

Rekomendasi Pembimbing dan Penguji Tugas Akhir Menggunakan *Vector Space Model*

Eka Pitriyani¹, Ardiansyah²

¹Kelompok Keilmuan Rekayasa Perangkat Lunak dan Data
Program Studi Teknik Informatika, Universitas Ahmad Dahlan Yogyakarta

¹eka1500018242@webmail.uad.ac.id, ²ardiansyah@tif.uad.ac.id

ABSTRAK

Penentuan pembimbing dan penguji skripsi di perguruan tinggi merupakan hal yang sangat penting. Penentuan pembimbing dan penguji skripsi yang dilakukan secara tidak objektif berpotensi mengakibatkan hasil penentuan yang kurang tepat, bias, dan belum tentu sesuai antara bidang keahlian calon dosen pembimbing atau penguji dengan topik skripsi mahasiswa. Penelitian terdahulu telah berusaha untuk memberikan rekomendasi penguji skripsi dengan menggunakan *Vector Space Model*. Akan tetapi, hasil yang diperoleh bukan berupa ranking, dan yang direkomendasikan hanya penguji saja. Dengan adanya ranking maka dapat diketahui relevan atau tidaknya rekomendasi yang dihasilkan. Penelitian ini bertujuan untuk menerapkan metode *Vector Space Model* dengan hasil rekomendasi pembimbing dan penguji tugas akhir sekaligus dalam bentuk ranking. Hasil penelitian menunjukkan bahwa sistem rekomendasi dosen pembimbing dan penguji skripsi mampu berfungsi untuk memberikan rekomendasi yang sesuai dengan *query* secara lebih tepat. Hasil evaluasi *Mean Average Precision* menggunakan 143 *query* dan 1257 dokumen menunjukkan nilai sebesar 0.994. Nilai evaluasi *Mean Average Precision* tersebut dinyatakan sangat baik atau masuk pada klasifikasi *excellent recommendation* berdasarkan *diagnostic test*. Hasil penelitian ini selanjutnya bisa digunakan oleh mahasiswa dan koordinator skripsi untuk menentukan dosen pembimbing maupun penguji skripsi secara otomatis dengan hasil yang objektif dan lebih tepat.

Kata kunci: Pembimbing, Penguji, Rekomendasi, Skripsi, *Vector Space Model*

ABSTRACT

Thesis supervisor and examiner assignment in higher education is very important. An unobjective assignment can lead to unprecise and bias results. Moreover, the assigned supervisor and examiner are not matching with their expertise. Researchers have been employed vector space model to provide the objective thesis supervisor and examiner assignment. Unfortunately, the results were not formed as a rank and recommendation only for the thesis examiner. The ranking is an important form of recommendation system because will make it easy to evaluate the relevance. This study aims to use the vector space model to generate the ranking of the thesis supervisor and examiner assignment. The result shows that the function of the system performs well to generate a recommendation. The evaluation using mean average precision (MAP) with 143 queries and 1257 documents. From the evaluation, the MAP yielded a score of 0.994 and classified as an excellent recommendation based on diagnostic test. In the future, our results may be used by the thesis coordinator and students to choose and assign thesis supervisor and examiner automatically, properly and objectively.

Keyword: Supervisor, Examiner, Recommendation, Thesis, Vector Space Model

1. PENDAHULUAN (10 PT)

Dosen merupakan pendidik profesional dan ilmuwan dengan tugas utama mentransformasikan, mengembangkan, dan menyebarkan ilmu pengetahuan, teknologi, dan seni melalui pendidikan, penelitian, dan pengabdian kepada masyarakat [1]. Dalam melaksanakan kegiatan akademik dosen memiliki tanggung jawab dan wewenang salah satunya membimbing dalam menghasilkan skripsi. Skripsi adalah sebuah mata kuliah yang harus ditempuh oleh seorang mahasiswa menjelang akhir studinya [2]. Dalam proses pengerjaan skripsi, mahasiswa membutuhkan dosen pembimbing sebagai tempat untuk berkonsultasi. Dosen pembimbing mempunyai peran sangat penting dalam proses penyelesaian penulisan skripsi. Dosen pembimbing memiliki tanggung jawab terhadap mahasiswa agar mampu menyusun skripsi dengan baik dan berkualitas. Dosen pembimbing yang sesuai dan berkompeten akan mampu mengarahkan dan menentukan

keberhasilan skripsi mahasiswa yang dibimbing. Selain dosen pembimbing mahasiswa juga membutuhkan dosen penguji yang akan menguji laporan skripsinya kelak.

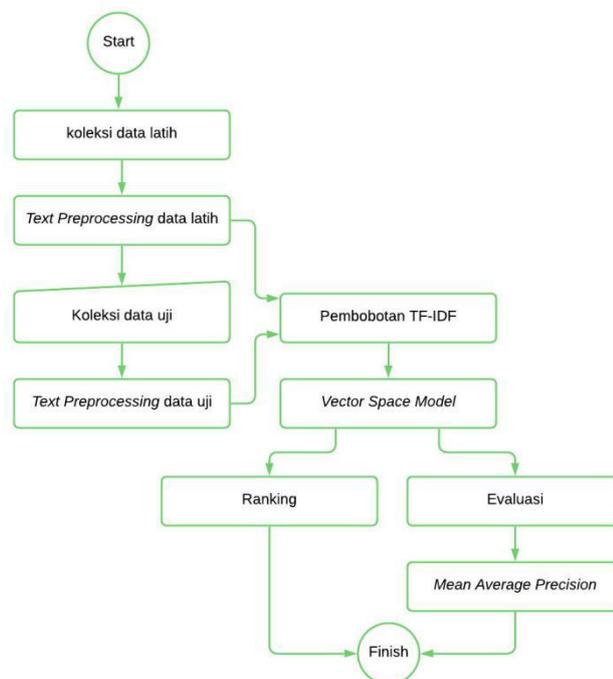
Hasil wawancara dengan Koordinator Skripsi Teknik Informatika Universitas Ahmad Dahlan bahwa penentuan dosen pembimbing skripsi saat ini dilakukan dengan cara mahasiswa menentukan secara langsung calon dosen pembimbing skripsi atau dosen sendiri yang mencari mahasiswa untuk mengerjakan skripsi sesuai topik yang ditawarkan oleh dosen. Berbeda dengan penentuan dosen pembimbing skripsi, penentuan dosen penguji dilakukan oleh Koordinator Skripsi dengan didata pada aplikasi Microsoft Excel. Kelemahan mendata secara manual untuk menentukan dosen penguji skripsi yang sesuai antara judul skripsi dengan konsentrasi dosen penguji membutuhkan waktu proses. Sedangkan kelemahan penentuan dosen pembimbing skripsi yang dilakukan oleh mahasiswa yaitu kesesuaian antara bidang keahlian calon dosen pembimbing dengan bidang minat mahasiswa yang bisa tidak sesuai dan kuota bimbingan calon dosen pembimbing yang sudah penuh.

Kecerdasan buatan, *machine learning*, dan data mining saat ini memungkinkan untuk dapat diterapkan dalam perekomendasi pembimbing dan penguji skripsi. Dari beberapa metode yang dapat diterapkan, *Vector Space Model* memiliki kelebihan yaitu tingkat akurasi dan tingkat keberhasilan data uji yang tinggi seperti yang dilakukan oleh Ruli, Siregar, Sinaga, dan Arianto [3]. Namun, pada penelitian tersebut tidak menghasilkan rekomendasi dosen penguji skripsi dalam bentuk ranking. Padahal dengan adanya ranking maka akan mudah terlihat dosen pembimbing ataupun dosen penguji mana yang relevan atau yang paling diprioritaskan.

Penelitian ini akan menerapkan metode *Vector Space Model* untuk menghasilkan rekomendasi dosen pembimbing dan penguji tugas akhir dalam bentuk ranking sehingga diharapkan pemilihan dosen pembimbing maupun penguji skripsi akan lebih tepat dan sesuai dengan kompetensi masing-masing.

2. METODE PENELITIAN (10 PT)

Objek pada penelitian ini yaitu perekomendasi dosen pembimbing dan penguji tugas akhir menggunakan *Vector Space Model*. Data yang digunakan dalam penelitian ini adalah data skripsi mahasiswa Teknik Informatika Universitas Ahmad Dahlan Yogyakarta dari semester Ganjil 2008/2009 sampai Ganjil 2018/2019. Dataset berupa *file excel* yang menampung 1257 data dan terdiri dari empat kolom yaitu judul, pembimbing, penguji pertama, dan penguji kedua. Data diperoleh dari Koordinator Skripsi Program Studi Teknik Informatika Universitas Ahmad Dahlan Yogyakarta. Data latihan berupa judul, pembimbing, penguji pertama, dan penguji kedua. Sedangkan data uji merupakan judul yang akan dicari rekomendasi dosen pembimbing ataupun pengujinya. Tahapan yang dilakukan pada sistem ditunjukkan pada Gambar 1 dan dijelaskan pada bagian 2.1 sampai 2.4.



Gambar 1. Tahapan sistem

2.1. Text Preprocessing

Text pada umumnya memiliki struktur kalimat yang tidak baik dan banyak noise. Untuk mendapatkan ekstraksi fitur pada text maka harus dilakukan text preprocessing. Beberapa proses yang dapat dilakukan pada text preprocessing adalah case folding, tokenizing, filtering, dan stemming.

- Case folding adalah proses mengubah semua huruf dalam dokumen menjadi huruf kecil [4].
- Tokenizing merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan memungkinkan pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca [5].
- Filtering (stopword removal) adalah tahap mengambil kata-kata penting dari hasil tokenizing dengan menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist atau stopwords adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words [4].
- Stemming digunakan untuk mengubah term menjadi kata dasar. Proses stemming dilakukan dengan cara menghilangkan semua imbuhan (affixes) baik yang terdiri dari awalan (prefixes), sisipan (infixes), akhiran (suffixes) dan confixes (kombinasi dari awalan dan akhiran pada kata turunan) [6].

2.1. Pembobotan TF-IDF

Term Frequency (TF) merupakan pembobotan yang sederhana dimana penting tidaknya sebuah kata diasumsikan sebanding dengan jumlah kemunculan kata tersebut dalam dokumen, sementara *Inverse Document Frequency* (IDF) merupakan pembobotan yang mengukur seberapa penting sebuah kata dalam dokumen apabila dilihat secara global pada seluruh dokumen [7]. Menghitung *Term Frequency* (TF) menggunakan persamaan 1.

$$TF = tf_{ij} \quad (1)$$

keterangan:

TF = term frequency

tf_{ij} = banyaknya kemunculan *term* dalam dokumen

menghitung *Inverse Document Frequency* (IDF) menggunakan persamaan 2

$$IDF = \log\left(\frac{N}{df_i}\right) + 1 \quad (2)$$

keterangan:

IDF = inverse document frequency

N = jumlah dokumen dalam koleksi

df_i = banyaknya dokumen dalam koleksi dimana *term* (t_i) muncul didalamnya

menghitung TF-IDF menggunakan persamaan 3

$$W = TF \cdot IDF \quad (3)$$

bobot dokumen (W) dihitung untuk mengetahui bobot suatu *term* yang terdapat pada suatu dokumen dari hasil perkalian TF dengan IDF .

2.2. Vector Space Model

Vector Space Model (VSM) adalah metode untuk melihat tingkat kedekatan atau kesamaan (*similarity*) [8]. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas diantara vektor dokumen dan vektor *query* [9]. Maning, Raghavan, dan Schutze [10] menyatakan bahwa kita dapat menggunakan *cosine similarity* antara vektor *query* dengan vektor dokumen sebagai ukuran skor dokumen untuk *query* tersebut. Nilai yang dihasilkan kemudian dapat digunakan untuk memilih penilaian atas dokumen untuk *query*. Proses perhitungan *cosine similarity* menggunakan persamaan 4.

$$Cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (4)$$

keterangan:

$\|x\|$ = panjang vektor x

$\|y\|$ = panjang vektor y

2.3. Evaluasi *Mean Average Precision*

Untuk mengetahui seberapa relevan dokumen yang di rekomendasikan maka terdapat beberapa hal yang harus dilakukan. Tahapan yang dilakukan adalah sebagai berikut:

a. Menghitung *precision*

Precision merupakan dokumen relevan yang diambil terhadap dokumen yang dikembalikan oleh sistem. *Precision* menunjukkan rasio antara dokumen relevan yang dikembalikan sistem terhadap dokumen yang diberikan sistem. Perhitungan *precision* menggunakan persamaan 5.

$$Precision = \frac{TP}{TP+FP} = P(\text{relevant}|\text{terambil}) \quad (5)$$

keterangan:

TP = dokumen relevan yang dikembalikan sistem

FP = dokumen tidak relevan yang dikembalikan sistem

b. Menghitung *average precision*

Average precision yaitu mencari nilai rata-rata dari jumlah *precision* relevan. *Average precision* hanya mengambil nilai presisi dari dokumen-dokumen yang relevan dan kemudian hasilnya dibagi dengan jumlah dokumen yang dilibatkan [11]. Menghitung *average precision* menggunakan persamaan 6.

$$AP = \frac{1}{|R|} \sum_{i=1}^n Precision(i) \cdot relevant(i) \quad (6)$$

keterangan:

|R| = total dokumen yang relevan

Precision(i) = precision dokumen ke-i

relevant(i) = >0 jika relevan, 0 jika tidak relevan

c. Menghitung *mean average precision*

Hasil *average precision* selanjutnya akan digunakan untuk mencari *mean average precision* (MAP) yaitu nilai rata-rata dari *average precision*. *mean average precision* telah terbukti memiliki penilaian yang sangat baik dan stabil [10]. Menghitung *mean average precision* dapat menggunakan persamaan 7.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad (7)$$

keterangan:

R_{jk} = himpunan hasil peringkat tertinggi

Q = jumlah *query*

Menurut Gorunescu [12] keakuratan *diagnostic test* terbagi menjadi lima klasifikasi seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Interpretasi *diagnostic test*

Nilai <i>Mean Average Precision</i>	Interpretasi <i>Diagnostic Test</i>
0.50-0.60	<i>Failure</i>
0.60-0.70	<i>Poor recommendation</i>
0.70-0.80	<i>Fair recommendation</i>
0.80-0.90	<i>Good recommendation</i>
0.90-1.00	<i>Excellent recommendation</i>

3. HASIL DAN ANALISIS

3.1. Hasil Implementasi

Data latih dan data uji sebelum dihitung kemiripannya harus melalui tahap *text preprocessing*, yang terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. Adapun tahap *text preprocessing* dapat dilihat pada Tabel 2.

Tabel 2. Tahap *text preprocessing*

Dokumen	Case Folding	Tokenizing	Filtering	Stemming
Sistem Pendukung Keputusan untuk Seleksi Mahasiswa Baru dengan <i>Cosine Similarity</i>	sistem pendukung keputusan untuk seleksi mahasiswa baru dengan <i>cosine similarity</i>	sistem	sistem	sistem
		pendukung	pendukung	dukung
		keputusan	keputusan	putus
		untuk	-	-
		seleksi	seleksi	seleksi
		mahasiswa	mahasiswa	mahasiswa
		baru	baru	baru
		dengan	-	-
		cosine	cosine	cosine
		similarity	similarity	similar

Proses selanjutnya yang harus dilakukan setelah *text preprocessing* yaitu pembobotan *Term Frequency - Inverse Document Frequency* (TF-IDF). Perhitungan dilakukan dengan mencari bobot setiap *term* pada setiap dokumen. *Term* merupakan kata dasar yang diperoleh dari *text preprocessing*. Tahap pembobotan TF-IDF ditunjukkan pada Tabel 3.

Tabel 3. Tahap Pembobotan TF-IDF

Term	TF					DF	IDF	W = TF*IDF				
	Q	D1	D2	D3	D4			Q	D1	D2	D3	D4
basis	0	0	0	1	1	2	1.301	0	0	0	1.301	1.301
web	0	0	0	1	1	2	1.301	0	0	0	1.301	1.301
mining	0	1	1	0	0	2	1.301	0	1.301	1.301	0	0
metode	0	1	1	1	1	4	1	0	1	1	1	1
data	0	1	1	0	0	2	1.301	0	1.301	1.301	0	0
cosine	1	1	0	0	0	1	1.602	1.602	1.602	0	0	0
similar	1	1	0	0	0	1	1.602	1.602	1.602	0	0	0
mahasiswa	1	0	1	0	0	1	1.602	1.602	0	1.602	0	0
sistem	1	0	0	1	1	2	1.301	1.301	0	0	1.301	1.301
dukung	1	0	0	0	1	1	1.602	1.602	0	0	0	1.602
putus	1	0	0	0	1	1	1.602	1.602	0	0	0	1.602
seleksi	1	0	0	0	1	1	1.602	1.602	0	0	0	1.602
baru	1	0	0	0	1	1	1.602	1.602	0	0	0	1.602

Setelah diperoleh hasil pembobotan *Term Frequency - Inverse Document Frequency* (TF-IDF) terhadap dokumen, yang akan dimanfaatkan dalam perhitungan *vector space model* adalah bobot *term* atau bobot kata pada dokumen. Berikut tahap *vector space model* yang ditunjukkan pada Tabel 4.

Tabel 4. Tahap *vector space model*

Term	W(Q) * W(D)				W ²				
	D1	D2	D3	D4	Q	D1	D2	D3	D4
basis	0	0	0	0	0	0	0	1.693	1.693
web	0	0	0	0	0	0	0	1.693	1.693
mining	0	0	0	0	0	1.693	1.693	0	0
metode	0	0	0	0	0	1	1	1	1
data	0	0	0	0	0	1.693	1.693	0	0
cosine	2.567	0	0	0	2.567	2.567	0	0	0
similar	2.567	0	0	0	2.567	2.567	0	0	0
mahasiswa	0	2.567	0	0	2.567	0	2.567	0	0
sistem	0	0	1.693	1.693	1.693	0	0	1.693	1.693
dukung	0	0	0	2.567	2.567	0	0	0	2.567
putus	0	0	0	2.567	2.567	0	0	0	2.567
seleksi	0	0	0	2.567	2.567	0	0	0	2.567
baru	0	0	0	2.567	2.567	0	0	0	2.567
Total	5.133	2.567	1.693	11.959	19.659	21.478	32.618	36.003	29.177
Panjang vektor					4.434	4.634	5.711	6.000	5.402

Langkah terakhir setelah mendapatkan hasil panjang vektor dari dokumen dan *query* yaitu menghitung nilai *cosine similarity*. Hasil perhitungan *cosine similarity* akan menentukan nilai kemiripan antara *query* yang dimasukkan dengan dokumen yang tersedia, sehingga dapat diketahui ranking dari tiap dokumen. Tahap *cosine similarity* ditunjukkan pada Tabel 5.

Tabel 5. Tahap *cosine similarity*

Kemiripan	Nilai <i>Cosine Similarity</i>	Ranking
Cos(D1,Q)	$5.133/(4.634*4.434) = 0.249$	2
Cos(D2,Q)	$2.567/(5.711*4.434) = 0.101$	3
Cos(D3,Q)	$1.693/(6.000*4.434) = 0.063$	4
Cos(D4,Q)	$11.959/(5.402*4.434) = 0.499$	1

3.2. Hasil Evaluasi

Evaluasi dilakukan menggunakan teknik evaluasi *information retrieval* untuk yang menghasilkan ranking [10]. Pengukuran dilakukan dengan pengujian kepresisian hasil pengembalian rekomendasi. Rekomendasi yang diberikan akan diberi label sesuai kriteria. Kriteria dan label ditentukan dari nilai *cosine similarity* seperti yang ditunjukkan pada Tabel 6.

Tabel 6. Ketentuan kriteria dan label

Nilai <i>Cosine Similarity</i>	<0	0-0.2	0.2-0.5	0.5-0.7	0.7-1.0
Kriteria	Tidak relevan	Kurang relevan	Cukup relevan	Relevan	Sangat relevan
Label	0	1	2	3	4

Untuk mencari skor *mean average precision* (MAP) digunakan dokumen yang cukup relevan, relevan, dan sangat relevan. Sedangkan dokumen kurang relevan dan tidak relevan tidak diikutsertakan untuk perhitungan MAP. Pengujian dilakukan menggunakan 143 *query* dan 1257 dokumen pada dataset. Hasil perhitungan nilai *mean average precision* yang diperoleh pada penelitian ini adalah 0.994 atau 99%. Berdasarkan klasifikasi keakuratan pengujian *diagnostic test* maka skor 0.9 sampai 1.00 masuk pada klasifikasi *excellent recommendation* atau dianggap sebagai rekomendasi yang sangat baik. Dengan hasil nilai *mean average precision* 0.994 maka hal ini menunjukkan bahwa rekomendasi yang diberikan pada sistem masuk pada klasifikasi sangat baik.

4. KESIMPULAN

- a. Telah dibuat sistem rekomendasi dosen pembimbing dan penguji tugas akhir yang dipilih secara otomatis sehingga memudahkan koordinator skripsi dalam mengatur pelaksanaan ujian pendadaran.
- b. Evaluasi sistem rekomendasi dosen pembimbing dan penguji menggunakan metode mean average precision sebesar 0.994. Evaluasi melibatkan 143 query dan 1257 dokumen. Nilai mean average precision tersebut dinyatakan excellent recommendation atau sangat baik berdasarkan klasifikasi keakuratan pengujian diagnostic test.

UCAPAN TERIMAKASIH

Alhamdulillah dengan rahmat Allah SWT yang telah memberikan nikmat kesehatan, kekuatan, kemudahan, dan kelancaran. Tak lupa sholawat serta salam bagi junjunganku, Nabi Muhammad SAW atas teladannya. Ucapan terimakasih kepada Bapak Ardiansyah yang telah membimbing, memberikan arahan dan motivasi. Semoga Allah membalas kebaikan yang telah diberikan.

DAFTAR PUSTAKA

- [1] Kemendikbud and K. Badan Kepegawaian Negara, *Peraturan Bersama Menteri Pendidikan dan Kebudayaan dan Kepala Badan Kepegawaian Negara Nomor 24 Tahun 2014 Tentang Jabatan Fungsional Dosen dan Angka Kreditnya*. 2014.
- [2] I. Laengge, H. F. Wowor, and M. D. Putro, "Sistem Pendukung Keputusan Dalam Menentukan Dosen Pembimbing Skripsi," *E-Journal Tek. Inform.*, vol. 9, no. 1, 2016.
- [3] R. Ruli, A. Siregar, F. A. Sinaga, and R. Arianto, "Aplikasi Penentuan Dosen Penguji Skripsi Menggunakan Metode Tf-Idf Dan Vector Space Model," *J. Comput. Sci. Inf. Syst.*, vol. 1, no. 2, pp. 171–186, 2017.
- [4] R. Feldman and J. Sanger, *The Text mining Handbook Advanced Approaches in Analyzing Unstructured Data*. Cambridge, New York: Cambridge University Press, 2007.
- [5] G. Salton, *Automatic text processing: The transformation, analysis, and retrieval of*. 1989.
- [6] F. Z. Tala, *A Study of Stemming Effects on Information Retrival in Bahasa Indonesia*. Amsterdam: Master of Logic Project, 2003.
- [7] T. M. Isa and F. Abidin, "Mengukur Tingkat Kesamaan Paragraf Menggunakan Vector Space Model untuk Mendeteksi Plagiarisme," *Semin. Nas. dan Expo Teknik Elektro*, pp. 229–234, 2013.
- [8] F. Amin, "Sistem Temu Kembali Informasi Dengan Metode Vector Space Model," *J. Ilm. Fifo*, vol. IX, no. 1, pp. 74–80, 2012.
- [9] R. Y. Baeza and B. N. Ribeiro, *Modern Information Retrieval*. New York: ACM Press, 1999.
- [10] D. C. Manning, P. Raghavan, and H. Schutze, "An Introduction to Information Retrieval," no. c, 2009.
- [11] J. Ghosh, R. Mooney, and A. Strehl, "Impact of Similarity Measures on Web-page Clustering," *Work. Artif. Intell. Web Search (AAAI 2000)*, pp. 58–64, 2000.
- [12] F. Gorunescu, *Data Mining Intelligent Systems Reference Library, Volume 12*. Springer, 2011.