Kombinasi Algoritma Data Reduksi untuk Optimalisasi Dokumen Cluster

Siti Mujilahwati[1]
Fakultas Teknik, Program Studi Teknik Informatika
Universitas Islam Lamongan
Lamongan, Indonesia
e-mail: [1]moedjee@gmail.com

Diajukan: 10 November 2023; Direvisi: 5 Oktober 2023; Diterima: 18 Oktober 2023

Abstrak

Clustering adalah proses pengelompokkan tanpa pelatihan (unsupervised learning), salah satu algoritma yang dapat diterapkan untuk clustering adalah K-Means. Algoritma ini memiliki kinerja dengan konsep menghitung jarak terdekat dari sebuah cluster. Penelitian ini bertujuan untuk melakukan optimasi hasil clustering data abstrak skripsi dengan algoritma K-Means tersebut. Upaya yang dilakukan untuk optimalisasi hasil cluster adalah dengan model kombinasi algoritma Latent Semantic Analysis (LSA), Term Frequency – Inverse Document Frequency (TF-IDF) dan Hashing. Seperti penanganan data teks pada umumnya sebelum dilakukan clustering telah dilakukan praproses untuk pembersihan dan normalisasi data. Setelah praproses selanjutnya dilakukan ekstraksi data dalam bentuk vektor dengan metode Term Frequency – Inverse Document Frequency (TF-IDF) dan Hashing. Hasil vektor yang dihasilkan pada proses ekstraksi selanjutnya dilakukan kombinasi dari algoritma LSA bertujuan untuk mereduksi data. Hasil pengujian dari 229 data skripsi dan 4 cluster menunjukkan kombinasi LSA dengan ekstraksi TF-IDF memiliki keunggulan waktu eksekusi lebih efisien, sedangkan kombinasi LSA-Hashing memiliki nilai F-measure lebih baik.

Kata kunci: *Teks mining, K-means, TF-IDF, Hashing, Latent Semantic Analysis (LSA).*

Abstract

Clustering is a grouping process without training (unsupervised learning). One of the algorithms that can be applied to clustering is K-Means. This algorithm has the concept of calculating the shortest distance from a cluster. This study aims to optimize the results of thesis abstract clustering data with the K-Means algorithm. Efforts are being made to optimize cluster results using a combination model of Latent Semantic Analysis (LSA), Term frequency-inverse document frequency (TF-IDF), and hash algorithms. As with handling text data in general, before clustering, pre-processing is carried out for data cleaning and normalization. After pre-processing, data extraction is carried out in vector form using the Term frequency-inverse document frequency (TF-IDF) and Hashing methods. The vector results generated in the extraction process are then carried out in combination with the LSA algorithm aimed at reducing data. Test results from 229 thesis data and 4 clusters show that the LSA combination with TF-IDF extraction has the advantage of more efficient execution time, while the LSA-Hashing combination has a better F-measure value.

Keywords*: Text mining, K-means, TF-IDF, Hashing, Latent Semantic Analysis (LSA).*

1. Pendahuluan

Data teks adalah data dalam format teks. Perkembangan teknologi digital mengubah semua data penyimpanan menjadi data digital. Data yang dapat disimpan dalam jumlah besar dan dalam jangka waktu yang lama. Data teks yang diuraikan dalam artikel ini adalah data abstrak skripsi mahasiswa Teknik Informatika. Kumpulan data abstrak skripsi mahasiswa dari tahun ke tahun semakin bertambah banyak, dan datanya disimpan secara digital. Dari data abstrak tersebut dapat dimanfaatkan untuk menggali informasi dari kategori tema skripsi dan dapat dimanfaatkan untuk pemetaan tema skripsi di tahun berikutnya. Informasi kategori tema skripsi juga dapat dimanfaatkan untuk pengambilan keputusan lainnya, seperti pemberian tambahan kelas khusus serta pelatihan sesuai bidang minat mahasiswa dan sesuai target

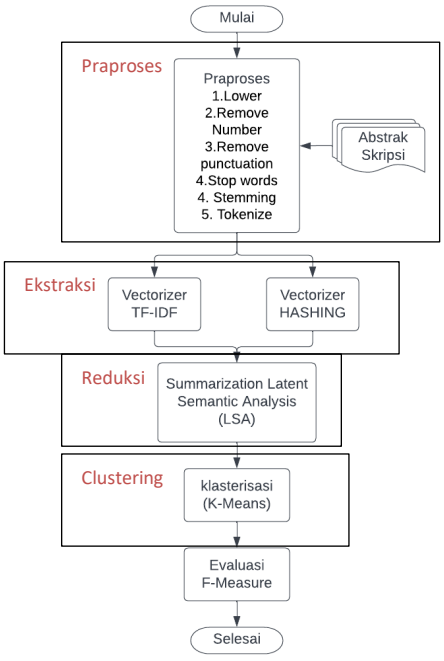
tema skripsi yang diharapkan. Untuk medapatkan informasi kategori dari tema skripsi tersebut dapat menggunakan Teknik mining. Teks mining merupakan sub bidang dari ilmu data mining.

Pengelompokkan data atau clustering merupakan bagian dari Teknik teks mining[1]. Pada Teknik ini kita dapat membagi beberapa kelompok sesuai dengan target analisis yang ingin dicapai. Pada penelitian ini menggunakan algoritma k-means untuk melakukan *clustering*[2]. Algoritma pengelompokan k-means menghitung *centroid* dan mengulanginya sampai *centroid* optimal ditemukan. Diperkirakan ada berapa banyak *cluster*. Ini juga dikenal sebagai algoritma pengelompokan datar. Jumlah *cluster* yang ditemukan dari data dengan metode dilambangkan dengan huruf 'K' dalam k-means[2]. Dalam algoritma ini, titik-titik data ditetapkan ke dalam cluster sedemikian rupa sehingga jumlah kuadrat jarak antara titik-titik data dan *centroid* sekecil mungkin[3]–[6]. Dalam beberapa penelitian sebelumnya telah banyak yang menguji algoritma K-Means untuk *clustering* data teks[2], [7]–[9], Adhe et al. telah melakukan penelitian yang sama dengan topik pengelompokkan data dokumen skripsi[10]. Gap yang dapat ditampilkan pada penelitian ini dengan penelitian sebelumnya adalah penambahan algoritma LSA untuk optimalisasi hasil *cluster*.

Data teks akan dirubah menjadi sebuah data *vector* dengan menghitung bobot setiap kata yang digunakan pada dokumen dengan algoritma TF-IDF (*Term Frequency – Inverse Document Frequency*) dan algoritma Hashing. Terdapat model atau metode peringkasan sebuah kalimat yaitu algoritma *Latent Semantic Analysis* (LSA). LSA adalah salah satu teknik dasar yang digunakan dalam pemodelan topik. Ide intinya adalah mengambil matriks dokumen dan istilah dan mencoba menguraikannya menjadi dua matriks terpisah yaitu Matriks topik dokumen dan Matriks istilah topik. Oleh karena itu, pembelajaran LSA untuk topik laten meliputi dekomposisi matriks pada matriks *term* dokumen menggunakan dekomposisi nilai Singular. Ini biasanya digunakan sebagai teknik pengurangan dimensi atau pengurangan data noise, dengan tujuan optimalisasi hasil cluster[11]. Penambahan LSA akan memberikan hasil lebih optimal dalam menilai kesamaan dokumen dan efisiensi waktu.

2. Metode Penelitian

Metode penelitian yang dilakukan secara umum digambarkan pada Gambar 1 berikut ini.



Gambar 1. *Flowchart* Penelitian

2.1. Praproses

Praproses pada umumnya dilakukan untuk membersikan dan normalisasi data, karena abstrak skripsi sudah memiliki struktur kalimat yang cukup baik. Maka praproses pada penelitian ini cukup menggunakan beberapa proses berikut ini :

1) Melakukan *case folding*, yaitu menyamakan semua ukuran font teks ke *lower case*
2) Menghapus angka
3) Menghapus tanda baca
4) *Stop words*, menghapus kata hubung

5) *Stemming*, mengembalikan kata dasar
6) *Tokenizing*, membentuk *bag of words*

2.2. Ekstraksi Data

Algoritma TF-IDF dan Hashing memiliki perbedaan cara mengekstraksi data teks. Algoritma TF-IDF dengan mengubah teks ke dalam *vector* berbasis bobot jumlah kata dalam sebuah korpus[12]–[14]. Untuk suku **t** dalam dokumen **d** , bobot **Wt,d suku** t dalam dokumen d diberikan oleh:

$$w_{t,d} = TF_{t,d} \log \frac{N}{DF_t} \quad (1)$$

- ◊ $TF_{t,d}$ adalah jumlah kemunculan t pada dokumen d.
- ◊ DF_t adalah jumlah dokumen yang mengandung istilah t.
- ◊ N adalah jumlah total dokumen dalam korpus.

Persamaan 1 akan digunakan untuk pembobotan tiap kata yang disebut term. Sebagai contoh penggunaannya sebagai berikut, dari data yang sudah dibersihkan dan dibentuk bag of word dalam bentuk table term dokumen matrik.

Tabel 1. Term Dokumen Matrik

Term(t)	Dokumen(d)/ TF			IDF =	w		
	D1	D2	D3	$Log_{10}(N/DF_t)$	D1	D2	D3
Data	1	1	2	0	0	0	0
Pelanggan	1	0	0	0.47	0.47	0	0
Sistem	1	1	2	0	0	0	0
Algoritma	2	1	1	0	0	0	0
Citra	0	1	3	0.17	0	0.17	0.51
Hasil	1	1	1	0	0	0	0
Mining	2	0	0	0.47	0.94	0	0

Sedangkan algoritma Hashing melakukan ekstraksi data ke dalam bentuk *vector* dengan cara mengubah data teks atau atribut kategorikal dengan kardinalitas tinggi menjadi *vector*[15].

function hashing_*vector*izer (fitur : **array string ,** N : integer) :
x : = vektor **baru** [N] **untuk** f **dalam** fitur :
h := hash (f) x [h **mod** N] += 1 return x

2.3. Clustering

Berikut adalah alur dari algoritma K-Means :

1. Pilih jumlah cluster (K) dan dapatkan titik data
2. Tempatkan *centroid* c_1, c_2, ... c_k secara acak
3. Ulangi langkah 4 dan 5 sampai konvergen atau sampai akhir sejumlah iterasi yang tetap
4. untuk setiap titik data x_i:
   - temukan *centroid* terdekat (c_1, c_2 .. c_k)
   - tetapkan titik ke cluster itu
5. Untuk setiap cluster j = 1..k
   - *centroid* baru = rata-rata semua titik yang ditetapkan ke cluster itu
6. End

3. Hasil dan Pembahasan

Penelitian ini menggunakan dataset abstrak skripsi mahasiswa Teknik informatika sebanyak 229 dokumen, dengan jumlah claster sebanyak 4 claster (k=4). Untuk hasil penelitian ini akan dibahas pada point berikut ini.

3.1. Hasil Praproses

Dikarenakan data teks abstrak skripsi merupakan teks yang sudah tersetruktur sehingga pada praproses ini hanya diperlukan beberapa tahap saja, yaitu *case folding (lower case), remove number, remove punctuation, stop word, stemming dan tokenize.*

Abstrak	Lowercase	Remove_Number	Remove_Punctuation	Stopwords_token	Stemming
Keretakan pada kulit telur merupakan terbentuk...	keretakan pada kulit telur merupakan terbentuk...	keretakan pada kulit telur merupakan terbentuk...	keretakan pada kulit telur merupakan terbentuk...	[keretakan, kulit, telur, terbentuknya, rongga...	[kereta, kulit, telur, bentuk, rongga, kulit, ...
Beras merupakan salah satu kebutuhan yang dibu...	beras merupakan salah satu kebutuhan yang dibu...	beras merupakan salah satu kebutuhan yang dibu...	beras merupakan salah satu kebutuhan yang dibu...	[beras, salah, kebutuhan, dibutuhkan, masyarak...	[beras, salah, butuh, butuh, masyarakat, konsu...
Toko Surya Phone merupakan usaha yang bergerak...	toko surya phone merupakan usaha yang bergerak...	toko surya phone merupakan usaha yang bergerak...	toko surya phone merupakan usaha yang bergerak...	[toko, surya, phone, usaha, bergerak, bidang, ...	[toko, surya, phone, usaha, gerak, bidang, jua...

Gambar 2. Hasil *Output* Praproses

Pada tahap praproses ini menggunakan fungsi library Natural Language Toolkit (NLTK) milik Python. Proses *stemming* menggunakan *library* sastrawi. Sastrawi dapat menemukan kata dasar berdasarkan kamus yang dimiliki dan bersifat kata dasar yang paling mendekati, misalkan pada kata keretakan oleh sastrawi ditemukan kata dasar kereta, yang mestinya retak.

3.2. Fitur Ekstraksi TF-IDF

Hasil perhitungan TF-IDF yang diperoleh dari penelitian ini pada data indek atau dokumen abstrak ke 90 dengan potongan isi dokumen asli sebagai berikut.

Tabel 2. Data Ekstraksi

Teks	Hasil pra-proses
"Dengan adanya sistem pendukung keputusan dapat meningkatkan kualitas pengambilan keputusan yang akan dibuat. Sebagai contoh, dalam penentuan kualitas songkok untuk mengetahui kualitas songkok tersebut. Penentuan kualitas songkok ini memiliki beberapa kriterian diantaranya bos-bosan, bludru, lapisan dalam, lapisan kain. Dan terdapat beberapa jenis kualitasnya yaitu super, premium, standard dan rendah"	['sistem', 'dukung', 'putus', 'tingkat', 'kualitas', 'ambil', 'putus', 'contoh', 'tentu', 'kualitas', 'songkok', 'kualitas', 'songkok', 'tentu', 'kualitas', 'songkok', 'milik', 'kriterian', 'bosbosan', 'bludru', 'lapis', 'lapis', 'kain', 'jenis', 'kualitas', 'super', 'premium', 'standard', 'rendah']

Dari data praproses yang sudah membentuk Term maka selanjutnya adalah membentuk sebuah korpus untuk bobot masing-masing Term dengan menggunakan algoritma TF-IDF, hasil telah diimplementasikan dan hasil dapat dilihat seperti pada Gambar 3 berikut ini.

term	TF	TF-IDF
sistem	0.021505376344086023	0.010582356579460975
dukung	0.021505376344086023	0.03920008797655947
putus	0.03225806451612903	0.045431300413519046
tingkat	0.010752688172043012	0.0091263927406846
kualitas	0.08602150537634409	0.15680035190623787
ambil	0.010752688172043012	0.015534816741827584
contoh	0.010752688172043012	0.034801047593742154
tentu	0.07526881720430108	0.08203673195595582
songkok	0.06451612903225806	0.3058435369673738
milik	0.010752688172043012	0.010042068099182523
kriterian	0.010752688172043012	0.04661408295576484
bosbosan	0.010752688172043012	0.050973922827895646
bludru	0.010752688172043012	0.050973922827895646
lapis	0.021505376344086023	0.07500670654836401
kain	0.010752688172043012	0.050973922827895646
jenis	0.010752688172043012	0.016579588230576484
super	0.010752688172043012	0.04352072733800376
premium	0.010752688172043012	0.050973922827895646
standard	0.010752688172043012	0.04661408295576484
rendah	0.010752688172043012	0.026766484133202144

Gambar 3. Contoh Hasil Output perhitungan TF-IDF

Gambar 3 di atas menampilkan beberapa term teratas atau yang memiliki bobot TF-IDF paling tinggi dari seluruh dokumen abstrak.

3.3. Fitur Ekstraksi Hashing

Contoh hasil matrik vektor algoritma hashing pada suatu kalimat. Karena dokumen abstrak skripsi susunan kata 150 -200 kata. Maka pada penelitian ini setting nilai Feature = 150.

```
array([[ 0.10392044,  0.04533166, -0.05398701, ..., -0.03253814,
        -0.01622935,  0.03245995],
       [ 0.17997055, -0.05700682, -0.09247331, ...,  0.04626602,
         0.06644425,  0.0315497 ],
       [ 0.59317419, -0.21297221, -0.4777263 , ...,  0.06447661,
         0.02224008,  0.02927621],
       ...,
       [ 0.206118  ,  0.00654875,  0.25885486, ..., -0.07560027,
         0.03489074, -0.01545117],
       [ 0.1711135 ,  0.03663546,  0.0299836 , ..., -0.07151858,
        -0.03878739, -0.00607227],
       [ 0.2352739 ,  0.03822675,  0.02758088, ...,  0.02469229,
        -0.03818907,  0.0353088 ]])
```

Gambar 4. Contoh Hasil *Matric vector* Algoritma Hashing

Gambar 4 di atas menggambarkan hasil *vector* yang diperoleh dari algoritma Hashing, tampilan yang ditunjukkan diperoleh dengan menggunakan code python dari dokumen ke 90 seperti yang dicontohkan pada data korpus di atas.

3.4. Hasil Claster dengan Algoritma k-means

Dengan jumlah K=4, diperoleh data pusat atau *centroid* sebagai berikut.

```
Cluster 0: buah kualitas jenis fitur pisang klasifikasi ikan warna tingkat keputusan
Cluster 1: aplikasi produk penyakit bayes internet wisata proses keputusan prediksi naive
Cluster 2: penjualan barang peramalan prediksi toko proses memprediksi nilai perhitungan stok
Cluster 3: game cerita petualangan android unity indonesia level bermain masyarakat orang
```

Gambar 5. Hasil *Centroid* terpilih

Hasil clusterisasi dari 229 dokumen abstrak skripsi dan perbandingan hasil kombinasi dari metode yang diusulkan adalah sebagai berikut.

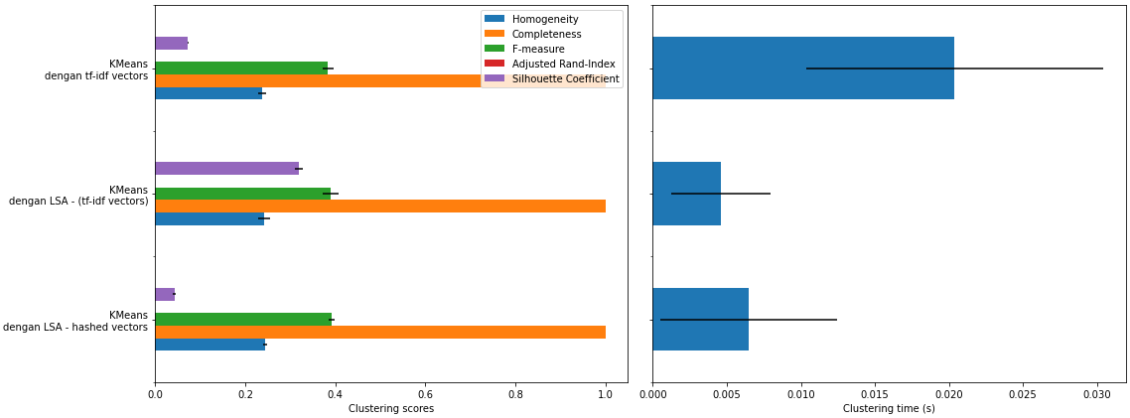
Tabel 3. Hasil Evaluasi Kombinasi Algoritma Cluster

Pengujian Kombinasi Algoritma		**Hasil evaluasi**	
		Score	**Waktu / detik**
k-means dengan TF-IDF	Homogeniety	0,238	
	Completeness	100	
	F-Measure	0,384	0.02
	Adjusted Rand-Index	0	
	Silhouette Coefficient	0,0783	
k-means dengan LSA-TF-IDF	Homogeniety	0,242	
	Completeness	100	
	F-Measure	0,389	0.00
	Adjusted Rand-Index	0	
	Silhouette Coefficient	0,319	
Kmeans dengan LSA-Hashing	Homogeniety	0,245	
	Completeness	100	
	F-Measure	0,393	0,01
	Adjusted Rand-Index	0	
	Silhouette Coefficient	0.043	

Evaluasi pengujian clustering dokumen abstrak meliputi Homogeniety, Completeness, F-Measure, Adjusted Rand-Index, Silhoutte Coefficient, dan waktu eksekusi. Uji homogeneity untuk mengukur tingkat populasi teks abstrak yang akan dilakukan cluster. Penelitian ini menggunakan 4 cluster dan banyak dokumen abstrak skripsi sebanyak 299. Pengujian melakukan 3 perbandingan pertama, clustering dengan *vector* TF-IDF saja. kedua cluster ditambahkan algoritma LSA, ketiga ditambahakn *vector* hashing.

Hasil pengujian Tabel 1, diperoleh tinggkat homogen sebesar 0,138 dan F-measure sebesar 0,384. Nilai tersebut dengan diberikan nilai 0 – 1. Begitu juga pada hasil clusterisasi dengan kombinasi algoritma

LSA dengan TF-IDF memiliki tingkat homogen 0,242 dan F-measure 0,389.sedangkan untuk kombinasi LSA- Hashing memiliki tingkat homogenitas 0,245 dan F-measure 0,393. Untuk evalusi kemiripan data dengan Silhouette Coefficient kombinasi LSA dengan ashing memiliki nilai jarak relative kecil. Nilai hasil Adjusted Rand-Index dari ketiga penujian tidak menunjukkan adanya nilai maksimum. Semua pengujian di setting dengan nilai max_iterasi = 100. Dari tiga pengujian waktu terbaik dalam memperoleh hasil adalah kombinasi algoritma LSA dengan TF-IDF.



Gambar 6. Visualisasi Hasil Evaluasi Cluster

4. Kesimpulan

Dari pengujian penelitian ini dapat disimpulkan bahwa penambahan fitur seleksi dengan algoritma *Latent Semantic Analysis* (LSA) dapat meningkatkan homogeneity dari dokumen skripsi. Dengan kombinasi LSA dengan ekstraksi TF-IDF memiliki keunggulan eksekusi lebih efisien, sedangkan kombinasi LSA-Hashing memiliki nilai F-measure lebih baik. Pada penelitian ini tidak dilakukan evaluasi *accuracy*, hal tersebut karena dataset yang digunakan tanpa adanya pelabelan kelas. Penelitian ini hanya mengujikan algoritma unsupervised untuk mengelompokkan data teks.

Daftar Pustaka

[1] R. Feldman and J. Sanger, *The Text Mining Handbook*. Cambridge: Cambridge University Press, 2006. doi: 10.1017/cbo9780511546914.

[2] A. Nur Khormarudin, "Teknik Data Mining: Algoritma K-Means Clustering," *Jurnal Ilmu Komputer*, 2016.

[3] A. Sulistiyawati and E. Supriyanto, "Implementasi Algoritma K-means Clustring dalam Penetuan Siswa Kelas Unggulan," *Jurnal Tekno Kompak*, vol. 15, no. 2, 2021, doi: 10.33365/jtk.v15i2.1162.

[4] G. Gustientiedina, M. H. Adiya, and Y. Desnelita, "Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 5, no. 1, 2019, doi: 10.25077/teknosi.v5i1.2019.17-24.

[5] Y. N. Andi Cuhwanto and D. A. R, "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K-Means," *PETIR*, vol. 15, no. 1, 2021, doi: 10.33322/petir.v15i1.1358.

[6] A. U. Fitriyadi, "Algoritma K-Means dan K-Medoids Analisis Algoritma K-Means dan K-Medoids Untuk Clustering Data Kinerja Karyawan Pada Perusahaan Perumahan Nasional," *KILAT*, vol. 10, no. 1, 2021, doi: 10.33322/kilat.v10i1.1174.

[7] L. Buitinck, "Clustering text documents using k-means," *Scikit Learn*. 2017.

[8] F. Bulyga, "CLUSTERING THE CORPORATION OF TEXT DOCUMENTS USING THE K-MEANS ALGORITHM," *University News. North-Caucasian Region. Technical Sciences Series*, no. 3, 2022, doi: 10.17213/1560-3644-2022-3-33-40.

[9] N. G. Yudiarta, M. Sudarma, and W. G. Ariastina, "Penerapan Metode Clustering Text Mining Untuk Pengelompokan Berita Pada Unstructured Textual Data," *Majalah Ilmiah Teknologi Elektro*, vol. 17, no. 3, 2018, doi: 10.24843/mite.2018.v17i03.p06.

[10] D. Adhe, C. Rachman, R. Goejantoro, F. Deny, and T. Amijaya, "Implementasi Text Mining Pengelompokkan Dokumen Skripsi Menggunakan Metode K-Means Clustering Implementation Of Text Mining For Grouping Thesis Documents Using K-Means Clustering," *Jurnal EKSPONENSIAL*, vol. 11, no. 2, 2020.

[11] S. Kumar, V. B. Singh, and S. K. Muttoo, "Bug Report Classification by Selecting Relevant Features Using Chi Square, Information Gain and Latent Semantic Analysis," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*, 2021. doi: 10.1109/ICRITO51393.2021.9596496.

[12] R. Kumbhar, S. Mhamane, H. Patil, S. Patil, and S. Kale, "Text document clustering using K-means Algorithm with dimension reduction techniques," in *Proceedings of the 5th International Conference on Communication and Electronics Systems, ICCES 2020*, 2020. doi: 10.1109/ICCES48766.2020.09137928.

[13] A. M. Jalil, I. Hafidi, L. Alami, and E. Khouribga, "Comparative Study of Clustering Algorithms in Text Mining Context," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 3, no. 7, 2016, doi: 10.9781/ijimai.2016.376.

[14] M. Priya and S. Surya, "A comparative study on clustering algorithms for clustering text documents," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 2 Special Issue, 2019.

[15] S. Kaur, P. Kumar, and P. Kumaraguru, "Automating fake news detection system using multi-level voting model," *Soft comput*, vol. 24, no. 12, 2020, doi: 10.1007/s00500-019-04436-y.