fuzzy Lower dan Fuzzy Upper Approximation

*Fuzzy Lower dan Fuzzy Upper Approximation* adalah untuk mendefinisikan sejauh mana set dari setiap elemen dapat diklasifikasikan ke dalam kelas tertentu apakah kuat atau lemah dan direpresentasikan dari fungsi *implicator I* dan *t-norm T*, untuk *Fuzzy Lower Approximation* menggunakan fungsi *implicator I*, sedangkan untuk *Fuzzy Upper Approximation* menggunakan fungsi *t-norm T* [7].

*Fuzzy Lower Approximation* menggunakan formula:

$$(RB \downarrow A)(y) = \inf_{x \in U} (RB(x, y), A(x)) \quad (4)$$

Sedangkan *Fuzzy Upper Approximation* menggunakan formula:

$$(RB \uparrow A)(y) = \sup_{x \in U} T(RB(x, y), A(x)) \quad (5)$$

### 1.1.6 Fuzzy Quick Reduct

*Fuzzy Quick Reduct* diimplementasikan pada *feature selection* yaitu untuk mereduksi atribut dan hanya memilih atribut yang mempunyai tingkat ketergantungan *degree dependency* untuk semua himpunan pada set data dan menggunakan pendekatan fungsi

Fuzzy Positive Region [6] sebagai berikut:

$$\mu_{POS_{RP}}(Q)(x) = \sup_{K \in U/Q} \mu_{Rp} \chi(x)$$

dan *degree dependency* direpresentasikan dengan fungsi berikut:

$$\gamma_p(Q) = \frac{\sum_{x \in U} \mu_{POS_{RP}}(Q)(x)}{|U|}$$

### 1.1.7 MapReduce

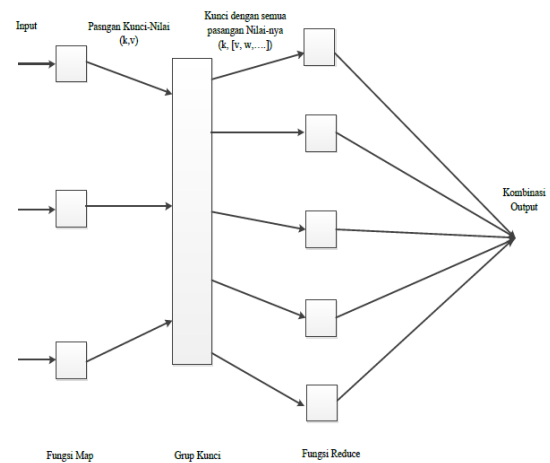
*MapReduce* merupakan sebuah model pemrograman atau algoritma yang diimplementasikan untuk mengolah set data yang berskala besar dalam sistem terdistribusi atau *Cluster*.

Model pemrograman pada *MapReduce* dengan cara mengotomatisasi proses paralel dan distribusi dalam menangani komputasi yang sangat besar dan telah diterapkan oleh Google untuk *crawled documents*, *web request logs* dan lain-lain [3].

Pada *MapReduce* terdapat dua tahapan proses atau fungsi yaitu fungsi *Map* dan fungsi *Reduce* dengan tahapan sebagai berikut:

- Fungsi *Map* merupakan sebuah fungsi untuk membagi *dataset* menjadi bagian-bagian kecil yang merupakan sebuah pasangan kunci dan nilai atau *keys values* serta mendistribusikan pasangan kunci dan nilai tersebut ke semua titik atau *Node* dalam komputasi terdistribusi .
- Diantara fungsi *Map* dan fungsi *Reduce* terdapat tahapan antara atau *Intermediate task* di mana pasangan kunci dan nilai atau *keys values* dari setiap fungsi *Map* dikumpulkan dan diurutkan berdasarkan kuncinya. Kunci-kunci tersebut kemudian akan menjadi input pada fungsi *Reduce*.
- Fungsi *Reduce* merupakan sebuah fungsi untuk menggabungkan semua nilai yang mempunyai kunci yang sama [6].

Tahapan komputasi *MapReduce* seperti pada gambar 1.1 berikut:



Gambar 1.1 Tahapan Komputasi MapReduce

## 2. METODOLOGI PENELITIAN

Penelitian ini menggunakan 3 buah server HP Proliant G10 dengan masing-masing memory 32GB dan Harddisk 600GB, sistem operasi menggunakan Linux Ubuntu Server versi 18.04 dan satu *master* dan dua *worker* serta *Framework Apache Spark* yang didalamnya sudah terdapat algoritma *MapReduce*, beberapa tahapan yang akan dijelaskan sebagai berikut:

### 2.1 Data

Data untuk penelitian ini menggunakan data baseline indikator kumuh yang terdiri dari 5154 objek atau baris dengan 69 atribut kondisi dan 1 atribut keputusan serta data ini mempunyai atribut diskret dan atribut kontinyu sehingga sangat cocok untuk digunakan sebagai data pada penelitian ini karena tidak diperlukan proses diskretisasi.

### 2.2 Preprocessing

Setelah mendapatkan data yang diperlukan, dilakukan pembersihan data, jika terdapat data yang tidak lengkap, ganda dan sebagainya.

### 2.3 Tabel Keputusan

Selanjutnya data yang telah melalui tahapan preprocessing atau pembersihan maka akan dilakukan transformasi data kedalam bentuk tabel keputusan *Information System* dari *Fuzzy Rough sets* dan dipilih atribut-atribut kondisi dan atribut keputusan dengan mengikuti formula:

$$A = (U, A \cup \{d\}) \quad (1)$$

Dimana  $U$  adalah *universe* atau himpunan,  $A$  adalah atribut dari objek dan  $d$  adalah atribut keputusan. Tabel keputusan *Information System* ditunjukkan pada table 3.1 berikut:

Tabel 3.1 Tabel Keputusan

| Jumlah Keteraturan Bangunan Hunian | Luas Permukiman | Jumlah Total Bangunan | Kumuh    |
|------------------------------------|-----------------|-----------------------|----------|
| <b>P01</b>                         | <b>P03</b>      | <b>P04</b>            | <b>d</b> |
| 106                                | 19.23           | 117                   | 0        |
| 74                                 | 13.58           | 101                   | 0        |
| 103                                | 26.56           | 146                   | 0        |
| 51                                 | 10.5            | 177                   | 1        |
| 26                                 | 16.8            | 182                   | 1        |
| 30                                 | 23.4            | 152                   | 1        |
| 0                                  | 14.96           | 99                    | 0        |

### 2.4 Fuzzy Similarity

*Fuzzy Similarity Relation* atau *Fuzzy Equivalence Relation* akan mendefinisikan matrik *Fuzzy set* yang ekuivalen dari setiap atribut. Data yang telah ditransformasikan kemudian ditentukan varian dan range dari setiap atribut, kemudian dilakukan penghitungan. Hasil dari proses *Fuzzy Similarity* ditunjukkan pada tabel 3.2 berikut:

Tabel 3.2 Matriks Fuzzy Similarity Atribut P04

|              |              |              |              |              |              |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1.00<br>0000 | 0.80<br>7229 | 0.65<br>0602 | 0.27<br>7108 | 0.21<br>6867 | 0.57<br>8313 | 0.78<br>3133 |
| 0.80<br>7229 | 1.00<br>0000 | 0.45<br>7831 | 0.08<br>4337 | 0.02<br>4096 | 0.38<br>5542 | 0.97<br>5904 |
| 0.65<br>0602 | 0.45<br>7831 | 1.00<br>0000 | 0.62<br>6506 | 0.56<br>6265 | 0.92<br>7711 | 0.43<br>3735 |
| 0.27<br>7108 | 0.08<br>4337 | 0.62<br>6506 | 1.00<br>0000 | 0.93<br>9759 | 0.69<br>8795 | 0.06<br>0241 |

|              |              |              |              |              |              |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 0.21<br>6867 | 0.02<br>4096 | 0.56<br>6265 | 0.93<br>9759 | 1.00<br>0000 | 0.63<br>8554 | 0.00<br>0000 |
| 0.57<br>8313 | 0.38<br>5542 | 0.92<br>7711 | 0.69<br>8795 | 0.63<br>8554 | 1.00<br>0000 | 0.36<br>1446 |
| 0.78<br>3133 | 0.97<br>5904 | 0.43<br>3735 | 0.06<br>0241 | 0.00<br>0000 | 0.36<br>1446 | 1.00<br>0000 |

### 2.5 Fuzzy Indiscernibility Relation

Setelah ditentukan *Fuzzy Similarity* kemudian tahap selanjutnya adalah menentukan *Fuzzy Indiscernibility Relation* dengan menggunakan formula *t-norm lukasiwicz* yaitu menentukan nilai maksimum antara hasil penjumlahan setiap indeks pada matrik *Fuzzy Similarity*:

$$\max(x_2 + x_1 - 1, 0) \quad (3)$$

*Fuzzy Indiscernibility Relation* digunakan pada setiap hubungan *fuzzy* untuk menentukan sejauh mana dua objek tidak dapat dibedakan *indiscernible* yang hasilnya ditunjukkan pada table 3.3 berikut:

Tabel 3.3 Fuzzy Indiscernibility Relation Atribut Keputusan (d)

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

### 2.6 Fuzzy Lower dan Upper Approximations

Setelah diketahui *Fuzzy Similarity Relation* selanjutnya bisa ditentukan *Fuzzy Lower* dan *Upper Approximations* dari *Fuzzy Rough sets* untuk menentukan sejauh mana set dari elemen dapat diklasifikasikan ke dalam kelas tertentu apakah kuat atau lemah yang ditunjukkan pada table 3.4 berikut:

Tabel 3.4 Fuzzy Lower Approximation (P04 dan d)

|              |              |              |              |              |              |              |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 0.42<br>1687 | 0.61<br>4458 | 0.07<br>2289 | 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.63<br>8554 |
| 0.42<br>1687 | 0.61<br>4458 | 0.07<br>2289 | 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.63<br>8554 |
| 0.42<br>1687 | 0.61<br>4458 | 0.07<br>2289 | 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.63<br>8554 |
| 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.37<br>3494 | 0.43<br>3735 | 0.07<br>2289 | 0.00<br>0000 |
| 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.37<br>3494 | 0.43<br>3735 | 0.07<br>2289 | 0.00<br>0000 |
| 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.37<br>3494 | 0.43<br>3735 | 0.07<br>2289 | 0.00<br>0000 |
| 0.42<br>1687 | 0.61<br>4458 | 0.07<br>2289 | 0.00<br>0000 | 0.00<br>0000 | 0.00<br>0000 | 0.63<br>8554 |

## 2.7 Knowledge

Sebagaimana telah dibahas pada bab sebelumnya maka pengujian pada tahap *Knowledge* menggunakan pendekatan *Fuzzy Quick Reduct* [6] yang hasilnya ditunjukkan pada tabel 3.5 dan *rule base* pada tabel 3.6.

Tabel 3.5 Degree Dependency

| P01         | P03         | P04         | P01,P04       | P03,P04      |
|-------------|-------------|-------------|---------------|--------------|
| 0.3167<br>1 | 0.1653<br>6 | 0.3752<br>2 | 0.345963<br>4 | 0.27028<br>9 |

Tabel 3.6 Rule Base

| No. | Generate Rules {P01, P04}             |
|-----|---------------------------------------|
| 1   | IF P01 = 106 and P04 = 117 THEN d = 0 |
| 2   | IF P01 = 74 and P04 = 101 THEN d = 0  |
| 3   | IF P01 = 103 and P04 = 146 THEN d = 0 |
| 4   | IF P01 = 51 and P04 = 177 THEN d = 1  |
| 5   | IF P01 = 26 and P04 = 182 THEN d = 1  |
| 6   | IF P01 = 30 and P04 = 152 THEN d = 1  |
| 7   | IF P01 = 0 and P04 = 99 THEN d = 0    |

## 3. HASIL PENELITIAN DAN PEMBAHASAN

Data yang digunakan pada penelitian ini untuk sistem informasi terdiri dari 5154 objek atau baris, 69 atribut kondisional dan 1 atribut keputusan yang berbentuk kontinyu.

Pengujian ini diimplementasikan menggunakan *Rough Sets R package* [8] pada perangkat lunak RStudio 1.1 dengan hasil sebagai berikut:

- Arsitektur dari algoritma *Fuzzy Rough Sets* dapat digunakan untuk mengolah set data yang berskala besar karena output atau keluaran yang dihasilkan sesuai dengan kaidah-kaidah *Fuzzy Rough Sets* yaitu matrik *Fuzzy Similarity*, matrik *Fuzzy Indiscernibility Relation*, matrik *Fuzzy Lower* dan *Fuzzy Upper Approximation* serta *Knowledge* yang berupa *Rule Base*.
- Algoritma *Fuzzy Rough Sets* dengan *Fuzzy Quick Reduct* merupakan algoritma yang dapat digunakan dalam mereduksi atribut-atribut yang tidak diperlukan terutama dalam *Feature Selection*.

## 4. KESIMPULAN

Berdasarkan hasil penelitian dapat diketahui bahwa *Fuzzy Rough sets* bisa diimplementasikan untuk pengolahan data yang berskala besar atau *Big Data* menggunakan metode *MapReduce* pada *framework Apache Spark* dengan pendekatan *Fuzzy Similarity* dan *Fuzzy Indiscernibility* serta *Fuzzy Quick Reduct*.

## DAFTAR PUSTAKA

- [1] L.A.Zadeh, 1965, Fuzzy sets. doi: [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
- [2] Z, Pawlak, 1982, Rough sets, International Journal of Computer Information Science, doi: <https://doi.org/10.1007/BF01001956>.
- [3] Jeffrey Dean and Sanjay Ghemawat, Google Inc 2004, MapReduce: Simplified Data Processing on Large Cluster. doi: <http://dx.doi.org/10.1145/1327452.1327492>.
- [4] Hasan Asfoor, Chris Cornelis, Rajagopalan Srinivasan, Gayatri Vasudevan, Nele Verbiest, Matthew Tolentino 2014,

- Computing Fuzzy Rough Approximations in Large Scala Information Systems. doi: <https://doi.org/10.1109/BigData.2014.7004350>.
- [5] Sarah Vluymans, Hasan Asfoor and Chris Cornelis, Yvan Saeys, Matthew Talentino, Martine De Cock, 2014, Distributed Fuzzy Rough Prototype Selection for Big Data Regression. doi: <https://doi.org/10.1109/NAFIPS-WConSC.2015.7284158>.
- [6] Richard Jensen and Qiang Shen, 2008, New Approaches to Fuzzy Rough Feature Selection. doi: <https://doi.org/10.1109/TFUZZ.2008.924209>.
- [7] Anna Maria Radzikowska and Etienne E. Kerre, 2002, A comparative study of fuzzy rough sets.
- [8] Lala Septem Riza, Andrzej Janusz, Christoph Bergmeir, Chris Cornelis, Francisco Herrera, Dominik Slezak and Jose Manuel Benitez, 2014, Implementing algorithm of rough set theory and fuzzy rough set theory in R package "RoughSets".
- [9] Didier Dubois and Henri Prade, 2014, Rough Fuzzy Sets and Fuzzy Rough Sets, International Journal of General System, doi: <http://dx.doi.org/10.1080/03081079008935107>.
- [10] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker and Ion Stoica, 2015, Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing.
- [11] F.F. Xu, D.Q. Miao, L. Wei, 2008, Fuzzy-rough attribute reduction via mutual information with an application to cancer classification.