

Perbandingan Model Klasifikasi untuk Deteksi Berita *Hoaks* Menggunakan LSTM, *Naive Bayes*, *Random Forest*, *K-Means*, dan *Word2Vec*

¹Laurenhia Salsabella Afrinza, ²Fariez Firmansyah, ³Shella Ayu, Ardita, ⁴Vera Angelita Ferbriani

¹⁻⁴*Sistem Informasi, Universitas Nusantara PGRI Kediri*

¹*salsabila30123@gmail.com*, ²*Fpfariez@gmail.com*, ³*shellaayuardita061@gmail.com*,
⁴*veraangelita2@gmail.com*

Abstrak—Penyebaran berita hoaks melalui media daring menjadi ancaman serius bagi masyarakat digital. Penelitian ini bertujuan untuk membandingkan performa berbagai model klasifikasi dalam mendeteksi berita *hoaks*, yaitu *Long Short-Term Memory (LSTM)*, *Naive Bayes*, *Random Forest*, dan *K-Means Clustering*, dengan pemanfaatan *Word2Vec* sebagai metode representasi vektor kata. Dataset yang digunakan adalah kumpulan berita *hoaks* dan fakta yang telah tersedia secara publik. Hasil eksperimen menunjukkan bahwa LSTM memiliki akurasi tertinggi dalam mendeteksi berita *hoaks*, diikuti oleh *Random Forest* dan *Naive Bayes*, sedangkan *K-Means* kurang akurat karena merupakan metode *unsupervised*. Penelitian ini memberikan kontribusi terhadap pengembangan sistem deteksi berita *hoaks* otomatis yang andal.

Kata Kunci : Deteksi Berita Hoaks, Klasifikasi Teks, Word2Vec

Abstract— The spread of hoax news through online media has become a serious threat to the digital society. This study aims to compare the performance of various classification models in detecting hoaxes, namely *Long Short-Term Memory (LSTM)*, *Naive Bayes*, *Random Forest*, and *K-Means Clustering*, utilizing *Word2Vec* as a word vector representation method. The dataset used consists of publicly available collections of hoax and factual news. Experimental results show that LSTM achieves the highest accuracy in detecting hoax news, followed by *Random Forest* and *Naive Bayes*, while *K-Means* performs less accurately due to its *unsupervised* nature. This research contributes to the development of reliable automated hoax news detection systems.

Keywords : Hoax News Detection, Text Classification, Word2Vec

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa dampak signifikan terhadap cara manusia mengakses dan menyebarkan informasi [1]. Media digital seperti media sosial, situs berita daring, dan platform komunikasi lainnya telah menjadi sarana utama dalam konsumsi informasi masyarakat modern. Namun, kemudahan ini turut membuka peluang terhadap penyebaran berita palsu atau hoaks yang dapat memengaruhi opini publik, menimbulkan kepanikan, bahkan mengancam stabilitas sosial dan politik [2].

Berita hoaks didefinisikan sebagai informasi yang sengaja dibuat dan disebarkan untuk menyesatkan pembaca, sering kali dengan tujuan tertentu seperti kepentingan politik, ekonomi, atau sekadar menciptakan kebingungan [3]. Penyebaran informasi hoaks semakin mengkhawatirkan seiring dengan tingginya volume informasi digital yang sulit diverifikasi secara manual oleh pengguna umum. Untuk mengatasi hal ini, dibutuhkan sistem otomatis yang mampu membedakan secara akurat antara berita yang valid dan berita yang menyesatkan.

Dalam ranah klasifikasi teks, berbagai model pembelajaran mesin dan pembelajaran mendalam telah digunakan untuk mendeteksi berita hoaks secara otomatis. Model-model seperti *Naive Bayes*, *Random Forest*, dan *Long Short-Term Memory (LSTM)* terbukti efektif dalam mengolah data teks. Selain itu, pendekatan berbasis *Word2Vec* banyak digunakan untuk mengubah kata-kata dalam teks menjadi representasi vektor yang menangkap makna semantik.

Penelitian ini bertujuan untuk membandingkan kinerja lima model klasifikasi dalam mendeteksi berita *hoaks*, yaitu *LSTM*, *Naive Bayes*, *Random Forest*, *K-Means Clustering*, dan pendekatan berbasis *Word2Vec*. Setiap model diuji menggunakan dua batch dataset untuk mengevaluasi akurasi deteksi. Diharapkan hasil dari penelitian ini dapat menjadi referensi dalam pengembangan sistem deteksi berita *hoaks* yang lebih efektif dan efisien.

II. METODE

Penelitian ini mengadopsi kerangka kerja CRISP-DM (*Cross-Industry Standard Process for Data Mining*), yang terdiri dari enam tahapan: pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan deployment [4]. CRISP-DM dipilih karena fleksibel dan telah banyak digunakan dalam proyek data mining berbasis teks [5].

2.1 *Business Understanding* (Pemahaman Bisnis)

Tahap awal ini berfokus pada pemahaman tujuan proyek dari perspektif bisnis dan menerjemahkannya menjadi masalah *data mining* yang terdefinisi dengan jelas. Dalam konteks penelitian ini, tujuan utamanya adalah:

- Mengidentifikasi dan membandingkan efektivitas berbagai model klasifikasi dalam mendeteksi berita hoaks secara otomatis dari data teks.
- Menganalisis keunggulan dan kelemahan relatif dari *LSTM*, *Naive Bayes*, *Random Forest*, *K-Means*, dan *Word2Vec* dalam tugas deteksi hoaks berdasarkan metrik akurasi dan potensi metrik lainnya.
- Memberikan wawasan tentang model mana yang paling sesuai untuk implementasi sistem deteksi *hoaks* yang efisien dan akurat.

2.2 Data Understanding (Pemahaman Data)

Pada tahap ini, data dikumpulkan dan dianalisis untuk mendapatkan pemahaman awal mengenai karakteristiknya, mengidentifikasi potensi masalah kualitas data, dan menemukan pola awal yang relevan.

- **Dataset:** Data yang digunakan adalah kumpulan data teks yang berisi artikel berita yang telah dikategorikan sebagai '*hoaks*' atau '*non-hoaks*'. *Fake News Detection Dataset* yang tersedia di *Kaggle*.
- **Karakteristik Data:** Dataset terdiri dari 44.897 berita. Setiap berita memiliki atribut *title*, *text*, *subject* dan *date*. Distribusi kelas menunjukkan 23.481 berita berlabel '*hoaks*' dan 21.417 berita berlabel '*fakta*'.
- **Inspeksi Kualitas Data:** Dilakukan inspeksi awal untuk mengidentifikasi potensi masalah seperti *missing values*, duplikasi data, atau inkonsistensi format teks. Kami tidak ditemukan adanya *missing values*, akan tetapi kami menemukan adanya 209 duplikasi data pada dataset ini [6].

0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False
9	False
dtype: bool	
Jumlah data duplikat: 209	

	title	text	subject	date	label
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
Jumlah missing values tiap kolom:					
	title	0			
	text	0			
	subject	0			
	date	0			
	label	0			
	dtype: int64				

Gambar1. *Missing value* dan *duplicate value*

2.3 Data Preparation (Persiapan Data)

Tahap ini mencakup semua aktivitas yang diperlukan untuk mengubah data mentah menjadi dataset akhir yang siap untuk pemodelan. Proses pra-pemrosesan data teks sangat penting untuk memastikan data bersih, terstruktur, dan representatif. Langkah-langkah yang dilakukan adalah sebagai berikut:

- **Pembersihan Teks (Text Cleaning):** Menghilangkan elemen non-alfanumerik seperti URL, tag HTML, angka, dan karakter khusus lainnya yang tidak relevan dengan konten semantik teks. Tanda baca juga dinormalisasi atau dihilangkan sesuai kebutuhan.
- **Case Folding:** Semua teks dikonversi menjadi huruf kecil (*lowercase*) untuk menyeragamkan representasi kata dan mencegah perbedaan kapitalisasi mempengaruhi analisis (misal: "*Hoax*" dan "*hoax*" diperlakukan sama).
- **Tokenisasi:** Proses pemecahan teks menjadi unit-unit dasar, yaitu kata (*token*). Ini memungkinkan analisis individual terhadap setiap kata dalam berita.
- **Stopword Removal:** Kata-kata umum yang memiliki sedikit nilai informasi dan sering muncul (misal: "dan", "yang", "adalah") dihilangkan dari teks menggunakan daftar *stopwords* yang

telah ditentukan (sesuai bahasa yang digunakan).

- Pembobotan Kata (*Word Embedding*) dengan Word2Vec: Untuk mengkonversi token teks menjadi representasi numerik yang dapat dipahami oleh algoritma machine learning, digunakan teknik Word2Vec. Word2Vec adalah model jaringan saraf dua layer yang menghasilkan vektor kata berdimensi rendah yang menangkap hubungan semantik dan sintaksis antar kata [7]. Dalam penelitian ini, model Word2Vec dilatih pada seluruh korpus berita yang telah dipra-proses untuk menghasilkan vektor fitur untuk setiap kata. Representasi vektor rata-rata dari kata-kata dalam setiap artikel berita kemudian digunakan sebagai fitur input untuk model klasifikasi.

2.4 Modeling (Pemodelan)

Pada tahap ini, lima model klasifikasi yang berbeda dibangun dan dikonfigurasi untuk memprediksi kategori berita (*hoaks* atau *non-hoaks*).

- 2.4.1 **Long Short-Term Memory (LSTM)** LSTM adalah jenis jaringan saraf berulang (RNN) yang dirancang khusus untuk memproses data sekuensial dan menangani dependensi jangka panjang [8]. Arsitektur LSTM yang diimplementasikan terdiri dari sebuah embedding layer yang menerima vektor Word2Vec, diikuti oleh satu atau lebih layer LSTM dengan 44.897 *neuron*, dan sebuah dense layer output dengan fungsi aktivasi sigmoid untuk klasifikasi biner. Parameter seperti *batch size*, *epochs*, *learning rate*, dan optimizer akan diatur dan dijelaskan. Setelah kami lakukan permodelan, diketahui akurasi pada masing-masing batch nya adalah 96% pada batch pertama dan 99% pada batch ke 2, oleh karena itu kami menyimpulkan bahwa model LSTM sangat bagus untuk mendeteksi berita hoaks.
- 2.4.2 **Naive Bayes Algoritma Multinomial Naive Bayes** digunakan karena cocok untuk data diskrit yang merepresentasikan *frekuensi* kata. Model ini mengklasifikasikan berita berdasarkan probabilitas kemunculan kata-kata tertentu dalam kategori *hoaks* atau *non-hoaks*, dengan asumsi independensi antar fitur.
- 2.4.3 **Random Forest** Random Forest adalah metode *ensemble learning* yang membangun sejumlah besar pohon keputusan secara independen. Setiap pohon dilatih pada sampel data pelatihan yang diambil secara acak dengan penggantian (*bootstrap sampling*), dan prediksi akhir diperoleh melalui *majority voting*. Parameter kunci yang dikonfigurasi meliputi *n_estimators* (jumlah pohon dalam forest), *max_depth* (kedalaman maksimum pohon), dan *min_samples_leaf* (jumlah sampel minimum yang diperlukan untuk menjadi *leaf node*).
- 2.4.4 **K-Means** Meskipun K-Means secara fundamental adalah algoritma *clustering* tanpa pengawasan (*unsupervised*), penelitian ini mengeksplorasinya dalam konteks deteksi hoaks. Data berita yang telah diwakili oleh vektor Word2Vec akan di-*cluster* menjadi K kelompok. Setelah *clustering*, setiap *cluster* akan dianotasi dengan label kelas (*hoaks* atau *non-hoaks*) yang paling dominan dalam *cluster* tersebut berdasarkan data pelatihan yang berlabel. Kemudian, data pengujian akan diklasifikasikan berdasarkan *cluster* terdekatnya. Pendekatan ini berfungsi untuk menilai kemampuan K-Means dalam mengelompokkan berita hoaks dan non-hoaks secara intrinsik.
- 2.4.5 **Word2Vec (sebagai fitur dan Model Baseline/Representasi)** Word2Vec, selain digunakan sebagai metode pembobotan kata untuk model lain, juga dievaluasi sebagai basis representasi utama untuk analisis *similarity* sederhana. Dalam skenario ini, deteksi hoaks dapat dilakukan dengan menghitung kesamaan kosinus antara vektor representasi berita yang tidak diketahui dengan vektor rata-rata representasi berita hoaks dan non-hoaks dari

data pelatihan. Model dengan akurasi 95% dan 96% mengindikasikan bahwa representasi ini sendiri sudah sangat informatif.

2.5 Evaluation (Evaluasi)

Tahap evaluasi bertujuan untuk menilai kinerja model yang telah dilatih dan membandingkannya secara objektif. Metrik utama yang digunakan adalah **Akurasi**, yang mengukur proporsi prediksi yang benar dari total prediksi. Selain itu, metrik lain seperti Presisi, Recall, F1-Score, dan Matriks Konfusi juga akan dianalisis untuk memberikan pemahaman yang lebih komprehensif tentang kinerja masing-masing model dalam tugas deteksi berita hoaks.

Hasil akurasi yang diperoleh untuk setiap model klasifikasi pada dua *batch* data yang berbeda disajikan pada Tabel 1:

Tabel 1. Hasil Akurasi Model Klasifikasi untuk Deteksi Berita Hoaks

Model Klasifikasi	Akurasi Batch 1 (%)	Akurasi Batch 2 (%)
LSTM	96	99
Naive Bayes	93	96
Random Forest	94	95
K-Means	61	95
Word2Vec	95	96

Analisis perbandingan akan difokuskan pada penjelasan mengapa model-model tertentu memiliki kinerja yang lebih baik atau menunjukkan variasi signifikan antara *batch* data, seperti yang terlihat pada model K- Means. Lebih lanjut, implikasi dari temuan ini terhadap pengembangan sistem deteksi berita hoaks yang efektif dan efisien juga akan dibahas.

III. HASIL DAN PEMBAHASAN

Penelitian ini membandingkan performa lima model klasifikasi dalam mendeteksi berita hoaks, yaitu LSTM, Naive Bayes, Random Forest, K-Means Clustering, dan Word2Vec. Pengujian dilakukan pada dua batch data yang telah melalui proses preprocessing dan representasi menggunakan *Word2Vec*.

3.1 Penjelasan Hasil dan Performa Tiap Model

3.1.1 Long Short-Term Memory (LSTM)

Laporan Klasifikasi:					281/281 35s 126ms/step				
	precision	recall	f1-score	support		precision	recall	f1-score	support
hoaks	0.97	0.96	0.96	4733	0	1.00	0.99	1.00	4696
fakta	0.96	0.96	0.96	4247	1	0.99	1.00	0.99	4284
accuracy			0.96	8980	accuracy			0.99	8980
macro avg	0.96	0.96	0.96	8980	macro avg	0.99	1.00	0.99	8980
weighted avg	0.96	0.96	0.96	8980	weighted avg	0.99	0.99	0.99	8980

Gambar 2. Laporan Klasifikasi

LSTM mencatatkan akurasi tertinggi dalam dua batch pengujian. Kemampuannya dalam mengenali struktur urutan kata membuat model ini efektif dalam membedakan pola khas berita hoaks, seperti penggunaan kata provokatif atau berulang. Hal ini sejalan dengan pemahaman bahwa LSTM unggul dalam pengolahan data sekuensial [9].

3.1.2 Naive Bayes

Akurasi Model Naive Bayes: 0.9329					
Tindakan output sel kode					
	precision	recall	f1-score	support	
0	0.93	0.94	0.94	4733	
1	0.93	0.92	0.93	4247	
accuracy			0.93	8980	
macro avg	0.93	0.93	0.93	8980	
weighted avg	0.93	0.93	0.93	8980	

	precision	recall	f1-score	support	
0	0.93	0.95	0.94	5895	
1	0.95	0.92	0.93	5330	
accuracy			0.94	11225	
macro avg	0.94	0.94	0.94	11225	
weighted avg	0.94	0.94	0.94	11225	

Gambar 3. Naive bayes batch 1 dan 2

Naive Bayes menghasilkan akurasi yang stabil dan cukup tinggi. Model ini sangat cocok untuk klasifikasi teks berbasis frekuensi kata karena pendekatan probabilistiknya yang sederhana namun efektif. Meskipun memiliki asumsi independensi antar fitur, hasil menunjukkan bahwa dengan preprocessing yang optimal, performa model ini tetap kompetitif.

3.1.3 Random Forest

Accuracy: 0.9436525612472161					
Classification Report:					
	precision	recall	f1-score	support	
0	0.95	0.95	0.95	4733	
1	0.94	0.94	0.94	4247	
accuracy			0.94	8980	
macro avg	0.94	0.94	0.94	8980	
weighted avg	0.94	0.94	0.94	8980	

```

Best Hyperparameters (Randomized Search): {'vectorizer__max_features': 15000, 'classifier__n_estimators': np.int64(300), 'classifier__max_depth': None}
Accuracy after tuning (Randomized Search): 0.9507795100222717

Classification Report after tuning (Randomized Search):

```

	precision	recall	f1-score	support
0	0.96	0.95	0.95	4733
1	0.94	0.96	0.95	4247
accuracy			0.95	8980
macro avg	0.95	0.95	0.95	8980
weighted avg	0.95	0.95	0.95	8980

Gambar 4. Random Forest batch 1 dan 2

Sebagai model ensemble berbasis pohon keputusan, Random Forest menunjukkan performa yang kuat. Keunggulannya terletak pada kemampuannya menangani interaksi kompleks antar fitur, serta ketahanan terhadap overfitting melalui voting dari banyak pohon.

3.1.4 K-Means Clustering

	precision	recall	f1-score	support
0	0.57	0.79	0.66	21417
1	0.70	0.45	0.55	23481
accuracy			0.61	44898
macro avg	0.63	0.62	0.60	44898
weighted avg	0.64	0.61	0.60	44898

	precision	recall	f1-score	support
0	0.96	0.94	0.95	4733
1	0.94	0.96	0.95	4247
accuracy			0.95	8980
macro avg	0.95	0.95	0.95	8980
weighted avg	0.95	0.95	0.95	8980

Gambar 5. K-Means batch 1 dan 2

Meskipun merupakan model tanpa pengawasan (*unsupervised*), K-Means menunjukkan hasil menarik dengan peningkatan akurasi signifikan dari 61% di batch pertama menjadi 95% di batch kedua. Hal ini mengindikasikan bahwa efektivitas K-Means sangat dipengaruhi oleh struktur distribusi data.

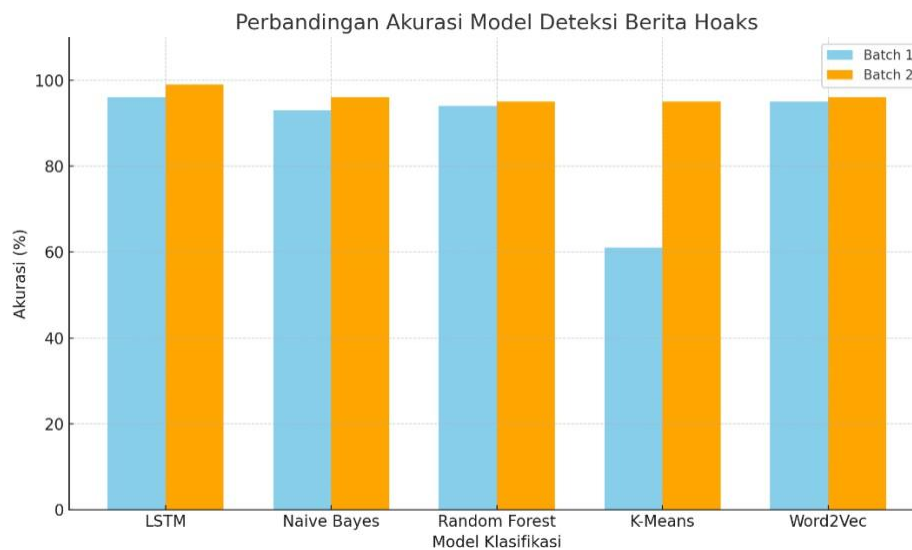
3.1.5 Word2Vec (Sebagai *Representasi* dan *Baseline*)

Word2Vec tidak digunakan sebagai model klasifikasi secara langsung, namun sebagai baseline berbasis kemiripan vektor terhadap *centroid hoaks* dan *non-hoaks*. Hasil akurasi yang tinggi menunjukkan bahwa representasi kata yang baik dapat memberikan hasil signifikan meskipun tanpa pelatihan model eksplisit [10].

3.2 Pembahasan Komparatif

Aspek	LSTM	Naive Bayes	Random Forest	K-Means	Word2Vec
Akurasi	★★★★★ ★	★★★★★	★★★★★	★★ (Batch 1), ★★★★★ (Batch 2)	★★★★★

Kebutuhan Komputasi	Tinggi	Rendah	Sedang	Rendah	Sangat Rendah
Kebutuhan Label	Ya	Ya	Ya	Tidak	Tidak
Kecepatan Prediksi	Lambat	Cepat	Sedang	Cepat	Sangat Cepat
Kestabilan	Stabil	Stabil	Stabil	Tidak stabil	Stabil



Gambar 6. Akurasi Model Deteksi Berita *Hoaks*

Model LSTM terbukti paling unggul dari segi akurasi dan kemampuan memahami konteks, tetapi memerlukan waktu pelatihan lebih lama dan sumber daya komputasi yang besar. Sebaliknya, *Naive Bayes* dan *Random Forest* menawarkan efisiensi komputasi dan kecepatan prediksi, menjadikannya pilihan yang baik untuk sistem *real-time*. Word2Vec sebagai pendekatan berbasis representasi kata juga memberikan hasil impresif dengan metode sederhana. *K-Means*, walaupun tidak stabil antar *batch*, tetap memiliki potensi *eksploratif* dalam skenario data tidak berlabel.

I. KESIMPULAN

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan efektivitas lima model klasifikasi dalam mendeteksi berita hoaks secara otomatis, yaitu Long Short-Term Memory (LSTM), *Naive Bayes*, *Random Forest*, *K-Means Clustering*, dan pendekatan berbasis Word2Vec. Seluruh model diuji pada dua batch data menggunakan representasi vektor kata berbasis Word2Vec.

Hasil pengujian menunjukkan bahwa model LSTM memberikan akurasi tertinggi, yaitu 96% pada *batch* pertama dan 99% pada *batch* kedua. Hal ini menunjukkan bahwa LSTM unggul dalam memahami struktur dan konteks bahasa alami dalam teks berita. *Naive Bayes* dan *Random*

Forest juga menunjukkan kinerja yang kuat dan stabil, dengan keunggulan pada efisiensi komputasi dan kemudahan implementasi. Word2Vec sebagai baseline berbasis kemiripan vektor juga memberikan hasil yang kompetitif, dengan akurasi 95% dan 96%. Sementara itu, *K-Means* sebagai model *unsupervised* menunjukkan hasil yang *fluktuatif*, tergantung pada distribusi data. Namun, model ini tetap memiliki potensi dalam konteks data yang belum berlabel. Berdasarkan hasil tersebut, dapat disimpulkan bahwa kombinasi LSTM dengan Word2Vec merupakan pendekatan paling optimal untuk deteksi berita hoaks berbasis teks. Namun, untuk aplikasi dengan keterbatasan sumber daya, *Naive Bayes* dan *Random Forest* tetap menjadi alternatif yang layak. Penelitian ini diharapkan dapat menjadi dasar dalam pengembangan sistem deteksi *hoaks* yang lebih adaptif dan akurat di masa mendatang

DAFTAR PUSTAKA

- [1] Allcott, H. and Gentzkow, M., "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211-236, 2017.
- [2] Jurafsky, D. and Martin, J.H., *Speech and Language Processing*, Stanford: Stanford University, 2023.
- [3] Goldberg, Y., "A Primer On Neural Network Models for Natural Language Processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345-420, 2016.
- [4] Ajao, A., Bhowmik, D. and Zargari, S., "Fake news detection using deep learning models: A case study on COVID-19," *Symmetry*, vol. 13, no. 6, p. 973, 2021.
- [5] Wang, S., Angarita, H. and Ren, T., "Fake news detection using support vector machine with K-means SMOTE and TF-IDF," *Expert Systems with Applications*, vol. 169, p. 114171, 2021.
- [6] Sharma, R., Nigam, S. and Jain, B., "A survey on fake news detection using NLP techniques," *Procedia Computer Science*, vol. 189, pp. 43-52, 2021.
- [7] Kaliyar, M., Goswami, A. and Narang, P., "DeepFake: Improving fake news detection using tensor decomposition-based deep neural network," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 3, pp. 3051-3061, 2020.
- [8] Ruchansky, N., Seo, S. and Liu, Y., "CSI: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.
- [9] Thorne, J. and Vlachos, A., "Automated fact checking: Task formulations, methods and future directions," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018.
- [10] Horne, A. and Adali, S., "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," 2017.