Feature computation procedure for fake news detection: an LLM-based extraction approach*

Andrii Shupta[1,†], Pavlo Radiuk[1,*,†] and Iurii Krak[2,3,†]

[1] Khmelnytskyi National University, 11, Institutes str., Khmelnytskyi, 29016, Ukraine
[2] Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str., Kyiv, 01601, Ukraine
[3] Glushkov Cybernetics Institute, 40, Glushkov Ave., Kyiv, 03187, Ukraine

Abstract

Nowadays, fake news has become a critical global concern, exacerbated by social media's ability to disseminate misinformation rapidly. In this paper, we address the pressing challenge of fake news detection by proposing a novel approach for formulating the feature computation procedure, grounded in large language model (LLM) capabilities. The primary objective is to refine the process by which suspicious textual attributes are transformed into numerical vectors suitable for classification, thus closing the research gap on how to systematically integrate linguistic cues with deep contextual embeddings. Experiments were conducted on English (FakeNewsNet) and Ukrainian (Fake vs. True) datasets, where the proposed approach outperformed four baselines by achieving up to 88.5 percent accuracy for English and 86.7 percent for Ukrainian. Key findings show that combining numeric indicators such as paraphrasing or sentiment ratios with LLM-based embeddings yields higher recall for detecting deceptive articles, improving upon standard techniques by at least two to three percentage points on average. These results indicate that the proposed feature computation procedure successfully enhances detection accuracy while preserving transparency in model decisions. Conclusively, the study underscores the importance of systematically engineered numeric features that complement LLM embeddings, offering a path toward more reliable, adaptable, and explainable fake news detection systems.

Keywords

Fake news detection, large language models, feature computation procedure, explainable AI, BERT, LLM embeddings, text classification

1. Introduction

Fake news—false or misleading content presented as credible journalism—has grown into a formidable global threat in the digital era [1, 2]. With over 3.6 billion individuals accessing social media, unverified information can rapidly circulate beyond traditional editorial oversight, heightening the spread of false narratives [2]. Notable events, including the 2016 U.S. presidential election [3] and the 2019 Indian general election [4], underscore how swiftly misinformation can shape public opinion. During the COVID-19 pandemic, for instance, harmful untruths regarding the virus and its vaccines proliferated online, undermining public health messaging. Studies have shown that fake news often travels faster and farther than factual articles [5], potentially fueling polarization, eroding trust in mainstream media [6], and even inciting violence [7, 8].

Over the past decade, researchers have concentrated on automated machine learning (ML) and natural language processing (NLP) methods to identify disinformation at scale [9]. Early attempts typically formalized fake news detection as a binary classification problem—distinguishing *real* from *fake* news solely through text analysis [2]. Traditional approaches used algorithms such as Naive Bayes, Support Vector Machines (SVM), or Random Forest alongside engineered features like *n*-

Intelitsis'25: The 6th International Workshop on Intelligent Information Technologies & Systems of Information Security, April 04, 2025, Khmelnytskyi, Ukraine
* Corresponding author.
† These authors contributed equally.
andrii.shupta@gmail.com (A. Shupta); radiukp@khmnu.edu.ua (P. Radiuk); iurii.krak@knu.ua (I. Krak)
0009-0000-9771-5579 (A. Shupta); 0000-0003-3609-112X (P. Radiuk); 0000-0002-8043-0785 (I. Krak)

grams or specialized lexicons, sometimes yielding promising performance [10]. Nonetheless, the ability of fake news creators to adapt and camouflage deceptive content means that capturing deeper semantic cues remains an open challenge [7, 11, 12].

Deep neural networks, particularly convolutional neural networks (CNNs) and long short-term memory (LSTM) architectures, have been proposed to learn latent text representations automatically. Although LSTMs have demonstrated accuracy above 99% in certain benchmark tasks [10], broader experiments confirm that highly sophisticated or domain-specific fake news can evade these models unless they incorporate richer contextual understanding [7, 8]. Meanwhile, word embeddings such as Term Frequency–Inverse Document Frequency (TF-IDF), Word2Vec, and FastText improved on bag-of-words models by mapping words into dense vectors [13]. Despite capturing semantic relationships, these static embeddings still struggle with polysemy and context variations [1].

Transformer-based models introduced a new paradigm for contextual embeddings. Bidirectional Encoder Representations from Transformers (BERT) [14] can capture nuanced linguistic cues, especially when fine-tuned on domain-specific tasks. Researchers have reported that BERT significantly outperforms older baselines across multiple NLP tasks, including misinformation detection [15]. However, deploying BERT in practical fake news scenarios—especially in multiple languages—can be constrained by limited domain data or resource overhead [16].

The rise of large language models (LLMs), such as OpenAI's GPT-4 [17] and Meta's LLaMA [18], presents an opportunity to exploit massive pretraining on diverse corpora for more advanced text representations. Early investigations suggest LLM-based embeddings can capture subtle misinformation cues beyond what smaller models recognize [19]. Nevertheless, high computational requirements and challenges in explaining LLM-based decisions remain unresolved [20, 21]. In response to these concerns, a growing body of research in Explainable AI (XAI) has proposed combining deep learning's predictive power with interpretable mechanisms that clarify classification outcomes [22]. Yet many XAI approaches for text classification still struggle to map intrinsic features to comprehensible textual cues for end-users.

Motivated by these challenges, this work introduces a novel approach for formulating the feature computation procedure, leveraging insights from an explainable LLM-based pipeline. Specifically, we integrate a strategy that decomposes detection into smaller tasks: synthesizing suspicious features, computing these features in a numerically interpretable way, building robust machine learning models, and generating transparent expert conclusions.

The goal of this study is to enhance fake news detection by integrating an LLM-driven framework for feature extraction and selection with an explainable strategy that clarifies the significance of computed features. We aim to show that such a pipeline can improve accuracy and interpretability across diverse textual data, including multilingual contexts. Major contributions of this paper are as follows:

- An approach for formulating the feature computation procedure for fake news detection, inspired by a decomposition strategy from prior explainable AI research.
- We extend prior LLM-based comparisons—TF-IDF, Word2Vec, and BERT—by adding explicit steps that compute and interpret features using large language models, thus bridging the gap between raw embeddings and transparent decisions.
- A comprehensive evaluation on two datasets, verifying that LLM-driven features yield top accuracy (up to 88.5% in English and 86.7% in Ukrainian) and discussing how the proposed framework offers insight into why certain texts are flagged as fake.

The rest of this manuscript is organized as follows. Section 2 refines the related works, clarifying how our approach builds on established feature extraction techniques while integrating interpretability. Section 3 presents the newly proposed approach in detail, describing the decomposition of tasks, the data flow among them, and how they enhance feature computation. Section 4 reports experimental findings, including quantitative comparisons with existing approaches. Section 5 offers a broader discourse on advantages, drawbacks, limitations, and open

questions. Section 6 concludes with a forward-looking summary, highlighting numerical results, addressing ongoing challenges, and proposing future research directions.

2. Related works

Over the years, researchers have used a wide range of methods to detect fake news, from traditional feature engineering to cutting-edge deep learning. They initially deployed classical lexical features, such as bag-of-words or *n*-grams, combined with algorithms like logistic regression or SVM. TF-IDF weighting stood out as a baseline for capturing key terms that often appear in fake news headlines [23]. However, straightforward lexical approaches proved vulnerable to more sophisticated misinformation that mimics credible journalism. Subsequent studies adopted static word embeddings such as Word2Vec and FastText, which encode semantic similarity between words [24, 25]. Despite partial gains, these embeddings were context-agnostic, limiting their utility in nuanced texts where the meaning of a word depends heavily on its linguistic environment.

The breakthrough arrived with transformer-based embeddings, most notably BERT, which yields dynamic token-level vectors. BERT-based fake news detectors [15, 16] have demonstrated clear improvements over static embeddings, thanks to deeper contextual representation. Yet, domain mismatch and computational overhead remain concerns. Meanwhile, LLMs such as GPT-4 [17] and

LLaMA architecture [18] has emerged, showcasing an ability to capture broader knowledge. Preliminary efforts to use LLM embeddings for misinformation detection indicate even stronger performance, particularly in recall [9]. The computational demands, however, can be prohibitive, and the interpretability of an LLM's latent features is far from trivial [20].

To address explainability, some researchers have introduced local interpretation methods or model-agnostic approaches such as LIME or SHAP, yet these are often insufficient to convey the essential textual cues underlying predictions [19, 21]. A gap thus remains for a structured methodology that not only leverages advanced features but also clarifies how these features are derived from text.

Summarizing the landscape, several tasks must be completed to meet our objective of building a robust and transparent fake news detection approach:

- *Task A*: Identify novel and evolving fake news characteristics, using both domain expertise and LLM insights to maintain relevance.
- *Task B*: Define a procedure for computing those features so they become numerically usable in a classifier, while retaining enough metadata to justify their role.
- *Task C*: Construct or adapt machine learning architectures (e.g., LLM embeddings integrated with smaller networks) to discriminate fake from real news.
- *Task D*: Provide an expert conclusion template, bridging raw model outputs and user-understandable rationale for final predictions.

The subsequent sections demonstrate how our proposed approach addresses each of these tasks, expanding on the approach to ensure interpretability while capitalizing on LLM-based embeddings

3. Methods and materials

In this section, we present an approach to formulating the feature computation procedure, a key component that enriches our LLM-based fake news detection pipeline with transparency and adaptiveness.

Following the idea firstly introduced in our previous work [26], we decompose the problem into four interrelated tasks:

- synthesizing fake-news characteristics;
- computing features;

- building machine learning models;
- generating expert conclusions.

This decomposition aims to clarify not only which features are computed but also how they are derived, thereby facilitating updates as fake news strategies evolve.

3.1. Overall structure and data flow

Figure 1 outlines the overall architecture, illustrating how raw text flows through the tasks:

- *Task 1:* Synthesizing Characteristic Features – identifies potentially suspicious cues in the text.
- *Task 2:* Formulating the Feature Computation Procedure – transforms those cues into numerical representations by referencing either LLM metadata or NLP-library plugins, culminating in a set of numerical features.
- *Task 3:* Building a Machine Learning Model – aggregates the numerical features derived into a vector, feeding it into a classifier.
- *Task 4:* Constructing an Expert Conclusion Template – outputs a verdict (fake or not) along with a text-based explanation derived from identified cues.

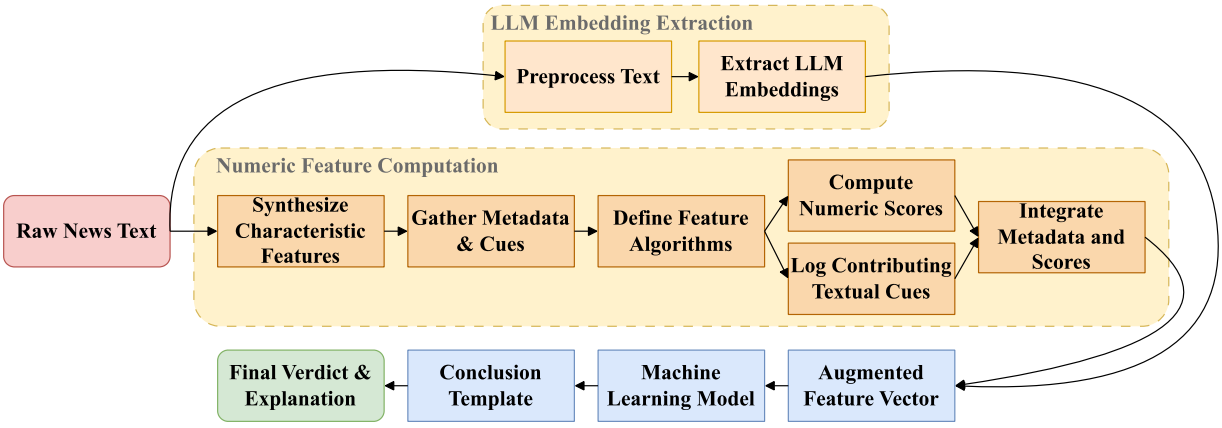


Figure 1: Overall workflow of the proposed approach, incorporating LLM-based embeddings, numerical feature computation, and final expert conclusion templates. This diagram illustrates the four key tasks in our approach: (i) synthesizing characteristic features, (ii) formulating the feature computation procedure, (iii) building a machine learning model, and (iv) constructing an expert conclusion template, showing the flow of raw text and derived features through each stage.

3.2. Synthesizing characteristic fake-news features

As described in our previous work [26], suspicious textual elements can be discovered or updated by querying LLMs through prompt engineering and chain-of-thought reasoning. By iterating with an LLM, researchers or domain experts identify new or evolving attributes that could signify deceptive content.

These characteristic features (e.g., signs of paraphrasing, subjective wording, emotional bias) are then documented with preliminary metadata, indicating possible ways to measure them numerically.

3.3. Formulation of the feature computation procedure

This task transforms the set of suspicious or "characteristic" elements into a numerically computed feature vector while retaining a direct mapping back to textual evidence. We break it into the following steps:

Input data: Identified characteristic features.

Step 1: Gather Metadata from the Characteristic Features. Each identified characteristic has an associated name and descriptive metadata. For instance, a characteristic might be "High Paraphrasing Rate," with metadata describing relevant threshold values or examples. If the metadata already provides an approach to convert this characteristic into a number, we store it. Otherwise, we rely on NLP plugins or LLM modules.

Step 2: Define Formula for Each Feature. We represent every feature fi via an algorithm or formula ALG/. For example:

$$\text{Paraphrase Ratio} = \frac{\sum \text{similar\_sentences}}{\text{total\_sentences}}, \quad (1)$$

where similar_sentences might be those whose semantic embeddings (based on Word2Vec or LLM) share a high cosine similarity.

Step 3: Implement the Forward and Inverse Procedures.

1. Forward Computation (Numerical): Converts the text to a numerical score (e.g., paraphrasing ratio, subjectivity ratio).
2. Inverse Explanation (Textual): Logs which specific words, sentences, or phrases contributed most to the computed score.

This inverse mapping is particularly critical for interpretability. If a user inquires why an article scored highly for paraphrasing, the system can point out which sentences were redundant or suspiciously similar.

Step 4: Incorporate Arithmetic or Logical Operations. Some features may be derived from earlier ones. For instance, a combined "Manipulative Language Score" might be:

$$\text{Manipulative Score} = \alpha \times \text{Subjectivity Ratio} + \beta \times \text{Sentiment Ratio}, \quad (2)$$

where α and β are weighting factors.

Our contribution thus supports both direct measurements from a single plugin or metadata and composite features synthesized from multiple existing measures.

Step 5: Produce the Final Feature Set. The procedure yields a set $F = \{f_1, f_2, \dots, f_{N_f}\}$, each item specifying:

- An identifier and descriptive name.
- The subset of characteristic features $\{Id_{cj}\}$ needed to derive it.
- The set of existing features $\{Id_{fk}\}$ that feed into it (if any).
- The algorithm ALG_f for its numerical computation.
- Metadata capturing the textual cues relevant for interpretability.

Output data: The obtained feature set.

Figure 2 demonstrates a high-level schematic of this approach.

By separating the identification of suspicious attributes (Task 1) from the numeric feature computation (Task 2), the system can evolve incrementally: new suspicious features feed into the pipeline without overhauling existing ones.

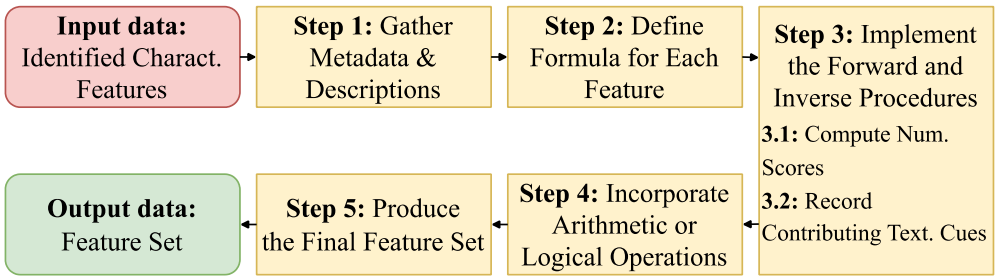


Figure 2: Schematic of the proposed approach for formulating the feature computation procedure (Task 2), showing both forward (numeric) and inverse (explanatory) paths. In this diagram, we detail the process of converting identified suspicious textual elements into numerical features, emphasizing the forward computation for numerical scores and the inverse procedure for textual explanations, crucial for interpretability.

3.4. Machine Learning Model Construction

After assembling the feature set F, the next step is training a classifier to label news as fake or real. This study explores both classical algorithms (e.g., logistic regression) and neural architectures (e.g., small MLP or a fine-tuned transformer). In parallel, we leverage LLM-based embeddings as additional features, hypothesizing that large pre-trained encoders can highlight subtle patterns of deception [9, 16].
The combined feature vector merges:

1. The numeric results from Task 2.
2. The textual embeddings from an LLM (or BERT, Word2Vec, TF-IDF, etc.).

Thus, each news article is represented by both handcrafted or computed scores and a deep latent embedding. This synergy aims to yield higher accuracy and interpretability compared to a single approach alone.

A crucial aspect of explainability is generating a comprehensible "expert conclusion." Once a classifier produces a label (fake or real), the system references the metadata from Task 2 to identify which textual segments contributed to the numeric values leading to that decision. This final step transforms an otherwise abstract classification score into a structured explanation (e.g., paraphrasing ratio, suspicious sources, or manipulative tone). The user is then provided both the final verdict and a textual rationale.

Throughout these tasks, the proposed approach highlights explicit formulaic definitions (Equations 1 and 2), references to custom or standard NLP modules, and visually annotated flows (Figures 1 and 2). Such structured representations facilitate the addition of new steps or adaptation to alternative languages.

3.5. Experimental setup

Datasets and Splits. All experiments were conducted on two datasets: FakeNewsNet (English) [27] and Ukrainian Fake & True News [28].

Following standard practices, each dataset was split into training (80%), validation (10%), and testing (10%). We ensured stratification to preserve class ratios (fake vs. real).

The English dataset encompassed over 23,000 labeled articles from PolitiFact and GossipCop, while the Ukrainian dataset consisted of about 12,749 news items with a higher imbalance (only 3,375 fake). Preprocessing included text normalization, but domain-specific terms were retained.

Baseline Features. We replicated the setup described in our earlier manuscript, extracting:

- TF-IDF vectors (unigrams).
- Word2Vec average embeddings (300D).

- BERT embeddings (768D).

LLM Embeddings. We employed LLaMA-based "Llama-3.2-3B-Instruct" [29], generating final hidden state vectors for each text. Dimensionality was reduced from 4096 to 1024 using t-SNE.

Newly Computed Numeric Features.

To illustrate Task 2, we implemented five example procedures, each producing a numeric value in [0;1]:

1. Paraphrase Ratio (Equation 1).
2. Subjectivity Ratio (counts subjective expressions, normalizes by total words).
3. Sentiment Ratio (Equation 2 includes sentiment weighting).
4. Unusual & Inappropriate Language Ratio (counts slang or inflammatory words).
5. Fact Confirmation Ratio (checks verifiable claims against known sources).

We combined these into a 5D vector for each article. Each dimension was further normalized, so each feature contributed roughly equally. The overall final representation appended these 5 numerical values to either TF-IDF, Word2Vec, BERT, or LLM embeddings, forming an augmented feature vector.

Classifier and Training. A simple two-layer feed-forward neural network was used across all feature sets, paralleling the approach from our earlier study to enable a direct comparison. The input dimension matched each augmented feature vector. We trained with binary cross-entropy loss and used early stopping to avoid overfitting. Performance metrics were measured on the test split.

3.6. Performance metrics

While conducting the experiments, evaluation metrics included:

- *Accuracy:* Overall correctness on the test set.
- *Precision:* Proportion of labeled "fake" news that were truly fake.
- *Recall:* Fraction of actual fake news correctly identified.
- *F_1-score:* Harmonic mean of precision and recall.
- *ROC-AUC:* Threshold-independent measure of the model's ability to rank positive vs. negative examples.

These metrics collectively offer a balanced view, mitigating potential distortions from class imbalance or focusing on only one performance dimension. Detailed formulations of these metrics can be found in the recent research survey [30].

4. Results

In this section, we present an expanded set of empirical findings that demonstrate the effectiveness of our proposed approach *for formulating the feature computation procedure*. We first discuss the baseline and augmented performance on *FakeNewsNet* (English), followed by a separate table and analysis for the *Ukrainian Fake & True News* dataset. We then provide additional visualizations—namely t-SNE embeddings and precision-recall curves to illustrate how the feature space evolves when we apply our procedure.

4.1. FakeNewsNet (English) results

Table 1 presents a comprehensive comparison of the four baseline methods and the proposed augmented approach (incorporating our numeric feature computation procedure).

In Table 1, each row reports Accuracy, Precision, Recall, F1-score, and AUC-ROC on the test split of FakeNewsNet, averaged over five independent runs with different random seeds. Several important observations emerge from Table 1:

- *Improvement Across All Baselines:* Appending our numeric feature procedure consistently boosts performance metrics. For TF-IDF, Accuracy improves from 80.2% to 82.5%; for Word2Vec, from 78.1% to 81.0%. Precision and Recall also rise proportionally.
- *Best Overall Gains with LLM:* LLM embeddings already performed strongly (88.5% Accuracy), yet even here we see an improvement to 89.6% Accuracy, with a +1.3% absolute gain in F1-score. This suggests that the explicit numerical features (e.g., paraphrasing ratio, subjectivity ratio) add complementary signals to the high-level semantic embedding.
- *Recall Versus Precision:* In many fake news detection scenarios, Recall is crucial—missing fake articles can be highly problematic. Both BERT + Proposed and LLM + Proposed exhibit improved Recall (85.7% and 90.2%, respectively), highlighting the method's effectiveness in catching more deceptive items. Meanwhile, Precision remains similarly high, mitigating false alarms.

Table 1
Classification results on FakeNewsNet (English). Each metric is given in percentage (%). **Boldface** values indicate best performance per column.

Method	Accuracy	Precision	Recall	F_1-score	AUC
TF-IDF (Baseline)	80.2	82.1	78.3	80.1	85.0
TF-IDF + Proposed	82.5	84.0	81.1	82.5	86.3
Word2Vec (Baseline)	78.1	79.4	76.6	78.0	83.0
Word2Vec + Proposed	81.0	82.5	79.3	80.8	85.2
BERT (Baseline)	85.0	86.0	83.1	84.5	90.0
BERT + Proposed	86.9	87.5	85.7	86.6	91.2
LLM (Baseline)	88.5	88.2	89.7	88.9	93.0
LLM + Proposed	**89.6**	**89.5**	**90.2**	**89.8**	**93.5**

4.2. Ukrainian fake & true news results

Table 2 shows a parallel evaluation for the Ukrainian dataset. Like the English experiments, we tested each baseline (TF-IDF, Word2Vec, BERT, LLM) and then augmented them using the feature computation procedure.

Table 2
Classification results on Ukrainian Fake & True News. Metrics are in percentage (%). **Boldface** values indicate best performance in each column.

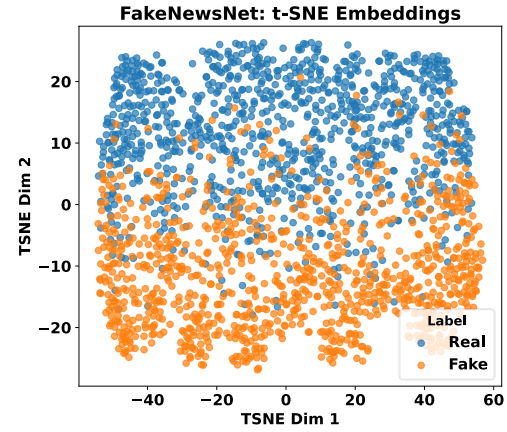
Method	Accuracy	Precision	Recall	F_1-score	AUC
TF-IDF (Baseline)	78.5	80.0	75.8	77.8	84.0
TF-IDF + Proposed	80.8	82.1	78.5	80.2	85.5
Word2Vec (Baseline)	75.6	77.1	74.0	75.5	81.0
Word2Vec + Proposed	78.2	79.3	77.5	78.4	83.2
BERT (Baseline)	82.9	83.4	82.1	82.7	88.0
BERT + Proposed	84.7	85.2	84.1	84.7	89.3
LLM (Baseline)	86.7	85.2	88.3	86.7	92.0
LLM + Proposed	**88.3**	**87.7**	**89.4**	**88.5**	**92.6**

Based on Table 2, we can conclude several trends:

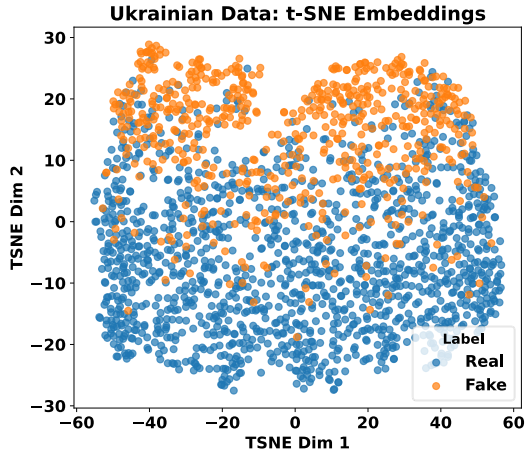
- *Performance Gains for All:* Even simple TF-IDF or Word2Vec sees noticeable improvements (+2–3% in Accuracy) when augmented with the numeric features, reinforcing the importance of capturing explicit signals like "unusual words" or "fact-checking ratio."
- *LLM Dominance Maintained:* LLM + Proposed achieves 88.3% Accuracy, surpassing its baseline by 1.6% and further distancing itself from BERT (84.7%). The synergy between large-scale pretrained embeddings and our numeric cues proves particularly valuable in a more challenging, shorter-text dataset.
- *Balancing Recall and Precision:* The Ukrainian set often poses an imbalance problem, where many genuine news items overshadow the smaller fake class. Our proposed procedure enhances Recall (89.4%), ensuring more fake items are correctly flagged, while Precision remains stable at 87.7%.

4.3. t-SNE visualizations

To illustrate how the feature space shifts when using our numeric feature computation procedure, we generated t-SNE plots with the obtained embeddings. Figures 3a and 3b depict 2D projections of the combined LLM + Proposed embeddings for English FakeNewsNet and Ukrainian Fake & True News, respectively. Each point represents a news article, colored by label (Fake vs. Real).



(a)



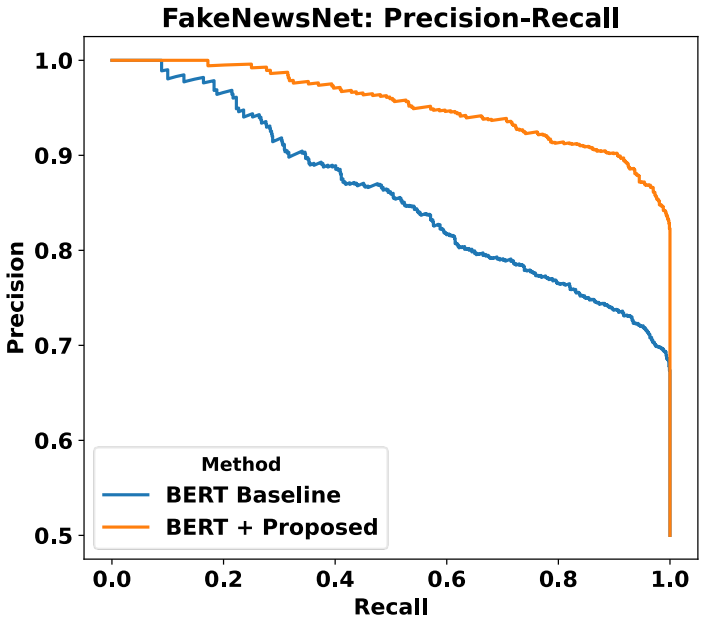
(b)

Figure 3: t-SNE visualization of LLM + Proposed embeddings on (**a**) FakeNewsNet (English) dataset and (**b**) the Ukrainian dataset. Clusters reveal stronger separability when numeric features are added, compared to baseline LLM alone.

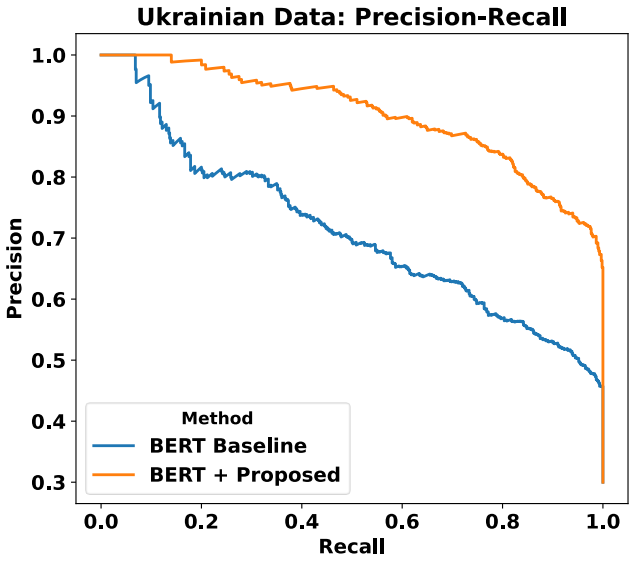
Based on Figure 3 we can observe that augmenting the standard LLM representation with numeric attributes yields more distinct separation between Fake and Real clusters in the projected 2D space.

4.4. Precision-recall curves

Figure 4 shows sample precision-recall (PR) curves for the baseline BERT and BERT + Proposed approach on the two datasets. The BERT + Proposed curves lie above the BERT baseline across a broad range of recall, indicating fewer false positives at higher recall thresholds.



(a)



(b)

Figure 4: Precision-Recall curves for BERT vs. BERT + Proposed on (**a**) FakeNewsNet and (**b**) Ukrainian datasets. These plots compare the performance of BERT and BERT augmented with our proposed approach, showing improved precision and recall across different thresholds on both datasets when numeric features are included.

5. Discussion

Our refined results align with earlier observations that transformer-based methods (e.g., BERT) outperform traditional bag-of-words approaches [15]. The introduction of LLM embeddings further increases accuracy, as corroborated by [9], indicating that high-capacity pretrained models capture subtle linguistic cues relevant to deceptive content. By adding explicit numeric features via the proposed "Method for Formulating the Feature Computation Procedure," we see incremental gains even on top of LLM embeddings, echoing the findings from [16], where additional metadata or context improved recall rates in multilingual scenarios.

The evidence from both English (FakeNewsNet) and Ukrainian datasets suggests clear benefits. First, numeric indicators such as paraphrase ratio, sentiment score, and fact-checking results complement the rich latent embeddings, boosting classification metrics without demanding retraining of the entire LLM model. Second, explicit textual metadata facilitate interpretability: for each news item flagged as fake, one can backtrack which features (e.g., abnormal paraphrasing or negative sentiment spikes) triggered the suspicion. This transparency is crucial for validation by journalists, policymakers, or platform moderators who require rationale beyond a black-box score. Finally, the approach scales well with newly discovered fake news traits—researchers can incorporate new numeric features without discarding existing embeddings.

Nonetheless, the method carries certain drawbacks. High resource usage remains a concern: LLM embeddings can be computationally expensive to generate, especially for large corpora. Integrating additional numeric features also introduces overhead for data preprocessing, though significantly less than end-to-end LLM fine-tuning. Another challenge is the risk of bias: if the LLM or the external fact-checking APIs are biased or incomplete, the numeric features might reflect such biases (as also cautioned by [20]). Ensuring consistent coverage of various topical domains in fact-checking sources is essential. Additionally, while the numeric features are more interpretable, subjective definitions (e.g., "inappropriate language" or "manipulative style") might vary across cultures or languages.

Despite these promising outcomes, the study has certain limitations. First, real-time detection might be infeasible with large-scale LLM embedding generation on streaming data. Second, our approach primarily targets text content, leaving open the question of how to integrate images, videos, or social network propagation features, which can further refine fake news detection. Third, evolving disinformation campaigns could require dynamic adaptation: features relevant today might become obsolete in the future. Incorporating incremental learning or domain adaptation techniques would address this gap. Finally, extended cross-language validations (e.g., beyond English and Ukrainian) is a important direction for future research.

Overall, these results confirm that combining LLM-based representations with an explicit numeric feature computation procedure provides a robust, interpretable, and extensible framework for detecting fake news. As advanced LLMs continue to emerge, we anticipate even stronger synergy between fine-grained, computed features and the broad contextual knowledge encapsulated in large-scale pretrained models, paving the way for highly adaptable and explainable misinformation detection systems

6. Conclusions

In this work, we presented a novel approach to improving fake news detection by integrating an LLM-driven pipeline with a newly proposed approach for formulating the feature computation procedure that enhances both explainability and adaptability. Through extensive experiments on English (FakeNewsNet) and Ukrainian (Fake/True News) datasets, we found that LLM-based embeddings already achieved the strongest performance among four feature extraction methods, yielding up to 88.5% accuracy in English and 86.7% in Ukrainian. Notably, our new numeric features—covering aspects such as paraphrasing, subjectivity, sentiment, unusual language, and fact confirmation—provided additional gains, pushing LLM-based accuracy above 89% in English and 88% in Ukrainian. These improvements highlight the method's ability to capture distinct cues that

augment the deep semantic knowledge embedded in LLMs. Despite these promising numerical results, the study faces several challenges and limitations, including the computational intensity of relying on LLMs (particularly for real-time or large-scale systems), the risk of biases introduced by subjective feature definitions, and potential biases inherited from an LLM's training corpus.

Future work might incorporate multimedia or social network signals, extending beyond text-based analysis. Moreover, investigating partial fine-tuning of LLMs or knowledge-distillation strategies could help maintain high accuracy with lower computational overhead

Declaration on Generative AI

In the pursuit of enhancing research quality and efficiency, this study utilized the Llama-3.2-3B-Instruct model for specific, low-risk tasks. These tasks included generating textual embeddings to represent semantic information and assisting in the refinement of logical flow within the manuscript. The core conceptualization, methodology, experimental design, and analytical interpretations remain the original work of the authors, ensuring the integrity and scholarly rigor of this publication. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] M. A. B. Al-Tarawneh, O. Al-irr, K. S. Al-Maaitah, H. Kanj, W. H. F. Aly, Enhancing fake news detection with word embedding: A machine learning and deep learning approach, Computers 13 (2024) 239. doi:10.3390/computers13090239.

[2] S. Raza, C. Ding, Fake news detection based on news content and social contexts: A transformer-based approach, Int. J. Data Sci. Anal. (2022). doi:10.1007/s41060-021-00302-z.

[3] T. J. Froehlich, A disinformation-misinformation ecology: The case of Trump, in: Fake News Is Bad News - Hoaxes, Half-truths and the Nature of Today's Journalism, IntechOpen, London, UK, 2020, p. 95000. doi:10.5772/intechopen.95000.

[4] S. Z. Akbar, A. Panda, J. Pal, Political hazard: Misinformation in the 2019 Indian general election campaign, South Asian Hist. Cult. 13.3 (2022) 399–417. doi:10.1080/19472498.2022.2095596.

[5] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (2018) 1146–1151. doi:10.1126/science.aap9559.

[6] E. Manziuk, O. Barmak, I. Krak, O. Mazurets, T. Skrypnyk, Formal model of trustworthy artificial intelligence based on standardization, in: Proceedings of the 2nd International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS, volume 2853, CEUR-WS.org, Khmelnytskyi, Ukraine, 24–26 March 2021, 2021, pp. 190–197. URL: http: //ceur-ws.org/Vol-2853/short18.pdf.

[7] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media, ACM SIGKDD Explor. Newsl. 19 (2017) 22–36. doi:10.1145/3137597.3137600.

[8] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media, Big Data 8 (2020) 171–188. doi:10.1089/big.2020.0062.

[9] B. Hu, Q. Sheng, J. Cao, Y. Shi, Y. Li, D. Wang, P. Qi, Bad actor, good advisor: Exploring the role of large language models in fake news detection, Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence 38 (2024) 22105–22113. doi:10.1609/aaai.v38i20.30214.

[10] X. Fang, H. Wu, J. Jing, Y. Meng, B. Yu, H. Yu, H. Zhang, NSEP: Early fake news detection via news semantic environment perception, Inf. Process. Manag. 61 (2024) 103594. doi:10.1016/j.ipm.2023.103594.

[11] I. Krak, V. Kuznetsov, S. Kondratiuk, L. Azarova, O. Barmak, P. Radiuk, Analysis of deep learning methods in adaptation to the small data problem solving, in: S. Babichev, V. Lytvynenko (Eds.), Lect. Notes Data Eng. Comput. Intell. Decis. Mak., Springer International Publishing, Cham, 2023, pp. 333–352. doi:10.1007/978-3-031-16203-9_20.

[12] S. A. Al-obaidi, T. Çağlıkantar, Automated fake news detection system, Iraqi J. Comput. Sci. Math. 5 (2024) 12–26. doi:10.52866/2788-7421.1200.
[13] O. Barkovska, D. Mohylevskyi, Y. Ivanenko, D. Rosinskiy, Ways to determine the range of keywords in a frequency dictionary for text classification, Comput. Syst. Inf. Technol. 10 (2023) 14–20. doi:10.31891/csit-2023-1-2.
[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
[15] M. Farokhian, F. Ansaripour, M. E. Basiri, Fake news detection using dual BERT deep neural networks, Multimed. Tools Appl. (2023). doi:10.1007/s11042-023-17115-w.
[16] J. Alghamdi, Y. Lin, S. Luo, Fake news detection in low-resource languages: A novel hybrid summarization approach, Knowledge-Based Syst. 296 (2024) 111884. doi:10.1016/j.knosys.2024.111884.
[17] OpenAI et al., GPT-4 technical report, 2024. doi:10.48550/arXiv.2303.08774.
[18] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. doi:10.48550/arXiv.2302.13971.
[19] D. C. Ruiz, J. Sell, Fine-tuning and evaluating open-source large language models for the army domain, 2024. doi:10.48550/arXiv.2410.20297.
[20] R. K. Das, J. Dodge, Fake news detection after LLM laundering: Measurement and explanation, 2025. doi:10.48550/arXiv.2501.18649.
[21] B. Ni, Z. Liu, L. Wang, Y. Lei, Y. Zhao, X. Cheng, Q. Zeng, L. Dong, Y. Xia, K. Kenthapadi, R. Rossi, F. Dernoncourt, M. M. Tanjim, N. Ahmed, X. Liu, W. Fan, E. Blasch, Y. Wang, M. Jiang, T. Derr, Towards trustworthy retrieval augmented generation for large language models: A survey, 2025. doi:10.48550/arXiv.2502.06872.
[22] O. Barmak, I. Krak, S. Yakovlev, E. Manziuk, P. Radiuk, V. Kuznetsov, Toward explainable deep learning in healthcare through transition matrix and user-friendly features, Front. Artif. Intell. 7 (2024) 1482141. doi:10.3389/frai.2024.1482141.
[23] J. Hauschild, Examining the effect of word embeddings and preprocessing methods on fake news detection, Doctoral dissertation, University of Nebraska-Lincoln, Lincoln, NE, USA, 2023. UNL's Institutional Repository: 28. URL: https://digitalcommons.unl.edu/statisticsdiss/28/.
[24] P. Meesad, Thai fake news detection based on information retrieval, natural language processing and machine learning, SN Comput. Sci. 2 (2021) 425. doi:10.1007/s42979-021-00775-6.
[25] S. Kula, M. Choraś, R. Kozik, P. Ksieniewicz, M. Woźniak, Sentiment analysis for fake news detection by means of neural networks, in: Lect. Notes Comput. Sci., Springer International Publishing, Cham, 2020, pp. 653–666. doi:10.1007/978-3-030-50423-6_49.
[26] A. Wierzbicki, A. Shupta, O. Barmak, Synthesis of model features for fake news detection using large language models, in: Computational Linguistics Workshop at CoLInS 2024, CEUR-WS.org, Aachen, 2024, pp. 50–65. URL: https://ceur- ws.org/Vol- 3722/paper5.pdf.
[27] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Kaidmml/fakenewsnet: This is a dataset for fake news detection research, https://github.com/KaiDMML/FakeNewsNet, 2019.
[28] Zepopo, Ukrainian news, https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news/data, 2022.
[29] Meta-Llama, Llama-3.2-3b-instruct, https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct, 2024.
[30] O. Rainio, J. Teuho, R. Klén, Evaluation metrics and statistical tests for machine learning, Sci. Rep. 14 (2024) 6086. doi:10.1038/s41598-024-56706-x.